EVALUATING ACCURACY AND ROBUSTNESS IMPACTS OF POWER USER ATTACKS ON COLLABORATIVE RECOMMENDER SYSTEMS

by

Carlos E. Seminario

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2015

Approved by:

Dr. David C. Wilson

Dr. Mirsad Hadzikadic

Dr. Heather Lipford

Dr. Robin Burke

Dr. Cem Saydam

ABSTRACT

CARLOS E. SEMINARIO. Evaluating accuracy and robustness impacts of power user attacks on collaborative recommender systems. (Under the direction of DR. DAVID C. WILSON)

Recommender systems help people quickly sort through large amounts of possible options by providing lists of personalized recommendations tailored to satisfy the end-user's preferences and inquiries. Today, these systems are used in a variety of applications such as e-commerce, travel, health care, education, news, research articles, financial services, online dating, and many others. For example, many top online retailers use systems to provide shoppers with recommendations on what products and services to buy, such as Amazon.com's *"Customers Who Bought This Item Also Bought .."* list of product offerings generated by their underlying recommender system. As with many online systems, however, there is a potential for malicious users to "game the system" for personal benefit or pleasure. This constitutes an "attack" on recommender systems and usually consists of having malicious users enter a number of fake ratings or reviews in order to promote or disparage an item for personal gain, or just to disrupt the system's recommendations.

The problem with attacks on recommender systems is that they bias the underlying data and cause the system to deliver erroneous or misleading recommendations to users. This can cause users to lose trust in the system. In the case of online retail systems, the user may choose to either (1) shop elsewhere, negatively impacting the sales of the attacked service/product provider, or (2) purchase the product only to find out that it does not meet their needs, negatively impacting user satisfaction with

the online recommender. There is abundant evidence in the media regarding the negative impacts that attacks on recommender systems can have on consumer behavior and the concomitant negative effects on system and service/product providers. And the impacts of attacks on recommender systems can potentially extend over many different applications and domains beyond e-commerce.

Recommender systems allow users to rate items and store those ratings into "user profiles", and they use different approaches to determine recommendations. The most popular approach, and the focus of this dissertation, is to use Collaborative Filtering, wherein many users rate items and the recommender's predictions are computed based on ratings provided by other "similar" users or items. This is in contrast to Content-Based systems where the basic process consists of matching attributes of a user profile, where preferences and interests are stated, with the attributes of other items in order to find and recommend new items that may be of interest to the user. Because recommender systems attacks have not been studied "in the wild", prior research has relied on laboratory-developed models of malicious users. Those models consist of user profiles containing item ratings based on statistical distributions, e.g., random or "average" ratings for items. These attack models have been analyzed in detail and attack detection methods based on those models have been researched. Our concern is that attackers continue to find new and less detectable forms of attack; therefore, extensions to current attack research are needed in order to keep up with advances in attack formulation.

This dissertation extends the body of knowledge of attack models by designing novel attacks based on explicitly-determined "influence" characteristics of users within a

recommender system. For collaborative recommenders, i.e., those based on ratings similarities between users, a social graph describing the relationships between users can be defined. And from Social Network Analysis, we know that central, or "power", users are those that wield influence over other users. In the recommender system context, the term "power user" denotes users who have considerable influence over the recommendations presented to other users. We are not advocating or suggesting that real power users will use their influence to attack recommender systems; instead, the concern is whether malicious users can attack a recommender system using carefully crafted *synthetic* power user identities capable of eluding detection. Power users are known to have strong influence over large groups of users; therefore, attackers that can acquire the influence accorded to power users may have strong impacts on system recommendations. This gives rise to the main research questions for this dissertation: (1) Does the use of Social Network Analysis identify more influential Power Users than other methods? (2) Will synthetic Power User profiles generated from power user characteristics retain the same level of influence of real Power Users? (3) What happens to Recommender System accuracy and robustness when power users attack? (4) Can a novel attack be crafted to achieve power user capability with strong influence and "low" cost of attack? (5) What is the most effective method of mitigating power user attacks?

The research strategy adopted in this dissertation consists of a quantitative experimental design using system accuracy and robustness as the constructs. In this context, accuracy is a measure used to evaluate a recommender system's predictive performance based on the difference between the predicted and actual user ratings;

robustness is a measure used to evaluate the stability of recommendations in the presence of fake information. The approach for this dissertation is empirically-focused in order to evaluate the accuracy and robustness of recommender systems under influence-based models of attack rather than previously studied statistically-based attack models. The variables include novel attack models developed in this dissertation (Power User Attack, Power Item Attack), power user selection methods (In-Degree Centrality, Aggregated Similarity, Number of Ratings), collaborative filtering algorithms (User-based, Item-based, SVD-based), and publicly-available domain datasets (MovieLens, Yahoo! Music). To evaluate the experimental results, accuracy metrics (Mean Absolute Error) and robustness metrics (Hit Ratio, Prediction Shift, Rank) are used, and statistical analyses are performed to test pre-established dissertation hypotheses.

The major findings are:

- A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects (from the attacker's viewpoint) on recommender system predictions and top-N lists of recommendations across multiple power user selection methods, collaborative filtering algorithms, and the movie and music domains tested.
- Power User Attack profiles generated from characteristics of In-Degree and Number of Ratings power users result in more effective attacks (from the attacker's viewpoint) than attack profiles generated from characteristics of Aggregated Similarity power users across collaborative filtering algorithms and the movie and music domains tested.

- The use of In-Degree Centrality to select a set of power users results in power users with higher influence than other selection techniques for user-based recommenders in the movie domain tested.
- A significant percentage of synthetic user profiles generated from statistical characteristics of power users were identified by the In-Degree and Number of Ratings power user selection methods in the movie and music domains tested.
- Item-based collaborative recommenders, previously considered robust to attack, are vulnerable to the novel Power Item Attack using a novel Multiple-Target design approach.
- Reducing the influence of power users is a more effective and less impactful mitigation strategy than completely removing power users from the dataset for user-based recommenders for the movie domain tested.

The principal conclusions are: (1) Power user attacks can have significant impact on the predictions generated by popular collaborative recommender algorithms across the movie and music domains tested, i.e., these attacks can efficiently and effectively bias the recommender predictions as measured by accuracy and robustness metrics, (2) Synthetic power user profiles generated from the In-Degree and Number of Ratings power user selection methods result in effective power user attacks, and (3) Due to its low "cost" of attack, the simple Number of Ratings method is the most efficient approach for selecting and generating power user profiles.

The implications of these findings are that system operators should be aware that collaborative recommenders are vulnerable not only to traditional attack models, but also to new attack vectors such as the Power User Attack model. System operators

should consider using an "influence reduction" mitigation strategy to defend against such attacks rather than power user elimination, i.e., they should seek to balance system accuracy and robustness objectives given the trade-offs between these measures during power user attacks. In order to generalize these findings, future work will need to experiment using larger, production-sized datasets with millions of users and items as well as testing in several domains.

DEDICATION

To my loving wife, Sue Kleckner Seminario,

who encouraged me to take on

this challenging

work

.

ACKNOWLEDGMENTS

I would like to thank Dr. David C. Wilson, my research adviser and Dissertation Committee Chair, for the excellent guidance, encouragement, and support he provided during this dissertation effort. I am also indebted to the members of my Dissertation Committee for their reviews and helpful comments on my proposal and dissertation: Drs. Robin Burke, Mirsad Hadzikadic, Heather Lipford, and Cem Saydam.

I would like to gratefully acknowledge the financial support provided by the UNCC Graduate School that included four years of Graduate Assistant Support Plan funding and the Giles Dissertation-Year Fellowship for the fifth and final year of this PhD journey. I would also like to thank other UNCC entities, the Software and Information Systems Department and the Graduate and Professional Student Government, for the travel funding they provided to cover expenses I incurred in traveling to various US and international conference venues to present my research papers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Recommender Systems (RS) help people quickly sort through large amounts of possible options by providing lists of personalized recommendations tailored to satisfy the end-user's preferences and inquiries. Today, these systems are used in a variety of applications such as e-commerce, travel, health care, education, news, research articles, financial services, online dating, and many others. For example, online shopping and browsing presents users with a vast amount of information and product choices. In order to deal with this type of information overload, many top online retailers use Recommender Systems to provide shoppers with a more personalized, and less daunting, shopping experience. In general, recommender systems analyze large amounts of data and information on behalf of application/system users to provide (a) recommendations on actions the user should take such as buying a product, selecting a restaurant, renting a movie, reading an article, etc., and (b) predictions, i.e., the expected value of the user's rating for an item (product, restaurant, movie, article, etc.), as a single value or as a ranked list of predicted values. There exist various types of recommender systems based on how they determine their recommendations and predictions [1]; the most popular are Collaborative Filtering (CF), wherein many users rate items and the recommender's predictions are computed based on the ratings provided by other "similar" users or items and Content-Based, where the basic process consists of matching attributes of a user profile, where preferences and inter-

ests are stated, with the attributes of other items in order to find and recommend new items that may be of interest to the user.

Recommender systems not only help users overcome the problem of information overload, they can also help online businesses drive sales by providing recommendations of additional items to online shoppers based on products they are currently browsing, e.g., Amazon.com's *"Customers Who Bought This Item Also Bought .."* provides a list of product offerings generated by their underlying recommender system. As with many online systems, there is a potential for some users to abuse the prediction mechanisms or subvert the results. For example, in order to have recommender systems favor their own product or diminish their competitor's product, unscrupulous or malicious sellers may attempt to "shill" [1] the system to have their products recommended more often and, hence, to increase their sales volume.

1.1 Problem of Attacks on Recommender Systems

One of the key problem areas in recommender systems is protecting "robustness", i.e., *stability of recommendations in the presence of fake information* [38] while maintaining a high level of system "accuracy", i.e, *predictive performance based on the difference between the predicted and actual user ratings* [56]. The problem with RS attacks is that, if left undetected or unmitigated, the system's database becomes compromised and can generate biased recommendations for users, thereby negatively impacting the system's accuracy and robustness characteristics. For example, this can cause online shoppers to waste time and money by following inaccurate or false

[1] A person who publicizes or praises something or someone for reasons of self-interest, personal profit, or friendship or loyalty.

recommendations and, in the long term, this problem could also diminish the users' trust in the online shopping experience.

Evidence of attacks on recommender systems are difficult to come by directly in the research literature because online system operators are loathe to share details of attacks on their systems for obvious security, privacy, and business reasons. However, some attacks to online recommender systems have been documented in the media and research literature. Amazon[2] was forced to remove a link to a sex manual that appeared associated with a spiritual guide by a well-known Christian televangelist; the two titles were somehow linked as a result of a recommender system that tracks and displays lists of merchandise under the title, "Customers who shopped for this item also shopped for these items"[3]. Online auction website eBay[4] which uses a recommender system as a reputation mechanism, found users who subverted the system by purchasing good ratings (feedback) from other members in order to bolster their own reputations.[5] More recently, online websites such as TripAdvisor[6] and Yelp[7] have been subjected to attacks known as "opinion spam" that include fake reviews to either promote or degrade specific products and services[8] [9] [10]. Burson-Marsteller[11], a global Public Relations and Communications consulting firm, surveyed 1,000 influential consumers' trust of online reviews. The study found that, compared to a

[2] www.amazon.com
[3] http://news.cnet.com/2100-1023-976435.html
[4] www.ebay.com
[5] http://www.auctionbytes.com/cab/abn/y03/m09/i17/s01
[6] www.tripadvisor.com
[7] www.yelp.com
[8] http://www.businessweek.com/magazine/a-lie-detector-test-for-online-reviewers-09292011.html
[9] http://www.dailymail.co.uk/travel/article-2059000/TripAdvisorcontroversy-Reviews-website-launches-complaints-hotlines.html
[10] http://www.cs.uic.edu/~liub/FBS/media-coverage.html
[11] www.burson-marsteller.com

similar poll conducted previously, an increasing number of consumers believed that fake reviews or positive comments on online websites were considered a problem.[12] Detection and analysis of opinion spam reviews have been studied by researchers [22, 42] using machine learning and semantic analysis methods.

1.2 Attack Methodology and Illustrative Example

In general, attacks can be perpetrated against content-based or collaborative filtering recommenders. In collaborative recommenders, attacks are typically used to either promote ("push") or demote ("nuke") a target item. Content-based systems are subject to "opinion spam" or fake reviews that either promote or demote (disparage) a product or service. In the collaborative filtering literature, attempts to influence recommender system results by providing false ratings are known as "shilling attacks" [26], or "profile injection attacks" [36]. A user profile contains the set of ratings a user has made when using the recommender system and attackers will submit one or more user profiles (called attack user profiles) containing fake item ratings that push or nuke a specific item called the target item. For a push attack, the target item's rating is set to the maximum rating value and for a nuke attack, the target item's rating is set to the minimum rating value. So, in order to mount an effective attack against collaborative recommender systems, malicious users carefully construct attack user profiles so that they appear to be "similar" to many other users by manipulating the number of profiles, the number ratings per profile, and the specific rating values inserted into the profiles including the target item [26, 10, 8, 36].

In order to illustrate a simple attack on a collaborative recommender system, con-

[12] http://www.adweek.com/news/advertising-branding/influencers-wary-fakes-90768

sider the user-item matrix shown in Figure 1, with 8 legitimate users (Bob, Ted, Fred, Ginger, Jodie, Jill, Tom, Corey), 3 attack users (Alice, Axel, Alvin), and 6 movie items (Avengers, Titanic, Avatar, Twilight, Psycho, Alien), as adapted from [36]. It shows the ratings that users have given to the items on a scale of 1 (disliked) to 5 (liked very much). The attackers have implemented a push attack on the target item Alien, they have attempted to increase the similarity between Avengers and Alien with ratings of 5, and they have also attempted to increase the similarity between the attackers and Bob with ratings of 5. In this example, Bob will request a recommendation from the recommender system for the movie Alien. Before the attack, popular collaborative recommender algorithms used for prediction (see Section 2.1 for more details), such as the user-based algorithm, would provide Bob a predicted rating of 2 for the movie Alien. After the attack by just one attacker (Alice), this same user-based algorithm would provide Bob a predicted rating of 4 for the movie Alien. Although the predicted rating, after the attack, will vary depending on the collaborative filtering algorithm used, the prediction will be higher than it would have been before the attack, misleading Bob to believe that he would actually enjoy the movie (see Section 3.4 for more details).

1.3 Research Problem and Research Questions

The possibility of designing and injecting user profiles into recommender systems to deliberately and maliciously manipulate the recommendation output of a Collaborative Filtering system was first raised by O'Mahony et al in 2002 [39]. Further work determined that attacks on recommender systems can be mounted by using one of several attack models. Attack models such as Random, Average, Bandwagon,

User Profiles	Avengers	Titanic	Avatar	Twilight	Psycho	Alien
Bob	5	2	3	3		?
Ted	2		4		4	1
Fred	3	1	3		1	2
Ginger	4	2	3	1		1
Jodie	3	3	2	1	3	1
Jill		3		1	2	
Tom	4	3		3	3	2
Corey		5		1	5	1
Alice	5		3		2	5
Axel	5	1	4		2	5
Alvin	5	2	2	2		5

Figure 1: Example – Push attack on a target item

Segment, etc., define the attack user profile data based on the attacker's knowledge of the underlying recommender system's algorithms, database, items, and/or users [6, 7, 26, 36, 34, 59]. These attack models inject artificial or *synthetic* attack user profiles that contain either random item ratings whose values are selected from a normal distribution around the mean rating of the dataset (this is not a very effective attack), item ratings whose values are selected from a normal distribution around the mean rating for each item (a more effective attack against neighborhood-based collaborative filtering algorithms), or a variant of these approaches. An important dimension of attacks on recommender systems is known as the cost of attack [9]. 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. Furthermore, 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. Therefore, attack models that have low knowledge requirements have an

edge over other models, costs being equal. Attacks on recommender systems have continued to be studied using a variety of machine learning and statistical analysis techniques in the areas of attack detection and improvements in algorithm robustness [38, 40, 10, 8, 36, 31]. The Influence Limiter algorithm [46] proves successful at mitigating attacks, albeit with low prediction accuracy [5]. Recommender systems using dimensionality reduction techniques, such as matrix factorization based on Singular Value Decomposition (SVD) [30, 32, 24], also appear to be robust to attack. A recent summary on RS robustness has been provided in [9]. It should be noted that none of these attacks explicitly consider the impact that user influence can have on recommendations.

Although attacks on RS have been studied in the past, users with malicious intent continue to find new ways to bias predictions and disrupt the system. Given that existing attack models use synthetic user profiles that are not representative of actual users and are more like statistically "average" users, we posit that there is a gap in the prior research that has ignored the characteristics of real, and more influential, "power" users that can be used to generate synthetic user profiles for attacking recommender systems.

So, what is a power user? The definition varies according to the context and usually refers to a small percentage of the user population. For computer users, a power user is a user of a personal computer who has the ability to use advanced features of programs which are beyond the abilities of "normal" users, but is not necessarily capable of computer programming and system administration. Power user can also be a marketing term referring to a computer user who seeks and uses

products having the most features and the fastest performance. In the social media context, for example, Pew Internet research found that 20-30% of Facebook[13] users are considered to be power users[14]; they are active users and participate heavily in core Facebook activities such as sending friend requests, tagging friends in photos, posting status updates, commenting, pressing the "like" button, and sending private messages. Due to this high level of activity, they are likely to be influencers and are targeted by online businesses to influence other users[15]. Yahoo! researchers [63] found that about 20,000 users (less than 0.05% of the user population) generated 50% of all tweets read and shared on Twitter[16]. Pinterest posts a list of most followed users[17] and social media marketers have written a book on "how to pin down more customers, crush your competition, and increase your company's revenue" by taking advantage of the Pinterest power user list [18]. The common theme, in the social media context, is that a small number of power users are able to influence a large number of other (non-power) users in the spread of ideas and opinions.

Power users in recommender systems are similar to those defined in the social media context. Power users in RSs have been referred to as users with a large number of ratings [20] as well as those that are able to influence the largest number of other users [14, 44, 3, 18].

[13] www.facebook.com
[14] http://www.pewinternet.org/Press-Releases/2012/Facebook-users.aspx
[15] http://mashable.com/2012/05/21/facebook-power-user-infographic/
[16] www.twitter.com
[17] http://pinterest.com/pinterestpower/most-followed-pinterest-users/
[18] http://www.powerofpinterest.com/resources/pinterest-power-user-list/

> Therefore, in this dissertation, *the term "power user" denotes users who have considerable influence over the recommendations presented to other users and can be characterized as the top x% of users in a dataset when ranked by influence over the recommendations given to other users.*

To measure influence, Rashid et al [44] used the number of prediction differences above a prediction threshold when a user is removed from the dataset, Goyal and Lakshmanan [18] used the number of users that had the prediction for a target item shifted sufficiently above a threshold so that the item appears in their top-N list, Anand and Griffiths [3] used MAE and coverage to evaluate various seed (influential user) selection algorithms, and Domingos and Richardson [14] used the expected lift in profit earned by influencing other users, recursively. Influence, for our purposes, is measured as the ability of a power user i to change (positively or negatively) the RS prediction of another user j, or for an power user attacker i's target item to appear in user j's top-k list.

These power users are of interest to system operators and marketers when launching a new product because a positive endorsement (high rating) can translate into product recommendations to a large number of users. This is known as market-based use of RS and has been previously promoted as a solution to the "cold-start" or "new item" problem wherein new items cannot be recommended to users because they have few or no ratings [14, 3]. At the core of every CF RS is a user-item matrix, containing

user ratings for items; the user-user relationships and similarity matrix derived from the user-item matrix can, therefore, be viewed as a social network with users as nodes and nearest-neighbor relationships between users as edges. Social Network Analysis has well-known concepts that can be readily applied to RS, especially with regard to the levels of influence that some users have over others. The concept of Degree Centrality [57] specifies that nodes (users) who have more edges (connections) to other nodes may have advantages; high in-degree refers to nodes that many other nodes connect to and corresponds to high prominence, prestige, or popularity and high out-degree refers to nodes that connect to many other nodes and corresponds to high expansiveness. Also, some authors claim that "it is advantageous to be connected to those who have few options; power comes from being connected to those who are powerless" [4]. Collaborative relationships in recommender systems can be represented as a social network [43], where in-degree represents the number of contact lists a user appears in and out-degree indicates the number of users on a contact list that can be used to ask opinions or advice. A high in-degree indicates a higher level of trust in this user and that this user has more power because they can influence other users with their opinions; a high out-degree means that this user trusts the advice and opinions of others. Additionally, neighborhood characteristics and power user identification in recommender systems were analyzed in [27] as part of study on temporal social networks. Prior work in this area has determined that maximizing the spread of influence through a social network is an NP-hard problem to solve optimally [23, 18] and Rashid et al [44] proposed a technique to measure influence in a network of users and found it to be computationally expensive. To circumvent

these issues, heuristics have been used as power user selection methods to identify and select groups of influential users [18, 27]. In this dissertation, we follow the lead of [18, 27] and specify heuristic power user selection methods.

Studies such as [14, 3, 18] show that power users possess the influence to sufficiently impact a RS for "white hat" marketing purposes. The question is whether the influence these power users have could also be used for "black hat" purposes, i.e, what happens when attack user profiles look more like power user profiles? The potential for power user attacks exists and there are documented instances of power users who have gamed online systems, such as the July 2012 Wired article on Digg power users[19]. Our conjecture is that an influence-based attack model will enhance the effectiveness of RS attacks (from the attacker's perspective), i.e., attackers that can acquire the influence accorded to power users will have high impacts on RS recommendations. Thus, this dissertation posits a novel attack model known as the Power User Attack (PUA) that uses the concept of "power users" and their influence over other users. Our assertion is that attack user profiles, with the influential characteristics of power users, can have significant (negative) impacts on RS accuracy and robustness. We analyze various power user attack scenarios to determine the accuracy and robustness impacts on the RS in order to understand and mitigate these attacks. For clarity, the power user attack envisioned in this research is not about having hundreds or thousands of actual power users colluding to mount an attack, rather, it is about an attacker being able to generate a set of synthetic power user profiles that, when stealthily injected into an RS, can effectively bias the recommendations.

[19]http://www.wired.com/gadgetlab/2012/07/mklopez-digg-power-user-interview/

The *Research Gap* investigated in this dissertation can be summarized as follows: *Having the ability to generate a set of synthetic user profiles can leave systems vulnerable to exploitation from more subtle, yet powerful, attacks based on influential power user characteristics and properties.* This vector of attack remains an open question in RS robustness research.

The central thesis of this work is that attacks on recommender systems using influence-based methods of generating user profiles (rather than previously-studied statistical "average" methods) are able to effectively bias recommendations to suit the attacker's objectives and negatively impact system accuracy and robustness measures. Given that power users are known to have strong influence over large groups of users, attackers that can acquire the influence accorded to power users may have strong impacts on system recommendations. This gives rise to the main dissertation research questions (*DRQ*) that are addressed in the body of this dissertation (Chapters 5, 6, 7, 8, 9):

DRQ-1: Does the use of Social Network Analysis identify more influential Power Users than other methods?

DRQ-2: Will synthetic Power User profiles generated from power user characteristics retain the same level of influence of real Power Users?

DRQ-3: What happens to Recommender System accuracy and robustness when power users attack?

DRQ-4: Can a novel attack be crafted to achieve power user capability with strong influence and "low" cost of attack?

DRQ-5: What is the most effective method of mitigating power user attacks?

1.4 Dissertation Scope and Objectives

To address the problem of power user attacks on recommender systems, this dissertation covers the following topic areas:

1. Power User Selection and Evaluation of their Impact in RS: The following heuristic methods are described, analyzed, and evaluated in this dissertation:
   - In-Degree Centrality: Users with the highest user-user in-degree values are selected as power users.
   - Aggregated Similarity: Users with highest user-user similarity correlation values are selected as power users.
   - Number of Ratings: Users with the most number of ratings are selected as power users.

   The evaluation of power user selection techniques include analyses before and after a power user attack. *The objectives are to evaluate different techniques for power user selection in RS and to study alternative methods of evaluating power user selection in the context of power user attacks.*
2. Power User Characteristics and Generation of User Profiles: This research investigates the statistical characteristics of power users contained in the social graph within the RS, how they influence other users, how they differ statistically from typical users, how their influence can be used to modify RS recommendations, and how this data can be used to generate fake/synthetic attack user profiles that can be injected into the RS to impact recommendations. This work also specifies how power user profiles are different (or similar) to well known attack

signatures such as random, average, bandwagon, and segment attacks, among others. *The objectives are to model power user characteristics and to develop synthetic attack user profiles based on the power user model.*

3. Power User Attack Execution and Evaluation of their Impact in RS: This effort defines and executes the Power User Attack (PUA) model with simulated or synthetic power user attack profiles. This attack model is evaluated with respect to impacts to RS accuracy and robustness across multiple power user selection techniques, CF algorithms, datasets, and domains; trade-offs between accuracy and robustness measures are also analyzed. In addition, this dissertation investigates and evaluates the complementary Power Item Attack (PIA), where the power user profiles are populated with influential "power items" that are selected using the same techniques used to select power users. *The objectives are to develop new attack models based on the analysis of power user, and power item, characteristics and to study alternative methods of evaluating the impacts of power users and power items used to attack collaborative recommenders.*
4. Power User Attack Mitigation: While removing attack user profiles from recommendation calculations is a straightforward approach to eliminating the attacker's influence in a laboratory environment, in live RS environments this approach could also have unwanted side effects [32]. For instance, in cases where a legitimate power user is mistakenly identified as an attacker, the users that rely on that power user's neighborhood influence may be impacted and could lead to satisfaction issues. This dissertation analyzes the impacts on RS accuracy and robustness when power user attack profiles are removed from recommen-

dation calculations to mitigate the attack impacts as well as the impacts when power user influence is reduced by adjusting the similarity weighting during the prediction calculations. *The objectives are to determine more effective power user attack impact mitigation strategies compared to 100% removal of identified power users.*

1.5 Research Strategy and Hypotheses

1.5.1 Research Strategy

The research strategy adopted in this dissertation consists of a quantitative experimental design using system accuracy and robustness as the constructs. This work extends previous research [39, 26, 36, 9] on RS attack models, attack evaluation, and attack mitigation. This is also a data-driven and empirically-focused dissertation that evaluates the accuracy and robustness of recommender systems under attack using established metrics [20, 36, 9, 56]. The experimentation uses an Apache Mahout platform[20] and publicly-available research datasets in the movie and music domains (MovieLens[21] and Yahoo! Music[22], respectively).

The variables used in this dissertation include:

- Novel attack models: Power User Attack and Power Item Attack (see Chapters 5 and 7, respectively).
- Power user selection methods: In-Degree Centrality, Aggregated Similarity, and Number of Ratings (see Section 6.2).
- Collaborative filtering algorithms: User-based, Item-based, and SVD-based (see

[20] apache.mahout.org
[21] www.grouplens.org
[22] http://webscope.sandbox.yahoo.com/catalog.php?datatype=r

Section 2.1).

- Publicly-available domain datasets: MovieLens and Yahoo! Music.

To evaluate the experimental results, accuracy metrics (Mean Absolute Error) [56] and robustness metrics (Hit Ratio, Prediction Shift, Rank) [36] were used; descriptions of these measures are provided in Section 2.2. Hypotheses were developed for the experiments in this dissertation and were tested using statistical analysis.

1.5.2 Dissertation Hypotheses

In the body of this dissertation (Chapters 5, 6, 7, 8, 9), empirical analysis was conducted to answer research questions and to test specific hypotheses directly related to each of those experiments. In order to summarize the results of these various experiment hypotheses, the following dissertation hypotheses (DH) have been developed to answer the research questions posed in Section 1.3. Therefore, these dissertation hypotheses are general forms of the specific hypotheses used in the empirical analysis:

DH-1: The use of In-Degree Centrality to select a set of power users results in power users with higher influence than other selection techniques, across multiple datasets and domains.

DH-2: A significant percentage of synthetic user profiles generated from statistical characteristics of power users will be identified by selected power user selection techniques across multiple datasets and domains.

DH-3: Power user attack profiles generated from characteristics of InDegree-selected power users will result in more effective attacks (from the attacker's viewpoint) than attack profiles generated from characteristics of power users selected from other tech-

niques across CF algorithms, datasets, and domains.

DH-4: A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects on RS predictions and top-N lists of recommendations across multiple power user selection techniques, collaborative filtering algorithms, datasets, and domains.

DH-5: Reducing the influence of power users is a more effective and less impactful mitigation strategy than completely removing power users from the dataset.

The disposition of these dissertation hypotheses is provided in the Summary sections of Chapters 5, 6, 7, 8, 9 and in the Dissertation Summary Chapter 10. Furthermore, the dissertation research questions (DRQs) addressed by each of the dissertation hypotheses (DHs) can also be found in the Dissertation Summary Chapter 10.

1.6 Dissertation Contributions

The main contributions of this research are:

1. Power User Attack Model: This is a novel attack model based on influential power users. The model specifies how power users are selected from a dataset and how the power user profiles are configured for the attack. Different techniques for power user selection were evaluated and alternative methods of evaluating power user selection were analyzed in the context of power user attacks. See Chapters 5 and 6.
2. Power User Model: This model specifies the statistical characteristics of power users in sufficient detail so that synthetic power user attack profiles can be generated for attack purposes. This effort mainly involved characterizing power users according to their statistical properties and generating synthetic power

user profiles. The degree to which synthetic power user profiles resemble actual power user profiles was evaluated across multiple power user selection techniques. See Chapter 6.

3. Evaluation Approach for Power User Selection and Power User Attacks: The approach consists of metrics collected before and after the power user attack and is used to evaluate both the power user selection and the power user attack. A power user evaluation process that combines impacts to accuracy metrics before an attack and impacts to accuracy and robustness metrics after an attack was analyzed. The use of In-Degree Centrality to select a set of power users compared to other power user selection techniques was evaluated across multiple collaborative filtering algorithms, datasets, and domains. The degree to which power user attacks using synthetic profiles can impact RS recommendations across multiple power user selection techniques and collaborative filtering algorithms was also evaluated. See Chapters 5 and 6.
4. Power Item Attack Model and Power Item Model: The novel power item attack model uses synthetic power user profiles populated with power items in a novel attack configuration using multiple targets. The power item model describes how synthetic power users are generated using characteristics of influential (power) items. The power item attack was evaluated across multiple power user selection techniques and collaborative filtering algorithms. See Chapter 7.
5. Mitigation Approach for Power User Attacks: The approach is to reduce the impact of the power user attack without having to remove 100% of the power users because of the important role that power users play in maintaining a

higher level of recommender system accuracy. The approach reducing the influence of power users is a more effective and less impactful mitigation strategy than completely eliminating the influence of power users was evaluated across multiple power user selection techniques and collaborative filtering algorithms. See Chapter 8.

6. New Evaluation Metrics: Throughout this dissertation, several metrics were developed when evaluating accuracy and robustness measures. The AC metric discussed in Chapter 4 was used to show the trade-offs between accuracy and coverage when evaluating collaborative filtering algorithms, the NTPU and NNTPU metrics discussed in Chapter 7 were used to determine the effectiveness of the Power Item Attack within and between experiments, and the ARM metric discussed in Chapters 8 and 9 was used to evaluate the trade-offs between accuracy and robustness when evaluating power user attack mitigation strategies.

1.7 Dissertation Organization

This dissertation is organized as follows:

Chapter 1 is an introduction to the dissertation research effort, including the research problem, an example Recommender System attack, a discussion of research gaps, scope and objectives, research questions and hypotheses, contributions, publications related to this research, and organization of this dissertation.

Chapter 2 provides background on Collaborative Filtering Recommender Systems and Evaluation methods for Recommender Systems.

Chapter 3 provides background on Attacks on Collaborative Filtering Recommender

Systems.

Chapter 4 documents research [53] that describes and evaluates the changes made to Mahout, the recommender system platform used in this research, in order to make Mahout fit for our purposes. This research also shows how power users contribute to providing accuracy and robustness in RS using composite metrics, such as the novel AC measure [52] (see § 4.3) for accuracy and coverage.

Chapter 5 documents research [60, 51, 54] that describes the power user attack model and initial investigation relative to power user selection, power user attacks, and evaluation of power user selection methods and power user attacks against user-based, item-based, and SVD-based CF algorithms.

Chapter 6 documents research [61] that describes the power user model based on power user characteristics, the generation of synthetic attack user profiles based on the power user model, the evaluation of power user selection methods, and the execution and evaluation of the power user attack using synthetic power users against user-based and SVD-based CF algorithms.

Chapter 7 documents research [55] that describes a power user attack model using power items as well as the execution and evaluation of the power user attack using synthetic power users against user-based, item-based, and SVD-based CF algorithms.

Chapter 8 documents research [62] that describes and evaluates power user attack mitigation strategies using synthetic power users against the user-based CF algorithm for the movie domain.

Chapter 9 documents research experiments evaluating power user selection methods, execution and evaluation of the power user attack using synthetic power users, and

mitigation strategies for power user attacks against user-based recommender systems using music domain datasets.

Chapter 10 provides a summary of the results and contributions of this dissertation.

Appendix A provides the list of research papers published during the development of this dissertation.

Appenidx B contains a table of statistics for datasets used in this dissertation.

CHAPTER 2: COLLABORATIVE RECOMMENDER SYSTEMS

Collaborative filtering employs user profiles that typically consist of item ratings, often on a five point Likert scale. So, an initial user rating profile for a movie recommender might consist of a few ratings, such as: Avatar = 4; The Muppets = 5; Hugo = 2. Arranging data on dimensions of m users and n items gives the traditional CF user-item matrix data structure. In order to generate predictions, user-based and item-based CF recommender systems follow a consistent process:

- Establish similarity between users (user-based) or items (item-based)
- Weight the similarities to emphasize users (or items) that are most influential in establishing similarity
- Compute a prediction that takes into account the users' (or items') ratings as well as their similarities.

In addition to the user-item matrix, SVD-based CF recommenders require knowledge of the number of latent features and use matrix factorization techniques to compute predictions.

2.1 Collaborative Filtering Algorithms

2.1.1 User-Based Algorithms

In user-based systems, recommendations made to the active user are based on what other similar users have liked in the past; similar users are selected based on statistical methods and form a neighborhood of rating influence for the active user. For user-

based CF systems, similarities between users often employ Pearson Correlation, as described in [45] and [19]. Similarity weighting is then used to rank similarity values according to the number of co-rated items between two users; similarities calculated from user pairs with a large number of co-rated items will be ranked higher (i.e., given a higher weight). Other parameters include similarity thresholding (ignoring users with similarity below the threshold value) and kNN neighborhood size (bounding the number of users comprising the neighborhood).

Two popular methods are often used for prediction calculation: weighted and mean-centered. The weighted prediction method [13] ensures that predicted ratings are within an allowable range (e.g., between 1.0 and 5.0). After similarities are calculated, the k most similar users that have rated the target item are selected as the neighborhood. After identifying a neighborhood, a prediction is computed for a target item i and target user u as follows:

$$p_{u,i} = \frac{\sum_{v \epsilon V} sim_{u,v} * r_{v,i}}{\sum_{v \epsilon V} \mid sim_{u,v} \mid} \tag{1}$$

where V is the set of k similar users and $r_{v,i}$ is the rating of those users who have rated item i, and $sim_{u,v}$ is the mean-adjusted Pearson correlation coefficient described above. Rating predictions calculated based on *zero or one* co-rated items are typically discarded as one co-rated item is insufficient to provide a reliable prediction. The mean-centered prediction method, as documented in [45, 19, 13], is computed for a target item i and target user u as follows:

$$p_{u,i} = \overline{r_u} + \frac{\sum_{v \epsilon V} sim_{u,v}(r_{v,i} - \overline{r_v})}{\sum_{v \epsilon V} \mid sim_{u,v} \mid} \tag{2}$$

where V is the set of k similar users who have rated item i, $r_{v,i}$ is the rating of those users who have rated item i, $\overline{r_u}$ is the average rating for the target user u over all rated items, $\overline{r_v}$ is the average rating for user v over all co-rated items, and $sim_{u,v}$ is the mean-adjusted Pearson correlation coefficient described above. This technique is used to compensate for the fact that different users may use different rating values to quantify the same level of satisfaction for an item.

2.1.2 Item-Based Algorithms

In item-based systems, recommendations made to the active user are based on ratings of similar items within the active user's profile. Similar items are determined based on statistical methods across all the users in the user-item matrix. For item-based CF systems, similarities between items are determined using either the Pearson Correlation coefficient or the Adjusted Cosine Similarity measure [47]. Similar to Pearson Correlation, Adjusted Cosine subtracts the corresponding user average from each co-rated pair to take into account the differences in rating scale between different users. As with user-based approaches, similarity weighting and thresholding may be employed.

Popular prediction calculation approaches again include weighted and mean-centered. The weighted prediction method [47] ensures that the predicted ratings are within allowable range. The prediction of item i for user u is made by computing the sum of the ratings given by user u on the items similar to item i. Each rating is then weighted by the corresponding similarity $s(i, j)$ between items i and j.

$$p_{u,i} = \frac{\sum_{j \epsilon allsimilaritems}(s_{i,j} * r_{u,j})}{\sum_{j \epsilon allsimilaritems}(\mid sim_{i,j} \mid)} \tag{3}$$

This method computes the prediction on an item i for a user u by computing the sum of the ratings given by the user on the items similar to i. Each rating is weighted by the corresponding similarity $s_{i,j}$ between items i and j. This approach captures how the active user rates the similar items. Also, rating predictions calculated based on *zero or one* co-rated items are typically discarded as one co-rated item is insufficient to provide a reliable prediction. The mean-centered prediction method [13], is computed for a target item i and target user u as follows:

$$p_{u,i} = \overline{r_i} + \frac{\sum_{j \epsilon N_u(i)} sim_{i,j}(r_{u,j} - \overline{r_j})}{\sum_{j \epsilon N_u(i)} \mid sim_{i,j} \mid} \tag{4}$$

where $N_u(i)$ is the set of items rated by user u most similar to item i, $r_{u,j}$ is u's rating of item j, $\overline{r_j}$ is the average rating for item j over all users who rated item j, $\overline{r_i}$ is the average rating for target item i, and $sim_{i,j}$ is the similarity measure.

2.1.3 Singular Value Decomposition (SVD)

Item ratings provided by users are influenced by a set of (latent) factors or features specific to a domain. For example, in the movie domain latent factors may include genre, actors, directors, etc. Users tend to give high ratings to certain movies with actors/actresses they like or to action movies if that is the genre they prefer. Although these factors are not always obvious, the goal is to infer these latent factors using mathematical techniques known as matrix factorization. SVD is a matrix factorization technique used in RS [25]. In SVD-based systems, users and items are mapped into the same latent factor space; this latent space is then used to develop recommendations for the active user that are based on latent factors automatically inferred from user ratings, i.e., each rating is estimated as the dot product of the user

feature vector and the item feature vector. Moreover, recommender systems with many users (rows) and items (columns), consist of a dataset with factors that define a high-dimensional space and have sparse information in that space. The data matrix is sparse because, typically, most of the users have rated a small percentage of the items available. High dimensional data is difficult to work with because adding more factors can increase the noise and the error and there aren't enough observations to get good estimates or predictions. In order to deal this problem, dimensionality reduction techniques such as SVD have been applied [48, 25, 2].

2.2 Evaluating Recommender Systems

A comprehensive set of guidelines for evaluating recommender systems was provided by Herlocker et al [20] and recently in Shani and Gunawardana [56]. While the number of measures for evaluating has expanded over the years and includes metrics for accuracy, coverage, robustness, confidence, trust, novelty, serendipity, diversity, scalability, and others, the focus in this dissertation will be on the measurement of accuracy, coverage, and robustness of recommender systems under attack.

2.2.1 Evaluating Accuracy and Coverage

Measures of accuracy include Mean Absolute Error and Root Mean Squared Error; they are used to measure prediction accuracy of a recommender system.

Mean Absolute Error is calculated as follows,

$$MAE = \frac{\sum_{i=1}^{n} \mid PredictedRating_i - ActualRating_i \mid}{n} \tag{5}$$

where n is the total number of ratings predicted in the test run.

Another popular metric used in evaluating accuracy of predicted ratings is Root

Mean Squared Error (RMSE). This metric penalizes large errors, e.g., a prediction that is off by 2 points is more than twice as "bad" as one that is off by 1 point. RMSE is calculated as follows,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(PredictedlRating_i - ActualRating_i)^2}{n}} \tag{6}$$

where n is the total number of ratings predicted in the test run.

As suggested in [20], the easiest way to measure coverage is to select a random sample of user-item pairs, ask for a prediction for each pair, and measure the percentage for which a prediction was provided. To calculate coverage, compute the total number of rating predictions requested that are unable to be calculated as well as the total of number of rating predictions requested that are actually calculated; the sum of these two numbers is the total number of ratings requested. Coverage is calculated as follows:

$$Coverage = \frac{Total\#RatingsCalculated}{Total\#RatingsRequested} \tag{7}$$

2.2.2 Evaluating Robustness

Robustness metrics such as Hit Ratio and Prediction Shift have been discussed in detail in [36] and [9]. These were used to measure the success of the attack (from the attacker's standpoint) such that a high Hit Ratio or a high Prediction Shift meant that the attack succeeded in changing the recommendations produced by the CF system. The Prediction Shift metric is defined as follows: Let U_T and I_T be the sets of users and items, respectively, in the test data. For each user-item pair (u, i), the Prediction Shift denoted by $\Delta_{u,i}$ can be measured as $\Delta_{u,i} = p'_{u,i} - p_{u,i}$ where p and

p' are the pre-and post-attack predictions, respectively. A positive value means that the attack has succeeded in making the pushed target item more positively rated. The Average Prediction Shift for a target item i over all users can be computed as $\Delta_i = \frac{\sum_{u \epsilon U_T} \Delta_{u,i}}{|U_T|}$ and the Average Prediction Shift for all items tested can be computed as,

$$\overline{\Delta} = \frac{\sum_{i \epsilon I_T} \Delta_i}{\mid I_T \mid}. \tag{8}$$

Although prediction shift is a good indicator that an attack has successfully (from the attacker's standpoint) made a pushed item more desirable, or a nuked item less desirable, the item may still not make it into the top N list of recommendations presented to the user. So, another metric, Hit Ratio, was developed to indicate the percentage of users that have the target item in their top N list of recommendations. 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. Hit Ratio for a target item i is given by $HitRatio_i = \frac{\sum_{u \epsilon U_T} H_{u,i}}{|U_T|}$. The Average Hit Ratio can be calculated as,

$$\overline{HitRatio} = \frac{\sum_{i \epsilon I_T} HitRatio_i}{\mid I_T \mid}. \tag{9}$$

Average Rank [36], in the robustness context, is a measure that indicates the relative position of a target item i in a top-N list of recommendations produced after an attack. Let T_u be the set of predicted ratings in a top-N list for user u, and let $Rank_{ui}$ denote the ordinal position of target item i in the set T_u that is sorted in descending (highest to lowest) order based on the predicted rating value. The Average Rank for target item i, therefore, is the sum of the $Rank_{ui}$ over all users u

divided by the total number of users, i.e.,

$$\overline{Rank_i} = \frac{\sum_{u \epsilon U} Rank_{ui}}{\mid U \mid}. \tag{10}$$

2.2.3 Evaluating Recommender System Platforms

Revisiting evaluation in the context of recommender platforms has received recent attention in the thorough evaluation of the LensKit platform using previously tested collaborative filtering algorithms and metrics, as reported in [15]. A comprehensive set of guidelines for evaluating recommender systems was provided by Herlocker et al [20]; these guidelines highlight the use of evaluation metrics such as accuracy and coverage and suggest the need for an ideal "general coverage metric" that would combine coverage with accuracy to yield an overall "practical accuracy" measure. Many of these evaluation metrics and techniques have also been covered recently in [56].

Recommender system research has been primarily concerned with improving recommendation accuracy [29]; however, other metrics such as coverage [49, 17] and also novelty and serendipity [20, 16] have been deemed necessary because accuracy alone is not sufficient to properly evaluate the system. Mcnee et al [29] states that recommendations that are most accurate according to the standard metrics are sometimes not the most useful to users and outlines a more user-centric approach to evaluation. The interplay between accuracy and other metrics such as coverage and serendipity creates trade-offs for recommender system implementers and this has been widely discussed in the literature, e.g., see [17, 16] and our previous work discussing trade-offs between accuracy and robustness [53].

CHAPTER 3: ATTACKS ON RECOMMENDER SYSTEMS

Attempts to influence recommender system results by providing false ratings feedback are known as "shilling attacks" [26], or "profile injection attacks" [36]. A user rating profile contains the set of ratings a user has made using the recommender system. Attacks are used to either promote ("push") a target item by setting the rating to the maximum value or demote ("nuke") a target item by setting the rating to the minimum value; furthermore, attackers will submit one or more user profiles containing item ratings (called attack profiles) that push or nuke a specific item. Research in attacks on recommender systems started in 2002 [39] and has continued to be studied, especially in the areas of attack detection and improvements in algorithm robustness [26, 38, 40, 10, 8, 36, 31].

3.1 Attack User Profile

A user profile contains the ratings data that has been entered by a specific user; please refer to Table 1. Attack user profiles contain rating data consisting of the following items [35, 36, 59]:

1. *Ratings for selected items (IS)*, usually with particular characteristics determined by the attacker. The set of selected items represents a small group of items that have been selected because of their association with the target item (or a targeted segment of users). For some attacks, this set is empty.
2. *Ratings for filler items (IF)*, are usually set randomly according to some distri-

Table 1: Elements of an attack user profile

Attack User Profile			
Selected Items $IS_{1,..s}$	Filler Items $IF_{1,..f}$	Unrated Items $IU_{1,..u}$	Target Item IT_t
Ratings 1 to 5	Ratings 1 to 5	Ratings null	Rating 1 to 5

bution. On the other hand, the set of filler items represent a group of randomly selected items in the database which are assigned ratings within the attack user profile. Since the selected item set is small, the size of each profile (total number of ratings) is determined mostly by the size of the filler item set.

3. *Unrated items (IU)*. For some attacks, this set is empty.
4. *Rating for the target item (IT)*, is usually a single item that is typically set to the maximum or minimum rating depending on the attack intent.

Attack models, to be discussed below, can be defined by the methods by which they identify the selected items, the proportion of the remaining items that are used as filler items, and the way that specific ratings are assigned to each of these sets of items and to the target item. In experimental results, filler size is reported as a proportion of the size of the attack user profile (i.e., the set of all items in the attack user profile).

3.2 Attack Intent

The basic attack intents are carried out by adding attack user profiles to conduct an attack. The purpose of the attack can be [26, 39]:

1. Push attack: boost the ratings for a specific product or group of products (called a "push") so that they get recommended more often,

2. Nuke attack: to reduce the ratings for a specific product or group of products (called a "nuke") so that they get recommended less often.

It is also possible that a third intent could be added to this list, i.e., a purely malicious intent to disrupt the recommender system for nefarious, gratuitous or "entertainment" purposes; this attack intent has not been widely studied in shilling attack detection research.

3.3 Attack Models

The attack intent is carried out using one of several models defined in the literature [6, 7, 26, 36, 34, 59]; these models define the attack user profile data based on the attacker's knowledge of the underlying recommender system's algorithms, database, items, and users. Attack models include:

1. Average attack [26, 36]: In the average attack, each assigned rating for a filler item corresponds (either exactly or approximately) to the mean rating for that item, across the users in the database who have rated it. This attack user profile uses the individual mean for each item rather than the global mean (except for the target item). The attacker would have to have extensive knowledge of the ratings in the dataset in order to effectively implement this attack.
2. Random attack [26, 36]: The random attack user profiles consist of random ratings assigned to the filler items and a pre-specified rating assigned to the target item. In this attack model, the set of selected items is empty. Items not in the target set are rated randomly on a normal distribution with mean 3.6 and standard deviation 1.1. The attacker only needs a limited amount of

knowledge to implement this attack.

3. Bandwagon attack [36]: The goal of the bandwagon attack is to associate the attacked item with a small number of frequently rated items. This attack takes advantage of the Zipf's law distribution of popularity in consumer markets[23]. The attacker using this model will build attack user profiles containing those items that have high visibility; this attack only needs a limited amount of knowledge to implement. Such profiles will have a good probability of being similar to a large number of users, since the high visibility items are those that many users have rated. O'Mahony et al [40, 41] used the k-Nearest Neighbor user-based Collaborative Filtering algorithm for attack detection for early versions of the Bandwagon (Popular) attack model and the push and nuke attack intents; the authors coined the terms item *popularity* and item *likeability* to denote items (in this case, movies) that were frequently rated and highly rated, respectively.
4. Segment attack [35, 36]: The segment attack model is designed to push an item to a targeted group of users with known or easily predicted preferences. It is especially effective against item-based Collaborative Filtering algorithms and requires less knowledge on the part of the attacker than the Average attack model. O'Mahony et al [40] use the k-Nearest Neighbor user-based Collaborative Filtering algorithm for attack detection for early versions of the Segment (aka Probe) attack model and the push and nuke attack intents.
5. Love/Hate Attack [36]: This is a simple attack with no knowledge requirements.

[23]Zipf's law distribution of popularity in consumer markets: a small number of items, bestseller books for example, will receive the majority of attention and also ratings.

The attack consists of attack profiles in which the target item is given the minimum rating value while other ratings in the filler item set are the maximum rating value. A variation of this attack can also be used as a push attack by switching the roles of minimum and maximum values.

6. Average over Popular (AOP) [21]: Chooses filler items from the top x% of most popular items, rather than from the entire catalog of items, where x is chosen to ensure that the profiles are undetectable by the PCA detector proposed in that study.
7. Obfuscation Attack [58, 21]: This is an attack that deviates from other known attack models, mentioned in this section, to avoid detection. Three types of deviations from the known attack models have been proposed by Williams et al [58]:
    - Noise Injection: involves adding a Gaussian distributed random number multiplied by a constant governing the amount of noise to be added to each rating within a set of attack user profile items. This noise can be used to blur the profile signatures that are often associated with known attack models.
    - User Shifting: involves incrementing or decrementing (shifting) all ratings for a subset of items per attack user profile in order to reduce the similarity between attack users.
    - Target Shifting: for a push attack, is simply shifting the rating given to the target item from the maximum rating to a rating one step lower, or in the case of nuke attacks increasing the target rating to one step above the

lowest rating.

Random and Average attack models represent, respectively, the low and high ends of the dataset content knowledge spectrum. Random attacks require very limited knowledge of the ratings distribution of the items in the dataset and the only requirement is that the attacker know the overall average rating and standard deviation which can be obtained by sampling the system. On the other hand, Average attacks require knowledge of the average rating for each item in the dataset, which is much more difficult, if not impossible, knowledge to obtain.

It should be noted that none of these attacks explicitly consider the impact that user influence can have on recommendations. Thus, the *Research Gap* investigated in this dissertation can be summarized again as follows: *Having the ability to generate a set of synthetic user profiles can leave systems vulnerable to exploitation from more subtle, yet powerful, attacks based on influential power user characteristics and properties.* This vector of attack remains an open question in RS robustness research.

3.4 An Illustrative Example

In order to illustrate the various collaborative filtering algorithms and attacks on recommender systems, consider the matrix shown in Figure 2 that was also presented in Section 1.2. This matrix contains 8 legitimate users (Bob, Ted, Fred, Ginger, Jodie, Jill, Tom, Corey), 3 attack users (Alice, Axel, Alvin), and 6 movie items (Avengers, Titanic, Avatar, Twilight, Psycho, Alien), as adapted from [36]. It shows the ratings that users have given to the items on a scale of 1 (disliked) to 5 (like very much). The attackers have implemented a push attack on target item Alien, they

User Profiles	Avengers	Titanic	Avatar	Twilight	Psycho	Alien
Bob	5	2	3	3		?
Ted	2		4		4	1
Fred	3	1	3		1	2
Ginger	4	2	3	1		1
Jodie	3	3	2	1	3	1
Jill		3		1	2	
Tom	4	3		3	3	2
Corey		5		1	5	1
Alice	5		3		2	5
Axel	5	1	4		2	5
Alvin	5	2	2	2		5

Figure 2: Example – Push attack on a target item

have attempted to increase the similarity between Avengers and Alien with ratings of 5, and they have also attempted to increase the similarity between the attackers and Bob with ratings of 5. In this example, Bob will request a recommendation from the recommender system for the movie item Alien.

Bob, Alien	User-Based Prediction	Item-Based Prediction	SVD-Based Prediction
Before Attack	2.00	2.01	1.83
After Alice	5.00	2.56	2.28
After Alice, Axel	5.00	2.51	2.66
After Alice, Axel, Alvin	5.00	2.45	2.98

Figure 3: Example – Recommender system predictions, before and after attacks

Figure 3 shows the results for Bob and Alien across three sequential attack profile injections. Before the attack, Alien would likely not be recommended to Bob given that the predicted rating is somewhere between 1.83 and 2.01, depending on the algorithm used for prediction. However, if all three attackers struck at the same time (After Alice, Axel, Alvin), the user-based algorithm would recommend Alien to Bob

with a predicted rating of 5, whereas the item-based and SVD-based algorithms would provide Bob with predicted ratings for Alien of 2.45 and 2.98, respectively. This result indicates that the attackers successfully raised the predicted rating from 2 to 5 for Alien in a user-based recommender system: Bob may take the advice and purchase the movie item only to be disappointed by the manipulated recommendation. This result also shows that the item-based and SVD-based algorithms appear to be more resistant to attack than the user-based approach and that Bob would likely not be recommended Alien since both item-based and SVD-based algorithms produce ratings that are below 3.

This attack can also be viewed temporally, i.e., the attackers do not all strike at the same time, rather, they spread the attack over time. Figure 3 shows the predicted ratings for Bob and Alien, by algorithm, after each attacker has perpetrated their respective push attack on the recommender system. Note that the Alice profile alone is sufficient to significantly impact the user-based prediction. Subsequent attacks by Axel and Alvin serve only to vary the item-based and SVD-based predictions; there is no further change to the user-based prediction of 5 points. In this temporal view, the results are the same as above, i.e., the item-based and SVD-based algorithms appear to be more resistant to attack than the user-based approach.

CHAPTER 4: RECOMMENDER PLATFORM EVALUATION

In order to conduct the research described in this dissertation, it was necessary to not only select a development and test platform, but also to customize that platform to make it fit for purpose. This chapter describes some of the key functional changes made and presents the evaluation of the platform both as it is provided "out of the box' and after the changes were implemented [53]. As part of this evaluation, trade-offs between the evaluation metrics, accuracy and coverage, are discussed and a combined metric is developed to address this measurement trade-off.

4.1 Introduction

Selecting a foundational platform is an important step in developing recommender systems for personal, research, or commercial purposes. This can be done in many different ways: the platform may be developed from the ground up, an existing recommender engine may be contracted (e.g., OracleAS Personalization [24]), code libraries can be adapted, or a platform may be selected and tailored to suit (e.g., LensKit[25], MymediaLite[26], Apache Mahout[27], etc.). In some cases, a combination of these approaches will be employed.

For many projects, and particularly in the research context, the ideal situation is to find an open-source platform with many active contributors that provides a

[24]http://docs.oracle.com/cd/B14099_19/bi.1012/b14052/intro.htm
[25]http://lenskit.grouplens.org/
[26]http://www.ismll.uni-hildesheim.de/mymedialite/
[27]http://mahout.apache.org

rich and varied set of recommender system functions that meets all or most of the baseline development requirements. Short of finding this ideal solution, some minor customization to an already existing system may be the best approach to meet the specific development requirements. Various libraries have been released to support the development of recommender systems for some time, but it is only relatively recently that larger scale, open-source platforms have become readily available. In the context of such platforms, evaluation tools are important both to verify and validate baseline platform functionality, as well as to provide support for testing new techniques and approaches developed on top of the platform. We have adopted Apache Mahout as an enabling platform for our research and have faced both of these issues in employing it as part of our work in collaborative filtering recommenders.

This chapter presents a case study of evaluation for recommender systems in Apache Mahout, focusing on metrics for accuracy and coverage. We have developed functional changes to the baseline Mahout collaborative filtering algorithms to meet our research purposes, and this chapter examines evaluation both from the standpoint of tools for baseline platform functionality, as well as for enhancements and new functionality. The objective of this case study is to evaluate these functional changes made to the platform by comparing the baseline collaborative filtering algorithms to the changed algorithms using well known measures of accuracy and coverage [20]. Our goal is not to validate algorithms that have already been tested previously, but to assess whether, and to what extent, the functional enhancements have improved the accuracy and coverage performance of the baseline out-of-the-box Mahout platform. Given the interplay between accuracy and coverage in this context, we developed a unified metric

to assess accuracy vs. coverage trade-offs when evaluating functional changes made to Mahout's collaborative filtering algorithms.

4.2 Selecting Apache Mahout

To support our research in collaborative filtering, several recommender system platforms were surveyed, including LensKit, easyrec[28], and MymediaLite. We selected Mahout because it provides many of the desired characteristics required for a recommender development workbench platform. Mahout is a production-level, open-source, system and consists of a wide range of applications that are useful for a recommender system developer: collaborative filtering algorithms, data clustering, and data classification. Mahout is also highly scalable and is able to support distributed processing of large data sets across clusters of computers using Hadoop[29]. Mahout recommenders support various similarity and neighborhood formation calculations, recommendation prediction algorithms include user-based, item-based, SlopeOne and Singular Value Decomposition (SVD), and it also incorporates Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) evaluation methods. Mahout is readily extensible and provides a wide range of Java classes for customization. As an open-source project, the Mahout developer/contributor community is very active; the Mahout wiki also provides a list of developers and a list of websites that have implemented Mahout[30].

[28] http://easyrec.org/
[29] http://hadoop.apache.org/
[30] https://cwiki.apache.org/MAHOUT/mahout-wiki.html

4.2.1 Uncovering Mahout Details

Although Mahout is rich in documentation, there are implementation details on *how Mahout works* that could only be understood by looking at the source code. Thus, for clarity in evaluation, we needed to verify the implementation of baseline platform functionality. The following describes some of these details for Mahout 0.4 'out-of-the-box':

Similarity Weighting – Mahout implements the classic Pearson Correlation [45, 19] similarity coefficient. Similarity weighting is supported in Mahout and consists of the following method:

```
scaleFactor = 1.0 - count / (num + 1);
if (result < 0.0)
  result = -1.0 + scaleFactor * (1.0 + result);
else
  result = 1.0 - scaleFactor * (1.0 - result);
```

where *count* is the number of co-rated items between two users, *num* is the number of items in the dataset, and *result* is the calculated similarity coefficient.

User-Based Prediction Algorithm – Mahout implements a Weighted Average prediction method similar to the approach described in [13], except that Mahout does *not* take the absolute value of the individual similarities in the denominator, however, it does ensure that the predicted ratings are within the allowable range, e.g., between 1.0 and 5.0.

Item-Based Prediction Algorithm – Mahout implements a Weighted Average predic-

tion method. This approach is similar to the algorithm in [47], except that Mahout does *not* take the absolute value of the individual similarities in the denominator, however, it does ensure that the predicted ratings are within the allowable range, e.g., between 1.0 and 5.0. Also, Mahout does not provide support for neighborhood formation, e.g., similarity thresholding, for item-based prediction.

Accuracy Evaluation calculation – Mahout executes the recommender system evaluator specified at run time (MAE or RMSE) and implements traditional techniques found in [20, 56]. For MAE, this would be,

$$MAE = \frac{\sum_{i=1}^{n} \mid ActualRating_i - PredictedRating_i \mid}{n} \tag{11}$$

where n is the total number of ratings predicted in the test run.

4.2.2 Making Mahout Fit for Purpose

Through personal email communication with one of the Mahout developers, we were informed that Mahout intended to provide *basic* rating prediction and similarity weighting capabilities for its recommenders and that it would be up to developers to provide more elaborate approaches. Several changes were made to the prediction algorithms and the similarity weighting techniques for both the user-based and item-based recommenders in order to meet our specific requirements and to match the best practices found in the literature, as follows:

Similarity weighting – Defined as Significance Weighting in [19], this consists of the following method:

scaleFactor = count/50.0;

if (scaleFactor > 1.0) scaleFactor = 1.0;

*result = scaleFactor * result;*

where *count* is the number of co-rated items between two users, and *result* is the calculated similarity coefficient.

User-user mean-centered prediction – After identifying a neighborhood of similar users, a prediction, as documented in [45, 19, 13], is computed for a target item i and target user u as follows:

$$p_{u,i} = \overline{r_u} + \frac{\sum_{v \epsilon V} sim_{u,v}(r_{v,i} - \overline{r_v})}{\sum_{v \epsilon V} \mid sim_{u,v} \mid} \tag{12}$$

where V is the set of k similar users who have rated item i, $r_{v,i}$ is the rating of those users who have rated item i, $\overline{r_u}$ is the average rating for the target user u over all rated items, $\overline{r_v}$ is the average rating for user v over all co-rated items, and $sim_{u,v}$ is the Pearson correlation coefficient.

Item-item mean-centered prediction – A prediction, as documented in [13], is computed for a target item i and target user u as follows:

$$p_{u,i} = \overline{r_i} + \frac{\sum_{j \epsilon N_u(i)} sim_{i,j}(r_{u,j} - \overline{r_j})}{\sum_{j \epsilon N_u(i)} \mid sim_{i,j} \mid} \tag{13}$$

where $N_u(i)$ is the set of items rated by user u most similar to item i, $r_{u,j}$ is u's rating of item j, $\overline{r_j}$ is the average rating for item j over all users who rated item j, $\overline{r_i}$ is the average rating for target item i, and $sim_{i,j}$ is the similarity measure.

Item-item similarity thresholding – This method was added to Mahout and used in conjunction with the item-item mean-centered prediction described above. Similarity thresholding, as described in [19], defines a level of similarity that is required for two items to be considered similar for purposes of making a recommendation prediction;

item-item similarities that are less than the threshold are not used in the prediction calculation.

Coverage and combined accuracy/coverage metric – As suggested in [20], the easiest way to measure coverage is to select a random sample of user-item pairs, ask for a prediction for each pair, and measure the percentage for which a prediction was provided. To calculate coverage, code changes were made to Mahout to provide, for each test run, the total number of rating predictions requested that were unable to be calculated as well as the total of number of rating predictions requested that were actually calculated; the sum of these two numbers is the total number of ratings requested. Coverage was calculated as follows:

$$Coverage = \frac{Total\#RatingsCalculated}{Total\#RatingsRequested} \tag{14}$$

Code changes were also made to calculate a combined accuracy and coverage metric as defined in Section 4.3.

4.3 Accuracy and Coverage Metric

The metrics selected for this case study, accuracy and coverage, were chosen because they are fundamental to the utility of a recommender system [49, 20]. Although other metrics such as novelty and serendipity can, and should, be used in conjunction with accuracy and coverage, our objective was to evaluate the very basic requirements of a recommender system. Our implementation of coverage, referred to as prediction coverage in [20], measures the percentage of a dataset for which the recommender system is able to provide predictions. High coverage would indicate that the recommender system is able to provide predictions for a large number of items and is

considered to be a desirable characteristic of the recommender system [20]. A combination of high accuracy (low error rate) and high coverage are indeed desirable by users and system operators because it improves the utility or usefulness of the system from a user standpoint [49, 20].

Many collaborative filtering recommenders have a default value for predicting ratings for low-coverage situations, e.g., they provide average ratings. However, average ratings are not personalized and this may lead to user dissatisfaction with the recommendations provided. Marketing studies indicate that personalization is valuable to users and providers of e-commerce applications. Consumer perception of value for a personalized web experience ranked highly compared to other channels such as web ads, Facebook/Twitter, mobile ads, etc.; value was measured in terms of relevance, information accuracy, and memorability of experience.[31] In a survey of 120 marketers, 84% report that personalization impacts customer retention and loyalty.[32] Among more than 1,100 consumers surveyed for opinions on their shopping and browsing experiences, 40% said they buy more from retailers that personalize their shopping experience across channels.[33] And in a longitudinal experiment of how consumers value online personalization [11], results indicated that personalized recommendations led to more clicks than random suggestions and that a positive attitude towards personalization enhanced the consumer's attitude towards the web site. Therefore, it is important that recommender systems provide not only accurate recommenda-

[31] http://www.marketingcharts.com/online/marketers-value-personalization-of-the-web-experience-what-about-consumers-37772/

[32] http://www.exacttarget.com/company/newsroom/2014/08/independent-research-reveals-personalizing-customer-journeys-impacts

[33] http://www.marketingprofs.com/charts/2013/10235/personalized-marketing-drives-buyer-readiness-and-sales

tions, they also need to have a high degree of coverage to provide a more personalized experience.

What constitutes 'good' accuracy or coverage, however, has not been well defined in the literature: studies such as [49, 17, 19] and many others, endeavor to maximize accuracy (achieve lowest possible value) and/or coverage (achieve highest possible value) and view these metrics on a relative basis, i.e., how much the metric has increased or decreased beyond a baseline value based on empirical results. Furthermore, the interplay between accuracy and coverage, i.e., coverage decreases as a function of accuracy [17, 16], creates a trade-off for recommender system implementers that has been discussed previously but not been developed thoroughly. Inspired by the suggestion in [20] to combine the coverage and accuracy measures to yield an overall "practical accuracy" measure for the recommender system, we developed a straightforward "AC Measure" that combines both accuracy and coverage into a single metric as follows:

$$AC_i = \frac{Accuracy_i}{Coverage_i}, \tag{15}$$

where i indicates the ith trial in an evaluation experiment.

The AC Measure simply adjusts (upward) the Accuracy according to the level of Coverage metrics found in an experimental trial and is agnostic to the accuracy metric used, e.g., MAE or RMSE. Using a family of curves for the Mean Absolute Error (MAE) accuracy metric, Figure 4 illustrates the relationship between accuracy, coverage, and the AC Measure. As an example, following the *"*$MAE : 0.5$*"* curve we see that at 100% coverage, the AC Measure is 0.5, and at 10% coverage, the AC Measure

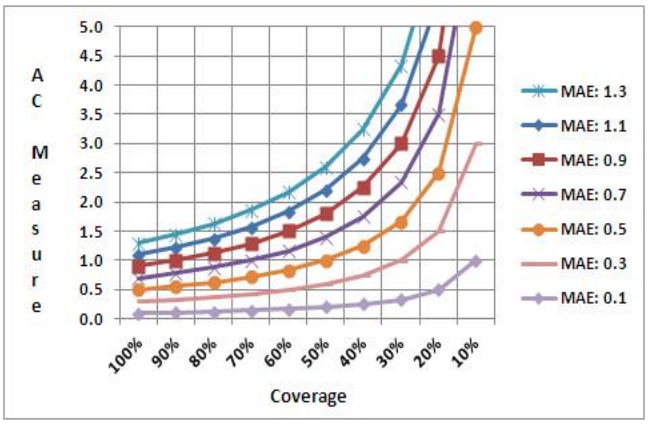

Figure 4: Illustration of the AC Measure

has increased to 5. The intuition behind this metric is that when the recommender system is able to provide predictions for a high percentage of items in the dataset, the accuracy metric more closely indicates the level of system performance; conversely, when the coverage is low, the accuracy metric is "penalized" and is adjusted upwards. We believe that the major benefit of the AC Measure is that it formulates a solution for addressing the trade-off between accuracy and coverage and can be used to create a ranked list of results (low to high) from multiple experimental trials to find the best (lowest) AC Measure for each set of test conditions. The simplified visualization of the combined AC Measure shown in Figure 4 is an additional benefit. For our evaluation purposes, the use of a combined metric was ideal in addressing the inherent trade-offs between accuracy and coverage, especially in the cases where accuracy is found to be high when coverage is low; we posit that the AC Measure will also be useful for other researchers performing evaluations using accuracy and coverage.

4.4 Experimental Design

The objective of this case study was to understand Mahout's baseline collaborative filtering algorithms and evaluate functional changes made to the platform using accuracy and coverage metrics. The main intent of making functional changes to Mahout recommender algorithms was to bring the Mahout algorithms in line with best practices found in the literature. Therefore, the overall hypothesis to be tested in this case study was that the modified algorithms improve Mahout's 'out-of-the-box' prediction accuracy for both user-based and item-based recommenders while maintaining reasonable coverage.

4.4.1 Datasets and Algorithms

The data used in this case study were the MovieLens datasets downloaded from GroupLens Research[34]: the 100K dataset with 100,000 ratings for 1,682 movies and 943 users (referred to as ML100K in this case study) and the 10M dataset with 10,000,000 ratings for 10,681 movies and 69,878 users (referred to as ML10M in this case study). Ratings provided in these datasets consist of integer values between 1 (did not like) to 5 (liked very much).

For User-based (see §4.2.1), Mahout uses Pearson Correlation similarity (with and without similarity weighting), Neighborhood formation (similarity thresholding or kNN), and Weighted Average prediction. This was tested against a modified algorithm (see §4.2.2) consisting of Pearson Correlation similarity (with and without similarity weighting), Neighborhood formation (similarity thresholding or kNN), and Mean-centered prediction. For Item-based (see §4.2.1), Mahout uses Pearson Correla-

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

tion similarity (with and without similarity weighting), no Neighborhood formation, and Weighted Average prediction. This was tested against a modified algorithm (see §4.2.2) consisting of the similarity measure (with and without similarity weighting), Neighborhood formation (similarity thresholding), and Mean-centered prediction.

4.4.2 Test Cases

In order to test the overall hypothesis, the following test cases were developed and executed for both user-based and item-based recommenders using the ML100K and ML10M datasets:

1. Mahout Prediction, No weighting
2. Mahout Prediction, Mahout weighted
3. Mahout Prediction, Significance weighted
4. Mean-Centered Prediction, No weighting
5. Mean-Centered Prediction, Mahout weighted
6. Mean-Centered Prediction, Significance weighted

4.4.3 Accuracy and Coverage Metrics

We used Mahout's MAE evaluator to measure the accuracy of the rating predictions. For prediction coverage, we used dataset training data to estimate the rating predictions for the test set; the random sample of user-item pairs in our testing was 30K pairs for ML100K and 25K pairs for ML10M (see §4.2.2). AC Measures were calculated for all test cases.

4.4.4 Dataset Partitioning

The Mahout evaluator creates holdout [35] partitions according to a set of run-time parameters. For the tests using the ML100K dataset, the training set was 70% of the data, the test set was 30% of the data, and 100% of the user data was used; a total of 30K rating predictions from 943 users were requested for each test set. For the tests using the ML10M dataset, the training set was 95% of the data, the test set was 5% of the data, and 5% of the user data was used; a total of 25K rating predictions from 3180 users were requested for each test set.

4.4.5 Test Variations

Various similarity thresholds and kNN neighborhood sizes were executed for each test case in order to understand and evaluate the corresponding behavior of the recommenders. For User-based recommender testing, similarity thresholds of 0.0, 0.1, 0.3, 0.5, and 0.7 and kNN neighborhood sizes of 600, 400, 200, 100, 50, 20, 10, 5, and 2 were tested. For Item-based recommender testing, in addition to using no similarity thresholding, similarity thresholds of 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, and 0.7 were tested.

4.5 Results and Discussion

4.5.1 ML10M Results

Figures 5 and 6 show the results of test cases 1 through 6 for user and item-based algorithms, respectively[36]. The key results of the experiment, for both user-based

[35]Holdout is a method that splits a dataset into two parts, a training set and a test set, and the partitioning is performed by randomly selecting some ratings from all, or some, of the users. The selected ratings constitute the test set, while the remaining ones are the training set.

[36]The following curves are superimposed over each other because the values are very similar: MAE results for mean-centered prediction (no weighting and Mahout weighted), MAE results for

and item-based algorithms unless otherwise noted, were as follows:

1. MAE for mean-centered prediction with significance weighting is a significant improvement ($p<0.01$) over MAE for Mahout prediction, regardless of weighting, across similarity thresholds (except item-based at similarity threshold of 0.7) and kNN neighborhood sizes (except user-based at kNN of 2, not shown).

2. Mahout similarity weighting does not significantly improve ($p<0.01$) Mahout prediction MAE over prediction with no similarity weighting (except Mahout prediction for user-based and item-based at a similarity threshold of 0.4, not shown). This would indicate that Mahout similarity weighting is not very effective as a weighting technique, especially as compared to significance weighting.

4.5.2 ML100K Results

The results and trend lines for the ML100K experiment are similar to ML10M. The key results, for both user-based and item-based algorithms unless otherwise noted, were:

1. MAE for mean-centered prediction with significance weighting is a significant improvement ($p<0.01$) over MAE for Mahout prediction, regardless of weighting, across similarity thresholds and kNN neighborhood sizes (except user-based at kNN of 400).

2. Mahout similarity weighting does not significantly improve ($p<0.01$) Mahout prediction MAE over prediction with no similarity weighting (except Mahout prediction for user-based and item-based at a similarity threshold of 0.4).

Mahout prediction (No weighting and Mahout weighted), Coverage results for Mahout prediction and mean-centered prediction (No weighting and Mahout weighted), Coverage results for Mahout prediction and mean-centered prediction (both Significance weighted).

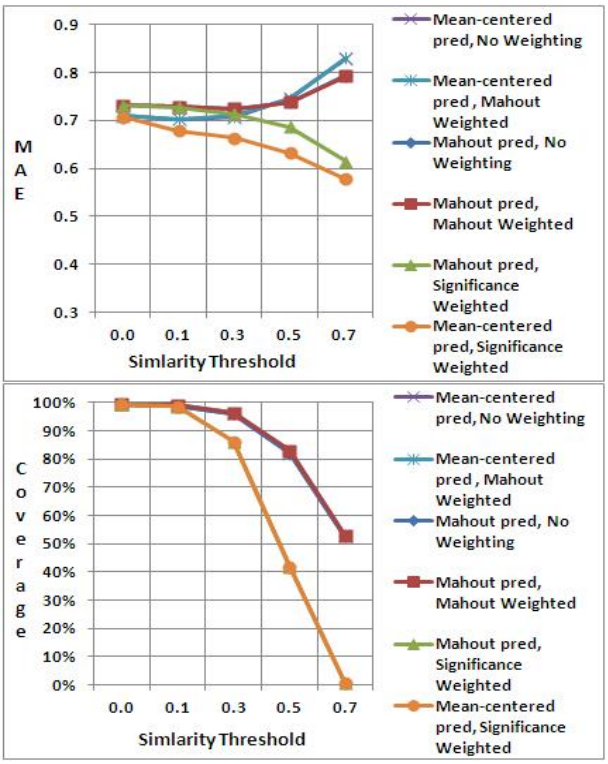

Figure 5: User-based Mahout recommender results for ML10M, Test cases 1 - 6

4.5.3 Discussion

As hypothesized, results for both of the ML100K and ML10M experiments show significant improvements in MAE using the mean-centered prediction algorithm with significance weighting compared to the Mahout baseline prediction algorithm. However, when coverage is considered, the "best" MAE results may need a second look. Can an MAE of 0.5 or less be considered "good" when the associated coverage is in the single digits? In this case, the recommender system may only be able to provide recommendations to a very small subset of its users and is a situation that must be avoided by system operators. To help address the accuracy vs. coverage trade-off, combined measures such as the AC Measure (Section 4.3), can help by con-

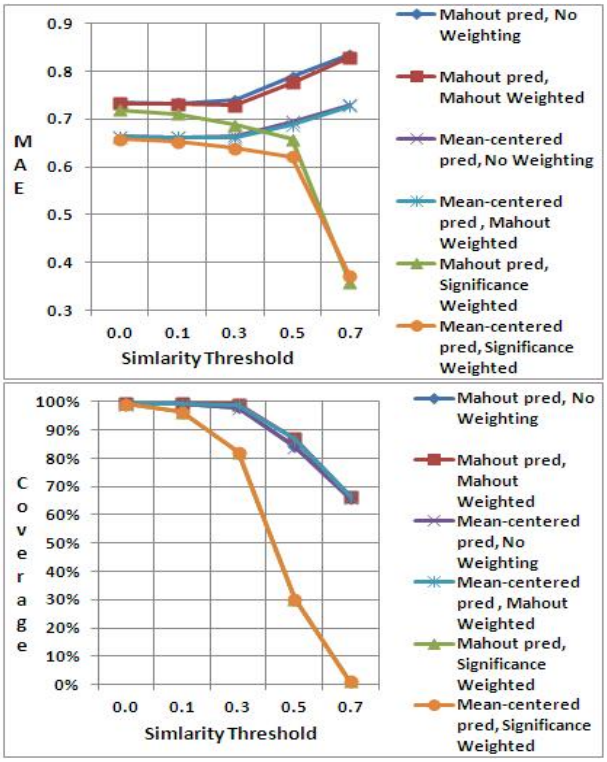

Figure 6: Item-based Mahout recommender results for ML10M, Test cases 1 - 6

sidering both accuracy and coverage simultaneously. For the ML10M experiment, we determined that the lowest MAE for the User-based algorithm using mean-centered prediction with significance weighting was 0.578 at a similarity threshold of 0.7 and coverage of 0.833%; the AC Measure for this result is calculated as 69.42. Similarly, the lowest MAE for the Item-based algorithm using mean-centered prediction with significance weighting was 0.371 at a similarity threshold of 0.7 and coverage of 1.02%; the AC Measure for this result is calculated as 36.32. In each of these cases, the exceedingly high values for the AC Measure indicate that these results are not very desirable in a recommender system.

Figures 7 and 8 show the AC Measure results for user and item-based algorithms using ML10M, respectively. Rather than show all 30 results for each algorithm (5

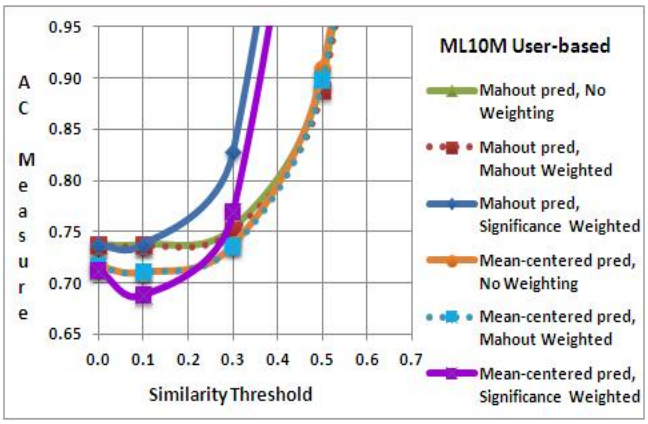

Figure 7: AC Measure for selected user-based results (lower is better)

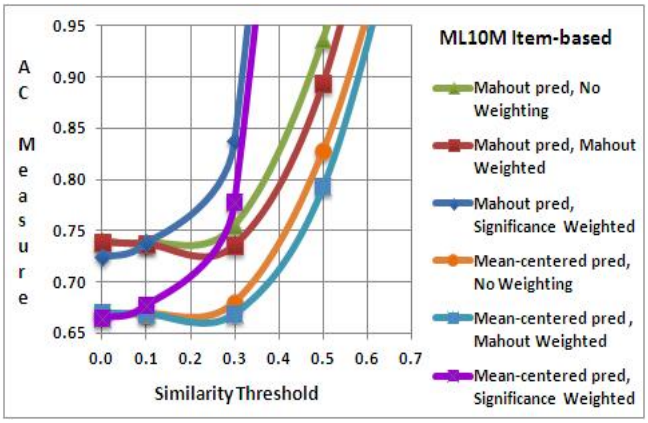

Figure 8: AC Measure for selected Item-based results (lower is better)

similarity thresholds x 2 prediction methods x 3 weighting types), we show only the results with calculated AC Measure values less than 1.0; therefore, the lowest MAE results reported above for user-based and item-based algorithms are clearly beyond the range of this chart. We found that the best combined accuracy/coverage results were found at higher levels of coverage and lower levels of similarity threshold, i.e., the best (lowest) AC Measure for user-based was 0.688 at a similarity threshold of 0.1 and for item-based was 0.665 at a similarity threshold of 0.0, both using mean-centered prediction and significance weighting. We can also see that, with few

exceptions, mean-centered prediction is improved over the Mahout prediction for the same similarity weighting and similarity threshold. We observed similar results using the ML100K dataset where the best (lowest) AC Measure for user-based was 0.765 and for item-based was 0.746, both at a similarity threshold of 0.0 and both using mean-centered prediction and significance weighting. These results demonstrate that the "best" MAE may not always be the lowest MAE, especially when coverage is also considered; furthermore, recommender system settings such as similarity weighting and neighborhood size also need to be considered during system evaluation.

Other observations of our experiments that match results reported in [19] and serve to validate our evaluation and increase our confidence in the results are: (a) In general, significance weighting improves prediction MAE, as compared to predictions using Mahout similarity weighting or no similarity weighting; (b) As the similarity threshold increases, MAE for mean-centered prediction with significance weighting improves and coverage degrades, whereas MAE and coverage both degrade for Mahout prediction with Mahout weighting; (c) Coverage decreases as neighborhood size decreases.

4.6 Summary of this Chapter

This case study of Mahout as a recommender system platform highlights evaluation considerations for developers and also shows how straightforward functional enhancements improve the performance of the baseline platform. We evaluated our changes against current Mahout functionality using accuracy and coverage metrics not only to assess baseline results, but also to provide a view of the trade-offs between accuracy and coverage resulting from using different recommender algorithms.

We reported cases where the lowest MAE accuracy results were not necessarily always the 'best' when coverage results were also considered, and we instrumented Mahout for a combined accuracy and coverage metric (AC Measure) to evaluate these trade-offs more directly. We believe that this case study will provide useful guidance in using Mahout as a recommender platform, and that our combined measure will prove useful in evaluating algorithm changes for the inherent trade-offs between accuracy and coverage.

In the context of this dissertation, the objective of this case study was to evaluate and validate the Mahout platform for our research purposes. The focus of the changes discussed here was to expand the functionality of the user-based and item-based collaborative filtering algorithms since they are the most popular and widely-used algorithms; analysis of the platform's SVD-based algorithms is provided in Chapters 5 and 6. Also, the work presented in this chapter does not consider attacks on recommender systems, therefore, robustness metrics are not considered as an evaluation metric; however, our research on attacks on recommender systems does consider robustness metrics and details are provided in Chapters 5, 6, 7, 8, 9.

CHAPTER 5: POWER USER SELECTION EVALUATION

5.1 Introduction

In a Collaborative Filtering (CF) Recommender System (RS) context, *power users* are those who can exert considerable influence over the recommendations presented to other users. Previous research has indicated that power users can have major impacts on RS ratings predictions and top-N recommendations lists, especially when the underlying RS algorithms are neighborhood-based [27]. However, recommender system operators encourage the existence of power user communities; e.g., Amazon Vine^{TM}invites the most trusted reviewers on Amazon to post opinions about new and pre-release items to help their fellow users make informed purchase decisions[37]. Furthermore, new items can sometimes pose significant market acceptance challenges to producers of goods and services; in order to address this issue, marketers may rely on the influence that power users have in recommending items to other users [14, 3]. But given the influence that power users can have over others, what happens when power users provide biased ratings for, as yet unrated, new items?

To address this issue, we investigate identification and attack potential of RS power users. We define and study a new "Power User Attack" as a set of power user profiles that influence the results presented to other users by providing biased ratings. This attack is distinct from previously studied types of RS attacks [39, 26, 36] (e.g.,

[37]http://www.amazon.com/gp/vine/help

"Random", "Average", "Bandwagon", etc.) that rely on a set of carefully configured false user profiles, which are injected into the dataset to mount the attack. Power User Attack analyses rely critically on power user identification, so we first develop and evaluate a novel use of degree centrality concepts from social network analysis [57, 27], for identifying the most influential RS power users.

The research questions for this analysis are:

RQ-1: How are power users best identified in a RS?

RQ-2: What happens to the accuracy and robustness of the RS rating predictions when power users attack new items?

RQ-3: How do the popular RS algorithms compare in their robustness against power user attacks?

The hypotheses to be tested are:

H-1: Using Degree Centrality to select a set of power users results in a more effective attack than using either Aggregated Similarity or Number of Ratings.

H-2: A small number of power users (50 or less) can have significant effects on RS predictions and top-N lists of recommendations for new items

H-3: User-based CF algorithms are more vulnerable to a power user attack than item-based CF algorithms.

H-4: SVD-based and item-based CF algorithms are robust to power user attacks.

5.2 Power User Attack Model

We define the *Power User Attack* (PUA) as a new attack model where power users bias the results of the RS predictions and top-N recommendation lists with targeted fake ratings. The *intent* of this attack is similar to other attack models

where a number of attack user profiles are injected into the dataset. Like other attack models, PUA profiles contain target items that are set to the maximum or minimum rating depending on attack intent. However, unlike classic attack models (e.g., random, average, bandwagon) that employ straightforward statistical templates (average rating, popularity, likability) for attack profile filler (non-target items), very little is known about the profile characteristics of power users. And without this knowledge, it is difficult to generate fake power user profiles. So, this initial work into studying the impact of a PUA will use power user profiles that already exist in the dataset (e.g., attack vector of external incentivization or internal collusion).

The number of power users participating in the attack defines the attack size; the larger the attack size, the larger the expected disruption in RS predictions and top-N recommendation lists.

5.3 Power User Selection

Power users in the RS context have been referred to as users with a large number of ratings [20] as well as those that are able to influence the largest number of other users [14, 44, 3, 18]. To measure influence, researchers have used the number of prediction differences above a prediction threshold when a user is removed from the dataset [44], the number of users that had the prediction for a target item shifted sufficiently above a threshold so that the item appears in their top-N list [18], MAE and coverage metrics to evaluate various seed (influential user) selection algorithms [3], and the expected lift in profit earned by influencing other users, recursively [14]. Although maximizing the spread of influence through a social network is an NP-hard problem to solve optimally [23, 18], several heuristics were analyzed by [18] in

Table 2: Similarity matrix between user i and user$j$

Users	*User1*	*User2*	..	*Userj*
User 1	$\emptyset$	S(u_1,u_2)	..	S(u_1, u_j)
User 2	S(u_2,u_1)	$\emptyset$	..	S(u_2, u_j)
..	..	..	..	..
User i	S(u_i,u_1)	S(u_i,u_2)	..	$\emptyset$

order to select groups of influential users including Most Central (those with highest aggregate similarity to other users), Most Positive (those with the highest positive average rating), Most Active (those who have rated the highest number of items), and Random (a control group comprised of randomly selected users). The Number of Unique Prediction Differences algorithm [44] was determined to be computationally inefficient and was not considered further in this study.

We have developed an approach to power user selection for attack purposes, based on social network analysis concepts of Degree Centrality [57, 27]. Specifically, we use In-Degree Centrality (users who appear in the highest number of other users' neighborhoods) with significance weighting [19] because when using similarity and neighborhood-based methods to select power users, significance weighting encourages strong connections between users who have rated many items in common. In a preliminary study, we found that both our approach and the Most Central heuristic [18] performed significantly better using significance weighting.

In our research, power users were identified using a method based on the in-degree centrality concept from social network analysis: using Table 2 as an illustration, for each user i compute its similarity with every other user j applying significance weighting, then discard all but the top-n neighbors for each user i. Then count the

number of similarity scores for each user j (column sums of similarity score counts will indicate the number of neighborhoods user j is in) and select the top-k user j's based on their number of similarity scores.

5.4 Evaluating Power User Influence

In this study, we evaluate power user influence as follows:

- Before the attack: After selecting a set of power users, we analyze the impact those power users have on the recommendations of other users in the dataset. This analysis is carried out by removing a percentage of power users incrementally (from 0% to 100% in increments of 10%) and then calculating the accuracy of the system using Mean Absolute Error (MAE). The intuition is that if the accuracy of the recommender were to get worse as power users are removed then this would be an indicator that those power users provided a positive influence on system accuracy. The accuracy of the system is measured removing selected power users (using the methods described in §5.3) and randomly-selected users. We understand that accuracy is not the same as influence, however, in this approach MAE is used as a proxy for an initial indicator of the "influence" power users can have on system accuracy and recommendations.
- After the attack: To analyze the results of a power user attack and its influence on system robustness, we use use Hit Ratio, Prediction Shift, and Average Rank robustness measures [36, 9] where a high Hit Ratio and a low Average Rank indicates that the attack was successful (from the attacker's standpoint). These robustness measures indicate a more reliable indicator of power user influence after attack.

5.5 Power User Attack against User-based and Item-based Algorithms

We have conducted an initial analysis to investigate several research questions (see §5.1) related to RS power users and attacks on new items potentially perpetrated by power users. To address these research questions we tested the following Hypotheses: (H-1) Using Degree Centrality to select a set of power users results in a more effective attack than using either Aggregated Similarity or Number of Ratings, (H-2) A small number of power users (50 or less) can have significant effects on RS predictions and top-N lists of recommendations for new items, and (H-3) User-based CF algorithms are more vulnerable to a power user attack than item-based CF algorithms.

5.5.1 Experimental Design

Datasets and Algorithms – The data used in this study were the MovieLens[38] 100K[39], 1M[40], and 10M[41] datasets, with item ratings from 1 (did not like) to 5 (liked very much). For the user-based CF algorithm [45, 19, 13], we used Pearson Correlation similarity (with significance weighting of $n_{ci}/50$, where n_{ci} is the number of co-rated items), Neighborhood formation (similarity thresholding = 0.0 and kNN = 25 and 50), and Mean-centered prediction. For the item-based CF algorithm [47], we used Pearson Correlation similarity (with significance weighting of $n_{ci}/50$), Neighborhood formation (similarity thresholding = 0.0), and Weighted prediction. Following earlier studies evaluating power user influence [14, 44, 18], we focus on traditional user-based and item-based CF algorithms for this analysis. We used SVD with 50 features and

[38] http://www.grouplens.org
[39] nominal 100,000 ratings, 1,682 movies, and 943 users.
[40] nominal 1,000,209 ratings, 3,883 movies, 6,040 users.
[41] nominal 10,000,054 ratings, 10,676 movies, 69,878 users.

50 iterations for the ML10M power user selection analysis only.

Power User Selection – The following methods were used:

InDegree (ID2550 and ID5050) - Our method is based on the in-degree centrality concept from social network analysis — power users are those who participate in the highest number of neighborhoods. For each user i compute its similarity with every other user j applying significance weighting $n_{ci}/50$, then discard all but the top-n neighbors for each user i (we used n=25 for ID2550 and n=50 for ID5050). Count the number of similarity scores for each user j (# neighborhoods user j is in) and select the top 50 user j's.

AggregatedSimilarity (AS25NO) - This is the Most Central heuristic from [18]. The top 50 users with the highest aggregate similarity scores become the selected set of power users. This method requires at least 5 co-rated items between user i and user j and does not use significance weighting[42].

NumberRatings (NR) - This method is based on [20] where "power user" refers to users with the highest number of ratings; it also is called the Most Active heuristic in [18]. Select the top 50 users based on the total number of ratings they have in their user profile.

Attack-related Target Item Selection – For each dataset, fifty target items were selected randomly from a set of items with no more than one rating regardless of their rating value, because our intent was to attack 'new' items.

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 is 50,

[42]Based on personal communication with the authors.

40, 30, 20, 10, 5, 3, and 2; 50 power user profiles equate to a 5% attack for ML100K and a 1% attack for ML1M. The attack profiles used were actual power user profiles with the target item rating inserted; the filler size is determined by each power users' profile size. To implement our attack, a group of power users were selected (§5.3), the attack intent (push / nuke) and target item(s) were specified, and the remainder of the profile for the PUA (the filler) remained unchanged for each power user in the attack. By keeping the power users' profiles the same except for the target item, the power users' connections to other users, from a social network standpoint, remain essentially the same[43]

Evaluation Metrics – As indicated in Section 5.4, we use Mean Absolute Error and prediction coverage for accuracy and coverage [20, 56] using a holdout-partitioned 70/30 train/test dataset. We also use Hit Ratio, Prediction Shift, and Average Rank robustness measures [36, 9] where a high Hit Ratio and a low Average Rank indicates that the attack was successful (from the attacker's standpoint). Since the PUA being evaluated here is for new items (zero rating value), the Prediction Shift is expected to be close to the maximum rating as defined by the RS.

Test Variations – Evaluation of the power user attack encompassed two prediction algorithms (User-based CF and Item-based CF), two datasets (ML100K and ML1M), four power user selection methods (Aggregated Similarity using 25 neighbors, InDegree using 25 neighbors, InDegree using 50 neighbors, and Number of Ratings), and eight attack sizes. Each test variation was executed once for each of the 50 target

[43]In a few cases, power user profiles that already had a target item rating were updated in certain attack scenarios and, although the target item rating change might alter their neighborhoods, we believe the impact to this initial analysis is not an overriding issue.

items and data results were averaged over the 50 target items.

5.5.2 Results and Discussion

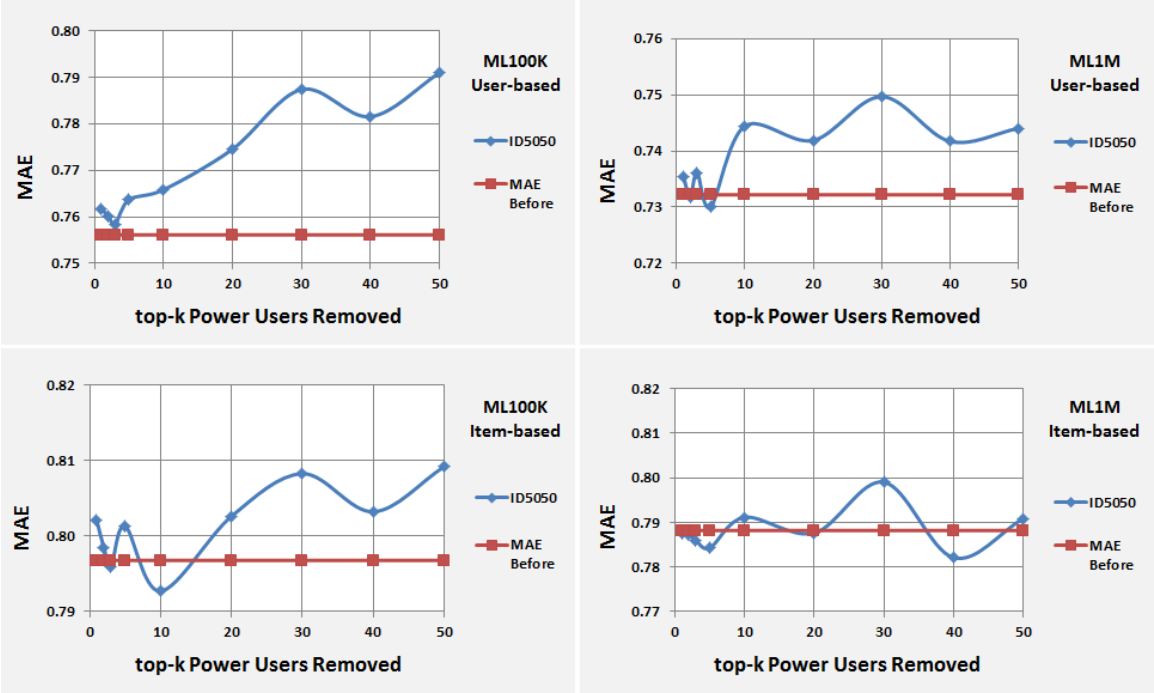

Figure 9: MAE impacts, before and after removing Power Users – In Degree using ML100K and ML1M

(RQ-1) How are power users best identified in a RS? Our approach to evaluating selection of power users and their influence is to use accuracy metrics (before the attack) and robustness metrics (after the attack).

Before the power user attack, the measure of influence is the negative impact on RS accuracy when removing power users. Prior to any attack, we removed power users from datasets of user-item ratings for all three methods of power user selection. The results for InDegree and ML1M (Figure 9) show that as power users are removed, accuracy impacts are more significant on user-based recommenders (p <0.001 when the number of power users removed is > 10) than on item-based recommenders (p <0.025

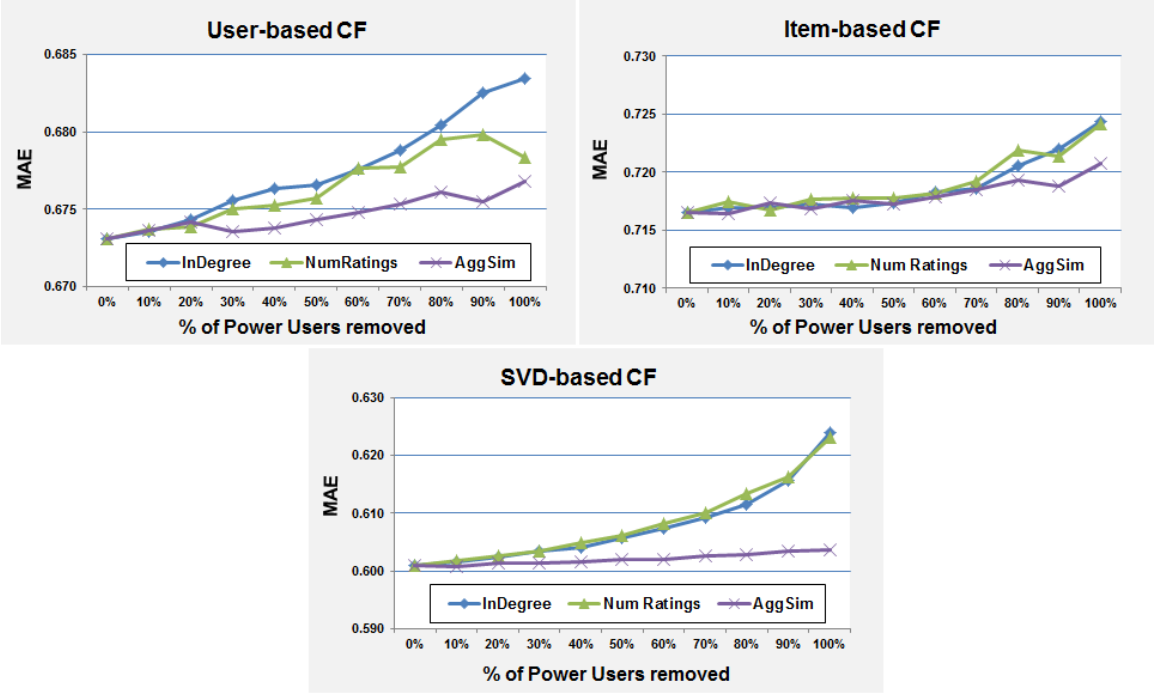

Figure 10: MAE impacts, before and after removing Power Users – using ML10M

when the number of power users removed is = 30) across both dataset sizes. Similar (ML1M) and weaker (ML100K) results occurred for AggregatedSimlarity and NumberRatings methods. Coverage results (not shown) indicate significant impacts as more power users are removed, especially for user-based CF.

The results for ML10M[44] (Figure 10) show that as power users are removed, MAE is negatively impacted in all CF algorithms tested. Compared to a baseline where no power users are removed, InDegree selection impacts MAE by an average high of 2.2% across all three CF algorithms, Number of Ratings by 1.8%, and Aggregated Similarity by 0.5%; all these impacts were significant (p <0.02). InDegree selection, compared to the other methods, has a higher MAE impact in User-based CF. Both InDegree and Number of Ratings methods are better at selecting power users than Aggregated

[44]We selected 3500 power users or about 5% of all ML10M users.

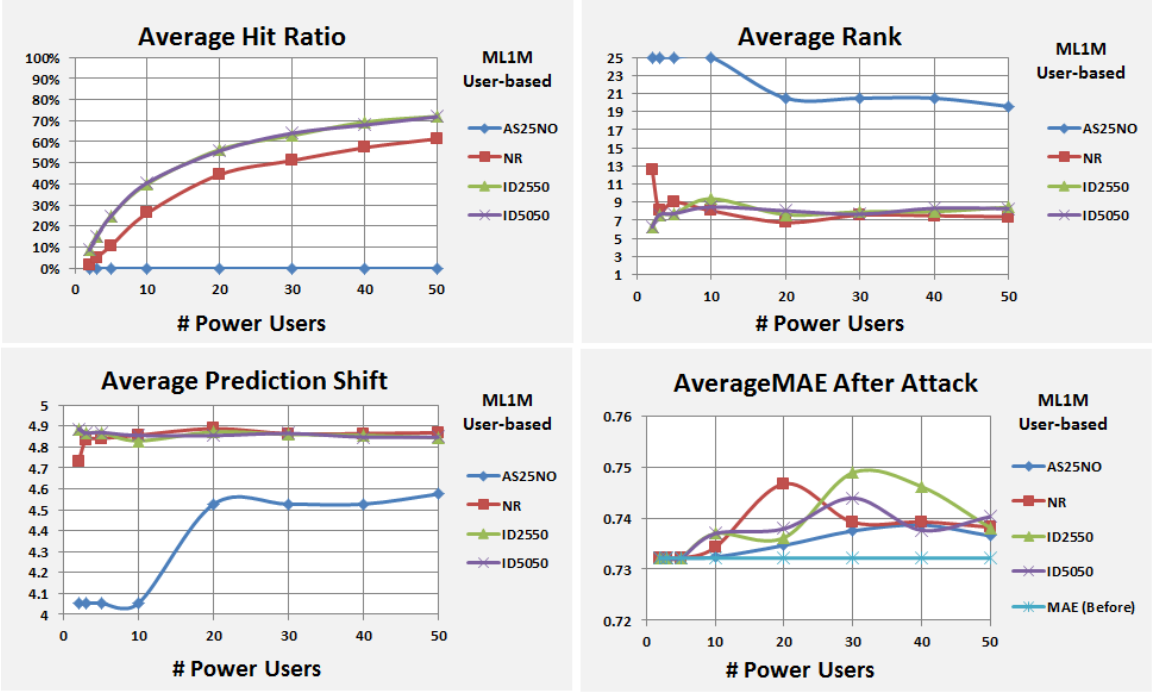

Figure 11: User-based results – ML1M

Similarity, i.e., they have the most significant impacts on MAE with removal of fewer power users, especially in SVD-based and User-based CF. The Number of Ratings method performs relatively well across all CF algorithms and begs the question of whether having a large number of ratings is really the only requirement to become a power user. These accuracy and coverage results are consistent with results found in [27].

After the attack, we believe that the method that selects the most influential set of power users is the one producing the highest Hit Ratio and lowest Average Rank. Both InDegree and NumberRatings power user selection methods dominated (attack was more successful) the AggregatedSimilarity method as indicated by the Average Hit Ratio and Average Rank results for user-based CF using ML100K and ML1M (Figure 11). While Hit Ratio and Average Rank results for InDegree and Number-

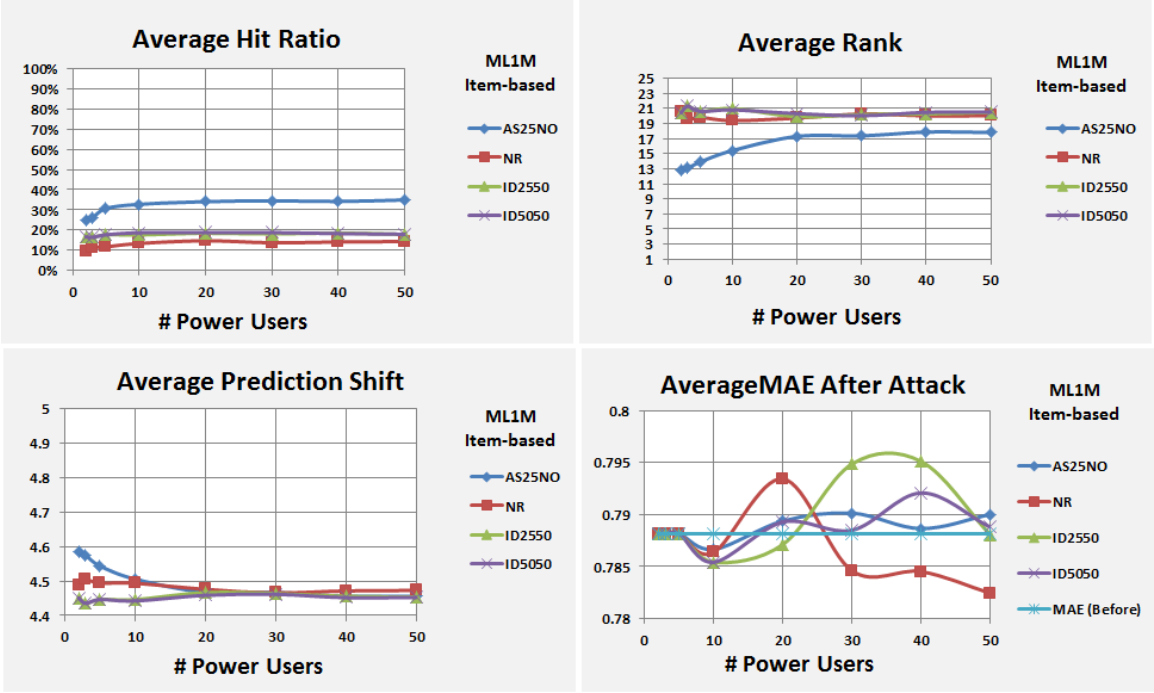

Figure 12: Item-based results – ML1M

Ratings were about the same for ML100K (not shown), InDegree performed better than NumberRatings on the ML1M dataset ($p<0.05$). For item-based CF, none of the power user selection methods tested here produced a major impact to Hit Ratio or Average Rank using ML100K or ML1M (Figure 12). From the attacker's perspective, the AggregatedSimilarity method was slightly more effective than InDegree and NumberRatings.

Therefore, using In Degree Centrality to select a set of power users results in a more effective attack than using either Aggregated Similarity or Number of Ratings, confirming our first Hypothesis H-1 for user-based CF only. Impacts to the accuracy metrics (Figure 9) and robustness metrics (Figure 11) indicate that a small number of power users[45] can have significant effects on RS predictions and top-N recommenda-

[45]Note that 50 power users is $< 1\%$ of the ML1M user base.

tion lists for new items, confirming our second Hypothesis H-2 for user-based CF only. Hypotheses H-1 and H-2 are accepted for user-based CF and rejected for item-based CF.

(RQ-2) What happens to the accuracy and robustness of the RS rating predictions when power users attack new items? Accuracy and robustness metrics were negatively impacted due to the PUA. Accuracy results were mixed across datasets, CF algorithms, selection methods, and size of attack. Notably after attack, all four selection methods yielded MAE values significantly higher (p <0.05) for ML1M user-based CF and 30 or more power users, and InDegree and NumberRatings yielded MAE values significantly higher (p <0.05) for ML1M item-based CF with 30 or 40 power users. As noted above, robustness results (Figures 11 and 12) indicate that InDegree and NumberRatings impact user-based CF recommenders more significantly than AggregatedSimilarity; however, we found the reverse to be true for item-based CF recommenders.

(RQ-3) How do the popular RS algorithms compare in their robustness against power user attacks? User-based CF is significantly more vulnerable to the power user attack than Item-based CF, confirming our third Hypothesis H-3. This is consistent with previous findings [26, 36, 9] because the PUA, like the random and average attacks, are able to exploit the similarity between the attackers and non-attackers to favor the target item. For item-based CF, the AggregatedSimilarity method produced a more effective set of power users for the attack as compared to InDegree and NumberRatings across both datasets (ID5050 and AS25NO Hit Ratio difference was significant at p <0.001 for ML1M); however, the impact of the attack was weak,

i.e. relatively low Hit Ratio and high Average Rank, compared to user-based CF. Hypothesis H-3 is accepted for user-based CF.

5.6 Power User Characterization

In order to understand the differences between power and non-power user groups within a given dataset, we have collected various statistical measures for each group across ML100K, ML1M, and ML10M. Notably for InDegree across these datasets, a statistically significant (p <0.02) pattern is emerging: power users have a lower average rating, higher average number of ratings, higher average number of co-rated items, and higher rating entropy when compared to the non-power user group and the entire dataset of users. Although the absolute number for each measure varies according to the dataset, the pattern is consistent. Knowledge of these and other statistical characteristics of power users will be needed in order to construct fake power user attack profiles required to mount a PUA.

5.7 Power User Attack against an SVD-based Algorithm

5.7.1 Singular Value Decomposition (SVD)

The implementation of matrix factorization SVD [48, 25, 2] we used was the Expectation Maximization (EM) algorithm [12] provided in the Apache Mahout platform[46]. This algorithm requires two parameters: number of features and number of training steps. A sensitivity analysis was performed on these parameters to observe the impact on Mean Absolute Error (MAE) using the MovieLens ML100K dataset [47] and results are shown in Figure 13. Based on these results, we found that when holding

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

[47] www.grouplens.org; MovieLens dataset with 100,000 ratings, 1,682 movies, 943 users, 93.7% sparsity.

the number of training steps constant, MAE remains relatively flat as the number of features is varied. Conversely, when holding the number of features constant, MAE decreases to a minimum and then begins to increase. For 100 features, the minimum MAE occurs at 75 training steps; the differences in MAE between 25 and 75 steps and between 100 and 75 steps are significant ($p < 0.01$).

Figure 13: MAE impacts when varying SVD parameters using ML100K

5.7.2 Experimental Design

To address our research questions, we conducted an experiment using the MovieLens 100K dataset with an SVD-based recommender. Power users were selected from the dataset using three identification/selection methods. To simulate the PUA, power user profiles were converted to attack profiles by setting target items in those profiles to the maximum rating. Target items selected had no more than one rating in order to simulate a "new" item. Evaluations of accuracy and robustness were performed before and after the attack.

Evaluation Metrics – We use Mean Absolute Error (MAE) and prediction coverage for accuracy and coverage [20, 56] using a holdout-partitioned 70/30 train/test dataset. We also use Hit Ratio, Prediction Shift, and Rank robustness measures [36, 9] where a high Hit Ratio and a low Rank indicates that the attack was success-

ful (from the attacker's standpoint). Since the PUA being evaluated here is for new items (zero rating value), the Prediction Shift is expected to be close to the maximum rating as defined by the RS.

Datasets and Algorithms – We used the ML100K dataset with item ratings from 1 (did not like) to 5 (liked very much). For the SVD-based CF algorithm, we used the EM (see §5.7.1) algorithm as implemented in Mahout 0.4. Run-time parameters used for this algorithm were number of features (100) and number of training steps (75); settings were determined empirically as described in §5.7.1. The more traditional user-based and item-based CF algorithms were studied in a previous effort (see § 5.5) and those results will be used here for comparative purposes.

Power User Selection – The following methods were used,

InDegree - Our method is based on the in-degree centrality concept from social network analysis, where power users are those who participate in the highest number of neighborhoods. For each user i compute its similarity with every other user j applying significance weighting, then discard all but the top 50 neighbors for each user i. Count the number of similarity scores for each user j (# neighborhoods user j is in) and select the top 50 user j's.

AggregatedSimilarity (AggSim) - This is the Most Central heuristic from [18]. The top 50 users with the highest aggregate similarity scores become the selected set of power users. This method requires at least 5 co-rated items between user i and user j and does not use significance weighting[48].

NumberRatings (NumRatings) - This method is based on [20] where "power user"

[48]Based on personal communication with the authors.

refers to users with the highest number of ratings; it also is called the Most Active heuristic in [18]. We selected the top 50 users based on the total number of ratings they have in their user profile.

Target Item Selection – For the ML100K dataset, 5 target items with no more than one rating, regardless of their rating value, were selected randomly, given our objective to attack only 'new' items. We recognize that 5 target items is a limitation in this study; however, new items are more vulnerable to attack than more popular items so this should provide a strong signal even with a small number of target items. We are considering a larger mix of new/existing target items as a future work.

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 is 50, 30, 10, 5, 3, 2, and 1. The maximum attack size (50) was selected based on previous research [36, 9], where a 5-10% attack was shown to be effective; with ML100K, a 5% attack size is about 50 users. The attack profiles used were actual power user profiles and we added the target item rating. The Filler Size, or number on non-target items in each attack user profile, is determined by each power users' profile size; therefore, filler size is not specified in this experiment.

Test Variations – One prediction algorithm, one dataset, three power user selection methods, and seven attack sizes. Each test variation was executed once for each of the 5 target items and data results were averaged over the 5 target items.

5.7.3 Results and Discussion

(RQ-1) How are power users best identified in a RS? Our assertion is that the amount of influence power users exerted on other users, before and after an attack,

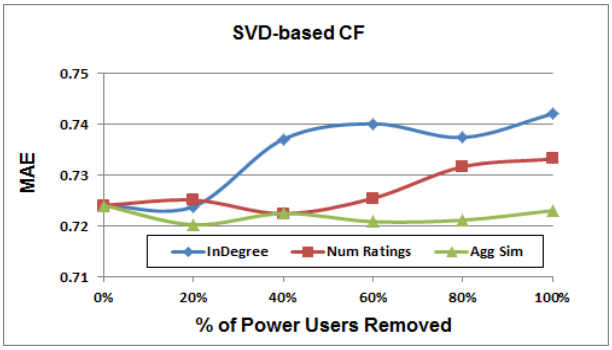

Figure 14: MAE impacts after removing power users using ML100K

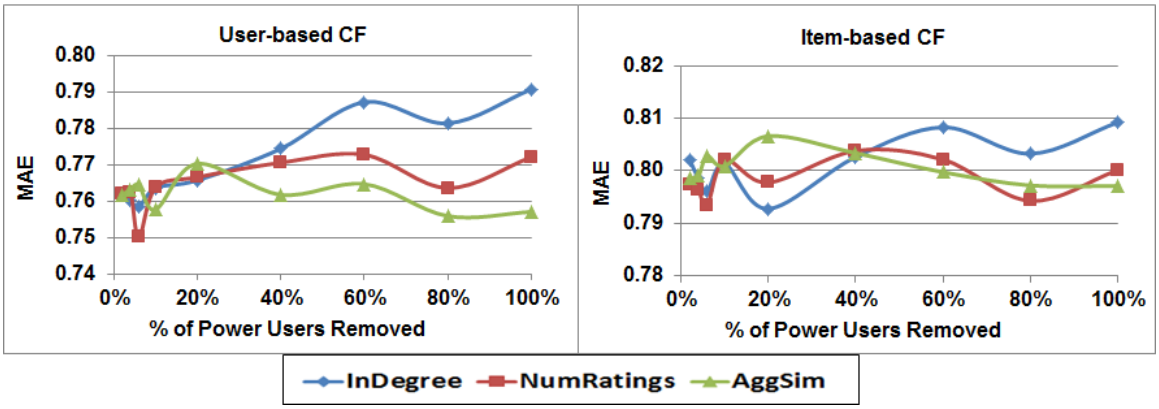

Figure 15: MAE impacts after removing power users using ML100K

would indicate the best identification method. *Before the power user attack*, one measure of influence is the negative impact on RS accuracy (MAE) when removing power users [27]. We removed from 0 to 50 (0% to 100%) of the identified power users from the dataset before any attacks took place for all three methods of power user selection; the most influential power users identified are removed first. The results for InDegree (Figure 14) show that as power users are removed, accuracy impacts are significant on SVD-based recommenders (p <0.01) when power users removed are > 20%. Similar results occurred for the NumRatings method when power users removed are > 60%, and influence of AggSim-selected power users remained flat. Furthermore, InDegree has significantly more impact on MAE than AggSim (p <0.01) at all levels

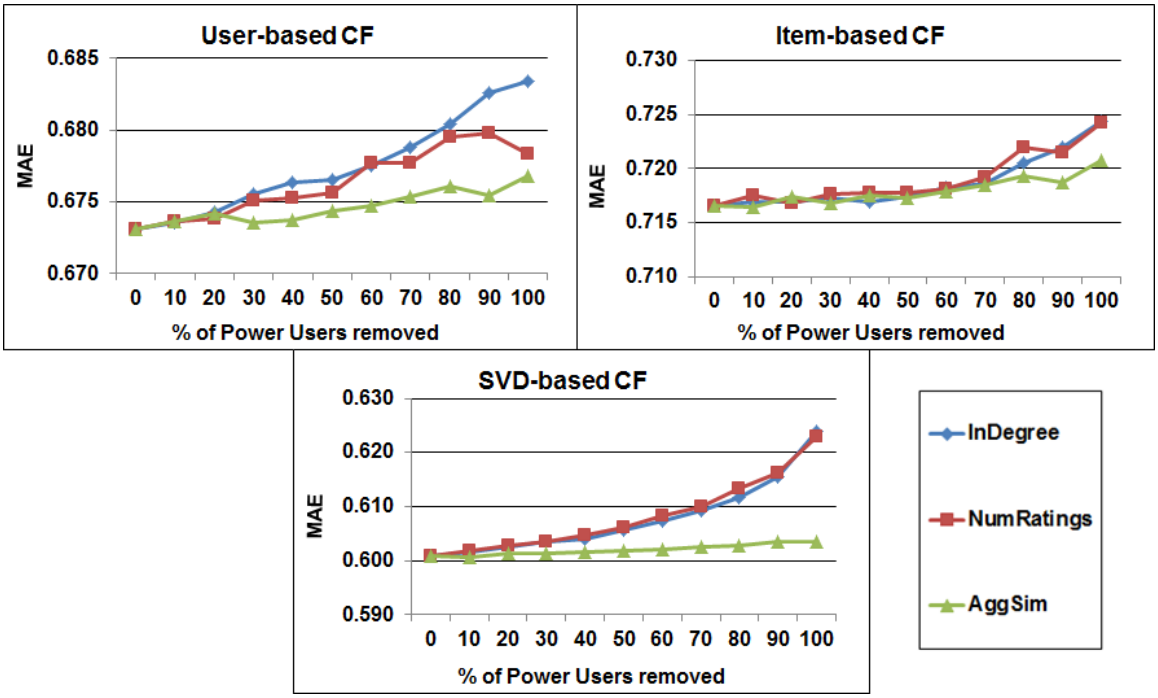

Figure 16: MAE impacts, after removing power users using ML10M

of power user removal and NumRatings (p <0.01) when power users removed are > 20% and < 80%. As a baseline, we removed users at random and found that the ablation curve for randomly-selected users is flat from 0% to 100% removed, i.e., their removal shows no significant impact on MAE. Coverage results (not shown) remained flat and at a high level (> 99%) for all power user selection methods and number of power users removed. Hypothesis H-1 is partially accepted, given that InDegree and NumRatings are about equal in their impact in the ablation results and after the power user attack; also, InDegree and NumRatings both have more impact than AggSim in the ablation results and all three selection methods are about the same after the power user attack. Hypothesis H-2 is accepted for InDegree and NumRatings for the ablation results, and rejected for AggSim for the ablation results. Hypothesis H-2 is accepted for all three selection methods based on results after the

power user attack. The results obtained here are also consistent with those observed in our previous work [60, 51] using various CF algorithms: Figure 15 shows results using the ML100K dataset and Figure 16 shows results using the ML10M dataset[49]. *After the attack*, we expect for influence to be measured mainly by the impact on robustness metrics, i.e., the method that selects the most influential set of power users is the one producing the highest Hit Ratio and lowest Rank. Results show that all the power user selection methods were successful (from the attacker's standpoint) at impacting the robustness metrics.

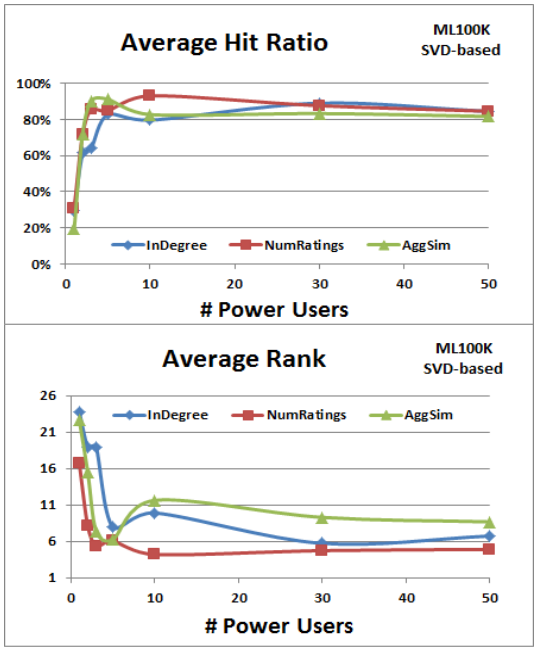

Figure 17: ML100K – SVD-based results

(RQ-2) What happens to the accuracy and robustness of the RS rating predictions when power users attack new items? We found that the PUA was successful (from the attacker's standpoint) at impacting RS robustness metrics across all three power user selection methods, as indicated by the Average Hit Ratio and Average Rank results

[49]MovieLens dataset with 10,000,054 ratings, 10,676 movies, 69,878 users, 98.7% sparsity

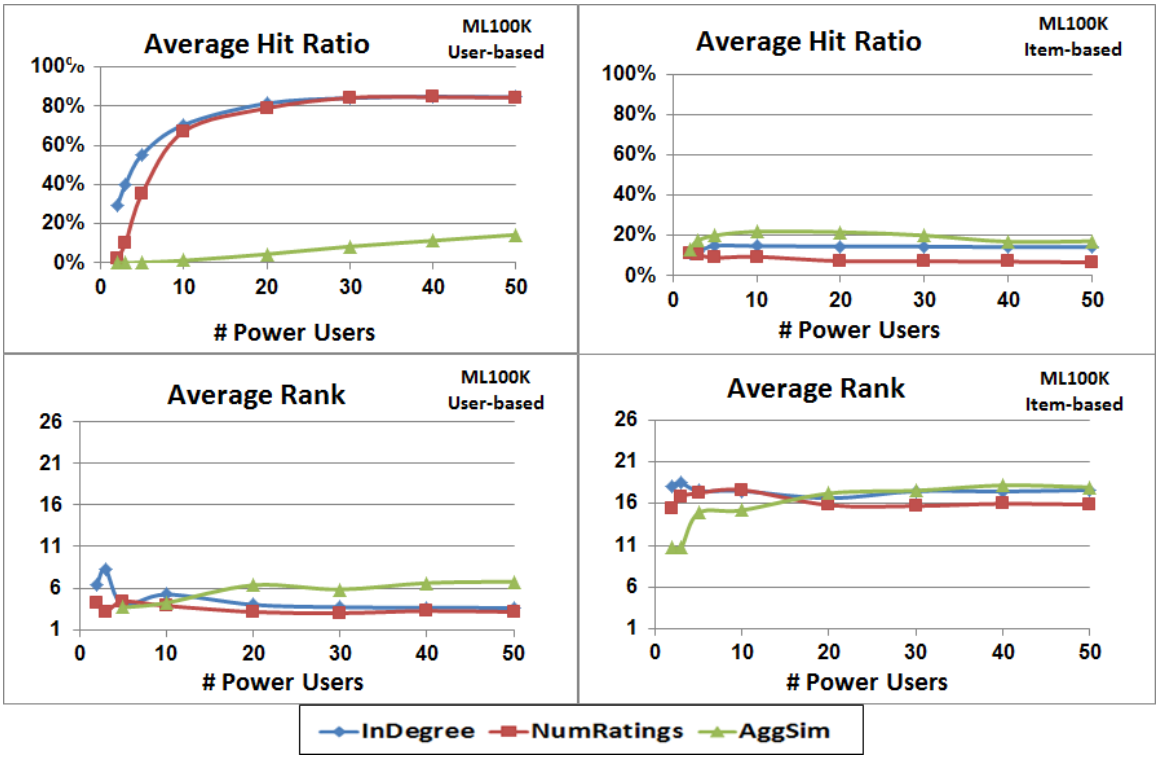

Figure 18: ML100K – User and Item-based results

shown in Figure 21; no significant differences were found between the three methods with 50 power user attack profiles. High levels of Average Hit Ratio and low levels of Average Rank were achieved with as few as 5 to 10 power users. Impacts to the robustness metrics indicate that a small number of power users[50] can have significant effects on RS predictions and top-N recommendation lists for new items. With 50 power user attack profiles, the InDegree method showed a significantly lower (better) Average Rank than AggSim (p <0.01) and a significantly higher (worse) Average Rank than NumRatings (p <0.01). As expected, Prediction Shift (not shown) was high (> 4) given that the target items were "new" items.

This result is interesting given that SVD-based systems have been shown to be robust to attacks [32]. In that work, the authors used clustering techniques to iden-

[50]Note that 10 power users is < 1% of the ML100K user base.

tify the attackers based on their statistical signatures, i.e., Random, Average, and Bandwagon attack models; the attack clusters were then eliminated from, or ignored during, the prediction calculation. In our experiment, the attackers were not eliminated from the dataset nor ignored during the prediction calculation, therefore, we see a more effective attack against the SVD algorithm.

(RQ-3) How do the popular RS algorithms compare in their robustness against power user attacks? As noted above, robustness results (Figure 21) indicate that SVD-based recommenders are vulnerable to attack by power users with results comparable to user-based recommenders as shown on the left side of Figure 18 [60], especially for the InDegree and NumRatings power user selection methods. The right side of Figure 18 indicates that item-based recommenders are less vulnerable to the impacts of the PUA. User-based CF is significantly more vulnerable to the power user attack than Item-based CF and is also consistent with previous findings [26, 36, 9] because the PUA, like the random and average attacks, are able to exploit the similarity between the attackers and non-attackers to favor the target item. For item-based CF, the AggSim method produced a more effective set of power users for the attack as compared to InDegree and NumRatings; however, the impact of the attack was weak, i.e. relatively low Hit Ratio and high Rank, compared to user-based CF. Hypothesis H-4 is rejected for SVD-based and accepted for item-based.

Compared to user-based and item-based algorithms, we have shown a strong attack using an EM implementation of SVD although it appears insensitive to power user selection methods. Additional research to determine whether this is due to scale given ML100K's size, use of the EM SVD algorithm vs. other SVD techniques, or the input

parameters to the EM SVD algorithm has been conducted in Chapters 6, 7, 8, and 9. Varying the SVD algorithm, algorithm parameters, and dataset size/domain does not appear to affect the fact that SVD results are insensitive to power user selection methods.

5.8 Summary of this Chapter

Power users are important to recommender systems and contribute to their improved prediction accuracy; however, we have found that power user attacks can be effective (from the attacker's perspective) against recommender systems. In this initial study, we have shown that a relatively small number of power users can have significant effects on RS predictions and top-N recommendation lists. We have also shown that the InDegree method of power user selection produces a set of power users that are able to mount more effective attacks than the AggregatedSimilarity and NumberRatings methods, especially on user-based CF systems. The contributions of this research have been to provide a new attack model called Power User Attack and to evaluate a social network analysis method for selecting the top-k influential power users for attack purposes.

With respect to the Dissertation Hypotheses provided in Section 1.5.2, this chapter has indicated the following level of support for the applicable hypotheses; final acceptance/rejection of the Dissertation Hypotheses are provided in the Dissertation Summary, Section 10.1:

DH-1: The use of In-Degree Centrality to select a set of power users results in power users with higher influence than other selection techniques, across multiple datasets and domains. This hypothesis is supported for the user-based algorithm for the

ML100K, ML1M, and ML10M datasets; the hypothesis is not supported for item-based and SVD-based algorithms. In some of the unsupported cases, In-Degree and Number of Ratings methods select power users with similar influence.

DH-4: A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects on RS predictions and top-N lists of recommendations across multiple power user selection techniques, collaborative filtering algorithms, datasets, and domains. This hypothesis is supported for user-based and SVD-based algorithms for the ML100K and ML1M datasets; the hypothesis is not supported for item-based algorithms.

CHAPTER 6: POWER USER ATTACK MODEL AND EVALUATION

6.1 Introduction

As a foundation for understanding influence based attacks, we adapt established network measures of influence to the context of RSs, in order to identify power users in the underlying dataset. In our previous work [54, 60], we identified *real power users* (RPUs) using selection methods based on network centrality, user-user similarity, and, rating behavior. We then used those RPUs to mount a Power User Attack (PUA) and found that accuracy and robustness metrics were negatively impacted for commonly used RS approaches. *For clarity, the power user attack envisioned in this research is not about having hundreds or thousands of actual power users colluding to mount an attack, rather, it is about an attacker being able to generate a set of power user profiles that, when stealthily injected into a RS, can effectively bias the recommendations.* Knowing that a Power User Attack with RPUs can be effective, the natural next question is whether RPUs can be modeled to enable / automate the generation of completely *synthetic power user* (SPU) profiles with the same degree of impact as attack vectors. In effect, the "evil twins" of the real power users. This Chapter describes our approach to generate synthetic attack profiles to emulate and exploit the influence characteristics of real power users, and it studies the impact of attack vectors that employ synthetic power user profiles.

The research questions for this analysis are:

RQ-1: Can SPU profiles be generated that effectively model RPUs?

RQ-2: Can SPU profiles be effective in attacking RSs?

The hypotheses to be tested are:

H-1: A majority of the SPU profiles injected into a given dataset will be successfully identified by the same power user selection method used to identify the respective RPU profiles, i.e., precision and recall scores will be $> 50\%$.

H-2: Datasets, with SPU profiles for each power user selection method, evaluated using an ablation approach will indicate a statistically significant increase in MAE as SPU profiles are removed from the dataset.

H-3: The MAE differences achieved for the power user selection methods will be comparable to what was observed when RPUs were removed from those same datasets.

H-4: Statistical characteristics of RPUs and PUM-generated SPUs will be measured and no statistically significant differences will be found between them for average number of ratings per user, average user rating, and average item rating.

H-5: SPU profiles identified using the InDegree power user selection method will have a higher level of impact, compared to SPU profiles identified using NumRatings or AggSim, on RS predictions and top-N recommendation lists as measured with Average Hit Ratio and Average Rank robustness metrics.

H-6: A relatively small number of power users (<=5% of all users) can have significant effects on RS predictions and top-N lists of recommendations, measured with an Average Hit Ratio $> 50\%$ and Average Rank < 10.

6.2 Overview of Foundational Power User Attack Research

The PUA relies critically on the method of power user identification/selection, so we also developed and evaluated a novel use of degree centrality concepts from social network analysis for identifying influential RS power users for attack purposes [60]. In addition, we chose to use the Most Central and Most Active heuristics from [18] because this would provide us with their best-case and worst-case scenarios that we could then use to compare with our degree centrality approach. The power user selection methods that we have used previously (see § 5.5.1) are as follows:

1. *InDegree*: Our approach based on in-degree centrality — power users participate in the highest number of neighborhoods. For each user i compute similarity with every other user j applying significance weighting $n_{cij}/50$, where n_{cij} is the number of co-rated items and 50 items was determined empirically by [19] to optimize RS accuracy; then discard all but the top-N neighbors for each user i. Count the number of similarity scores for each user j (# neighborhoods user j is in), and select the top-N user j's.
2. Aggregated Similarity (*AggSim*): The Most Central heuristic from [18]. Top-N users with the highest aggregate similarity scores become the selected set of power users. This method requires at least 5 co-rated items between user i and user j and does not use significance weighting.[51]
3. Number of Ratings (*NumRatings*): This method is based on [20] where "power user" refers to users with the highest number of ratings; it also is called the Most Active heuristic in [18]. We selected the top-N users based on the total

[51] Based on personal communication with the authors.

number of ratings they have in their user profile.

When evaluating power item selection methods, there are attack dimensions such as cost and knowledge required that should be considered [26, 9]. 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.

To evaluate power user selection methods, we use an ablation approach [27, 60], where accuracy of the RS is measured as power users are removed from the dataset. If accuracy gets worse when power users are removed, the implication is that power users are impacting the RS recommendations. The intuition is that the power user selection method that is able to identify the set of users with the greatest negative impact on system accuracy is the better method. We previously reported an ablation analysis for RPUs using the MovieLens 100K, 1M, and 10M datasets for user-based, item-based, and SVD-based recommenders [60, 51, 54]. The results indicated that all three power user selection methods described above show an increase in Mean Absolute Error (MAE), i.e., accuracy gets worse, as power users are removed and that this effect is stronger with the InDegree and NumRatings methods than with

AggSim.

6.3 Power User Model

Our first research question (RQ-1) is how to effectively generate synthetic power user profiles, which has two main aspects. First, we must be able to effectively identify real power users. For this study we employ the methods we have used previously in Section 5.5.1 and described above in (§ 6.2). Second, with a mechanism in place to identify real power users, the next step is to develop a generative model for synthetic power users based on the identified RPUs. This section describes our proposed new model.

In a laboratory environment it would be possible to create synthetic power users with specific influence characteristics; however, from a practical perspective, attackers may not have the ability or resources to be that precise. Attackers recognize that there is a tradeoff between the effort (or cost) of mounting an attack and the effectiveness (or impact) of the attack; their intent is to maximize the impact at a minimal cost. Furthermore, it is not our intention to create synthetic power users that maximize the values of the respective power user selection methods, i.e, maximum possible in-degree, or number of ratings, or aggregated similarity. Although it would be interesting to understand the maximal impacts that those power users could have when attacking a collaborative recommender, these attacks may not be very practical to mount from an attacker's perspective; additionally, such strong attacks may be easy to detect or mitigate, e.g., attacks with synthetic users that have rated 100% of all items. Therefore, we chose a more practical middle ground that generates synthetic power users that emulate characteristics of real power users. Our objective is

to investigate the extent to which synthetic power users (based on real power user characteristics) are influential enough to impact recommendations.

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 [36], very little is known about the characteristics of power users. And without this knowledge, it is difficult to generate synthetic power user profiles. Therefore, we have developed a Power User Model (PUM) that can be used to generate synthetic power users (SPU) for attack purposes. We base our PUM on the primary factors considered in order to build effective RS attacks [26, 36], which include:

1. Attack size: the number of attack user profiles to be injected. A larger attack size is more effective, however, it is more easily detectable.
2. Filler size: the number of item ratings in the attack user profile, excluding the target item. A larger filler size is more effective, however, it is also more easily detectable.
3. Filler item selection: items that are likely to correlate with many other users in the system will be more effective.
4. Filler item rating: ratings that are likely to correlate with many other users in the system will be more effective.
5. Target item selection: items with few ratings are more vulnerable to attack.
6. Target item rating: on a 1-5 rating scale, use 5 for "push" attacks and 1 for "nuke" attacks

With these dimensions as guidelines, we generate synthetic user profiles in the fol-

lowing manner by collecting targeted statistics of identified real power users. For this initial evaluation of the PUM, we use the MovieLens 100K dataset[52] to identify/select RPU's. We then collect user, item, and neighborhood characteristics from the dataset and begin to build the power user model to generate SPU profiles:

1. Power User selection methods are InDegree, NumRatings, and AggSim described in § 6.2.
2. Attack size: 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 [36] has shown that a 5-10% attack size should be sufficient to have an impact on recommendation robustness; therefore, we use a conservative 5% attack size or 50 SPU's for this analysis.
3. Filler 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. Previous work [36] has shown that a 5-10% filler size should be sufficient to have an impact on recommendation robustness. However, in this study, the filler size for each profile is selected randomly from a normal distribution around the mean and standard deviation (σ) of the number of ratings in the dataset for the RPUs identified/selected by each selection method. This approach was used so that it would closely mimic the behavior of real power users. For this study, the filler size distributions varied by selection method: InDegree (μ=317.78 and σ=124.981), NumRatings (μ=395.32 and σ=93.031), and AggSim (μ=35.64 and σ=21.886).

[52] 100,000 ratings, 1,682 movies, 943 users, 93.7% sparsity.

Table 3: Distribution of items by popularity bucket

% items/bucket	*Low*	*MedLow*	*Medium*	*MedHigh*	*High*
InDegree	25.79%	32.20%	21.65%	12.52%	7.83%
NumRatings	29.86%	32.75%	19.83%	11.08%	6.48%
AggSim	14.42%	23.79%	23.63%	16.44%	21.72%

4. Item selection is based on the average number of user ratings by item category for the RPU's identified in the dataset; for this study, the item category is popularity or the number of ratings for the item. We selected popularity as an initial approach with the intent of using other characteristics, such as likability and genre, in the future. For each power user selection method, we determined the distribution of items rated for the RPU's in five item popularity "buckets" and required each SPU profile to contain a similar distribution, as shown in Table 3. The buckets were defined taking into account that the "number of ratings" characteristic usually follows a power law wherein a large number of items have a relatively small number of ratings (i.e., the least popular movies) and a small number of items have a large number of ratings (i.e., the most popular movies):
    - Low: items with an average number of ratings
    - Medium Low: items with an average number of ratings + 1σ
    - Medium: items with an average number of ratings + 2σ
    - Medium High: items with an average number of ratings + 3σ
    - High: items with greater than average number of ratings + 3σ
5. Item rating value for each item in the profile is selected randomly from a normal distribution around the mean and standard deviation of the average item's

rating in the dataset for the RPU's identified/selected by each selection method. Our intent was for SPU's to have a rating profile similar to RPU's rather than just randomly assigning rating values. We used a normal distribution because this has been typical in RS attack research [26, 36] and because it may be the best fit given the overall average item rating for ML100K of μ=3.077 and σ=0.780 on a 1-5 scale.

6. Target items selected will be "new" items, i.e., those with only one rating.
7. Attack intent for this study will be "push", i.e., the target item rating will be set to the max rating value of 5.

6.3.1 Evaluating the Power User Model: Results and Discussion

For RQ-1, we want to know whether SPU profiles can be generated that effectively model RPUs. For this experiment, we consider the following hypotheses:

- *H-1:* A majority of the SPU profiles injected into a given dataset will be successfully identified by the same power user selection method used to identify the respective RPU profiles, i.e., precision and recall scores will be $> 50\%$.
- *H-2:* Datasets, with SPU profiles for each power user selection method, evaluated using an ablation approach will indicate a statistically significant increase in MAE as SPU profiles are removed from the dataset.
- *H-3:* The MAE differences achieved for the power user selection methods will be comparable to what was observed when RPUs were removed from those same datasets.
- *H-4:* Statistical characteristics of RPUs and PUM-generated SPUs will be measured and no statistically significant differences will be found between them for

average number of ratings per user, average user rating, and average item rating.

To evaluate the SPU profiles (before the attack), we remove the top 50 RPU's from the original ML100K dataset using each of the three selection methods (InDegree, NumRatings, AggSim) and replace them with 50 SPU profiles to create modified ML100K datasets.[53] We remove the RPU's to see how well the 50 SPU's would replace them. Then, we identify/select the top 50 power users from the modified datasets using each of the three selection methods.

First, we use precision and recall metrics to determine the extent to which the 50 SPU's are actually selected by each method. The PUM generated SPU profiles with varying degree of success based on the power user selection method used. For InDegree, 70% of the SPU's were identified and NumRatings achieved 83% precision and recall scores, while AggSim was only able to achieve a 32% precision and recall score. Although there is no precedent for determining whether these scores are adequate or inadequate, the next two evaluation methods will also need to be considered. Hypothesis H-1 is accepted for InDegree and NumRatings, rejected for AggSim, meaning that the PUM generated an acceptable number of SPU's that were successfully identified/selected by the InDegree and NumRatings methods and not the AggSim method.

Next, we look at the ablation results in Figures 19 and 20 comparing RPU (left graphs) and SPU (right graphs) behavior. We observe that as InDegree-selected SPU's are removed, MAE increases ($p < .01$ for SVD-based and $p < .05$ for User-

[53]NB: The desired attack size (5% of users in the dataset) is equivalent to 50 SPU's; the same number of SPU profiles are evaluated before and after the attack.

based); for NumRatings and AggSim, MAE is either flat or decreases ($p < .02$ for decrease in SVD-based NumRatings). Hypothesis H-2 is accepted for InDegree and rejected for NumRatings and AggSim, and this would indicate that InDegree-generated SPU's are more effective in influencing recommendations than the other two methods.

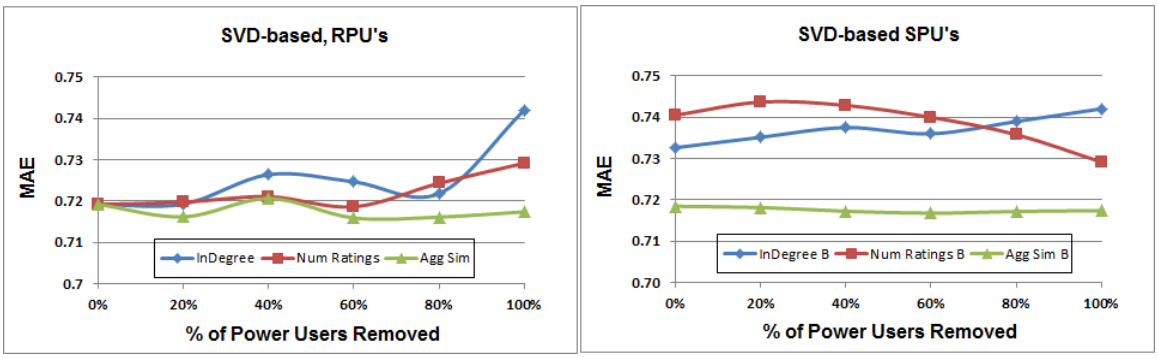

Figure 19: MAE impacts after removing power users using ML100K

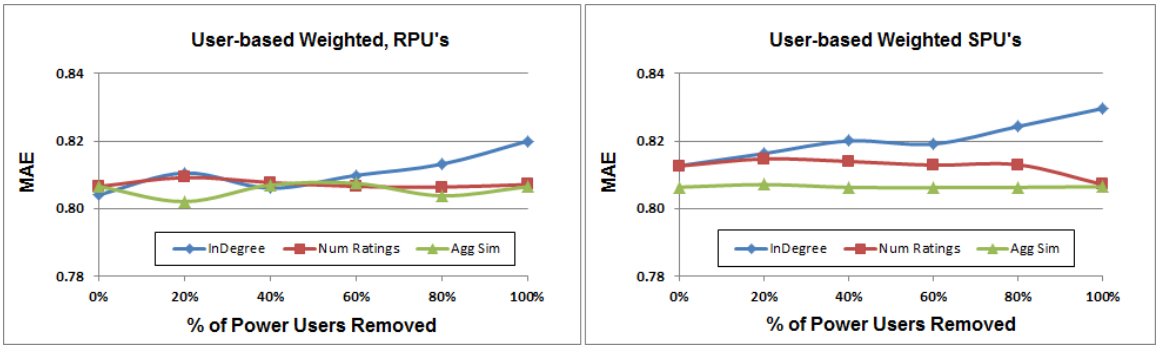

Figure 20: MAE impacts after removing power users using ML100K

For SVD RPU's, MAE differences for all three methods are only significantly different from each other at 100% removal ($p < .02$). For SVD SPU's, with the exception of NumRatings at 100% removal, both InDegree and NumRatings are significantly different from AggSim ($p < .01$) at all removal levels; and there is no significant difference between InDegree and NumRatings at any level of removal. For User-based RPU's, MAE differences for all three methods are only significantly different from

each other at 100% removal as follows: InDegree-NumRatings ($p < .05$), InDegree-AggSim ($p < .02$). For User-based SPU's, InDegree-AggSim are significantly different at all removal levels and InDegree-NumRatings are significantly different only at 100% removal ($p < .01$). Hypothesis H-3 is rejected for SVD-based and accepted for User-based analyses. This would indicate that, in general for SPU's, InDegree and NumRatings tend to have better ablation performance than AggSim; furthermore, InDegree SPU's achieve an equal or higher MAE at 100% removal than RPU's indicating a strong level of influence for these SPU's.

Finally, when comparing statistical characteristics between SPU's and the RPU's upon which they are based, we found significant differences in user and item rating entropy as well as global rating values for SPU's across all three power user selection methods ($p < .01$). Notably, the NumRatings method was able to significantly impact the global average rating value (downward from 3.302 to 3.190) between the RPU's and SPU's groupings as well as for the full ML100K dataset ($p < .01$ in both cases); this may help to explain the performance of the NumRatings method in the ablation study as well as in the PUA results. Hypothesis H-4 is accepted for InDegree, NumRatings, and AggSim, i.e., no statistically significant differences were found between RPU's and SPU's for the key measures of average number of ratings per user, average user rating, and average item rating. This indicates that the PUM is generating SPU's that match the key statistical measures, however, work is needed to improve the user and item rating entropy measures.

6.4 Synthetic Power User Attack

For RQ-2, we want to understand whether generated SPU profiles can be effective in attacking RSs. For this experiment, we consider the following hypotheses:

- *H-5:* SPU profiles identified using the InDegree power user selection method will have a higher level of impact, compared to SPU profiles identified using NumRatings or AggSim, on RS predictions and top-N recommendation lists as measured with Average Hit Ratio and Average Rank robustness metrics.
- *H-6:* A relatively small number of power users (<=5% of all users) can have significant effects on RS predictions and top-N lists of recommendations, measured with an Average Hit Ratio $> 50\%$ and Average Rank < 10.

To mount the PUA, synthetic power user profiles were generated as described in § 6.3 and converted to attack profiles by setting target items to the max rating. Target items were selected to simulate a 'new' item attack because this is a typical scenario in which power users are asked to provide ratings. Evaluations were performed before and after the attack using the Apache Mahout 0.8 platform[54].

Evaluation Metrics – To evaluate the PUA, we use Mean Absolute Error (MAE) and prediction coverage [20, 56] using a random holdout-partitioned 70/30 train/test dataset. We also use Hit Ratio, Prediction Shift, and Rank robustness measures [36, 9] where a high Hit Ratio and a low Rank indicates that the attack was successful (from the attacker's standpoint). Since the PUA being evaluated here is for new items (i.e., not very many ratings in the dataset), the Prediction Shift is expected to be close to

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

the max rating defined by the RS.

Datasets and Algorithms – We used MovieLens 100K (ML100K) where each user has 20 or more ratings to avoid the 'new' user problem. The algorithms used were provided in Apache Mahout. For 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. The user-based weighted CF algorithm was used for comparative purposes.

Power User Selection – Methods are described in § 6.2.

Target Item Selection – Given our objective to attack 'new' items, 50 target items with only one rating were selected randomly from the dataset.

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 in this experiment: 50, 30, 10, 5, 2, and 1, where 50 power user profiles equate to a 5% attack for ML100K. The Attack profiles used were SPU profiles described in § 6.3 and we injected the target item rating at run time. The Filler size for each profile varied for each SPU and is described in § 6.3.

Test Variations – The test variations consisted of 2 prediction algorithms, one dataset, 3 power user selection methods, and 6 attack sizes. Each test variation was executed 50 times (once for each of the 50 target items) and data results were averaged over the 50 target items.

6.4.1 Evaluating the Power User Attack: Results and Discussion

The PUA on SVD in Figure 21 shows significant impacts to recommendations between 5 and 50 SPU's for InDegree, NumRatings, and AggSim. NumRatings SPU

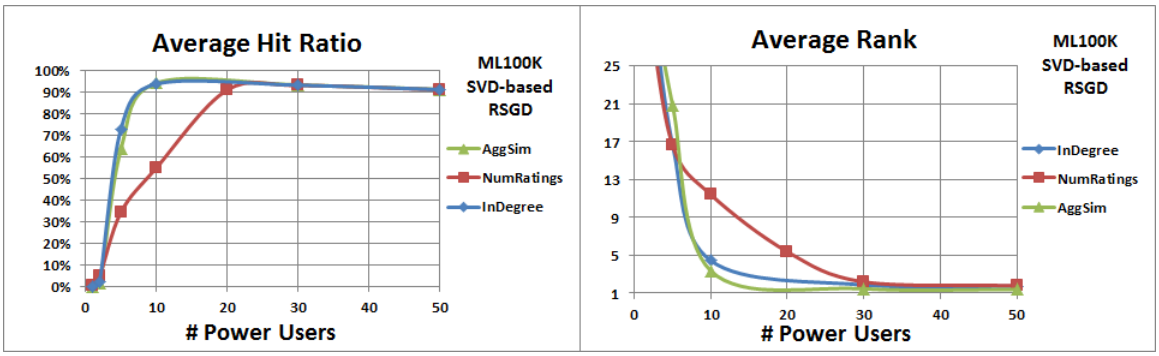

Figure 21: ML100K – SVD-based results

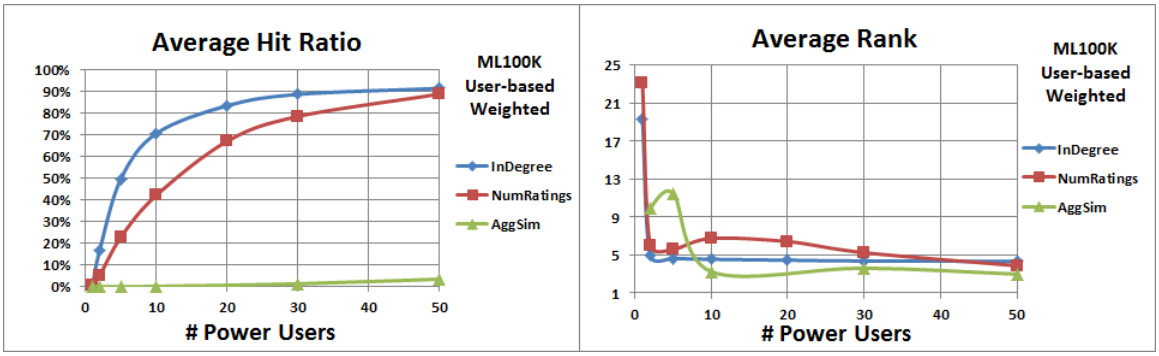

Figure 22: ML100K – User-based results

influence on Avg Hit Ratio begins to break down below 30 power users and is also evident in the Avg Rank; the difference between NumRatings and both InDegree and AggSim is significant between 5 and 30 power users ($p < .01$). The results for InDegree and AggSim, as well as the trend for NumRatings, are consistent with our previous work with attacks on SVD recommenders using RPU's [54]. Previous work on RS attacks has indicated that SVD is robust to attack [31]. However, this is the case only when attackers have been detected and removed from the recommendations; our experimentation does not remove attackers prior to generating recommendations. The PUA on User-based in Figure 22 shows significant impacts to recommendations between 10 and 50 power users for InDegree and NumRatings while AggSim is never a factor($p < .01$). These findings are consistent with our previous PUA research [60].

The InDegree-generated SPU's produces a strong set of power users, both before and after the attack; InDegree results are significantly different from NumRatings in the range of 2 to 30 power users ($p < .01$). Hypothesis H-5 is rejected for SVD-based and partially accepted for User-based analyses (no difference at 50 SPU's). The interpretation of this result is that InDegree may be a more superior power user selection method than NumRatings and AggSim for User-based recommenders and that there is no clear superior power user selection method for SVD-based recommenders. Hypothesis H-6 is accepted for both SVD-based and User-based analyses, 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.

6.5 Summary of this Chapter

Power users are important to recommender systems and contribute to their improved prediction accuracy; however, we have found power user attacks that are effective against popular recommender systems. In this chapter we have developed a power user model that is able to generate synthetic power user profiles that, in specific configurations, can be used to mount effective power user attacks against SVD-based and User-based recommenders measured by Hit Ratio and Rank robustness metrics. We have shown that our power user model generates effective synthetic (vs. actual) power user profiles as measured with accuracy, precision, and recall metrics. We have also shown that a relatively small number of synthetic power users can have significant effects on RS predictions and top-N recommendation lists for new items. The contributions of this research have been to provide a new Power User Model

and a process for generating synthetic power user attack profiles based on statistical characteristics of power users.

In Chapters 5 and 6, all of the experiments use "new" target items, i.e., those with only one rating. The reason for this was to ensure that the attacks were capable of producing a strong signal or impact, essentially providing an upper bound on attacker influence. However, there are well-known techniques that system operators can use to protect the robustness of new items until they have enough ratings and become less vulnerable to attack, e.g., new item quarantine,[55] content-boosted collaborative filtering [33], market-based use of recommender systems [14, 3], and Local Collective Embeddings, a matrix factorization approach that exploits items' properties and past user preferences [50]. So to continue to establish the effectiveness of power user attacks on collaborative recommenders, in Chapters 7, 8, and 9, we conduct power user (and power item) attacks using not only "new" target items but also "new and established" targets. In many instances, we find that power user/item attacks with "new and established" targets are effective albeit not as effective as those with only "new" target items.

With respect to the Dissertation Hypotheses provided in Section 1.5.2, this chapter has indicated the following level of support for the applicable hypotheses; final acceptance/rejection of the Dissertation Hypotheses are provided in the Dissertation Summary, Section 10.1:

DH-1: The use of In-Degree Centrality to select a set of power users results in power users with higher influence than other selection techniques, across multiple datasets

[55] http://www.inf.unibz.it/dis/research/seminar_slides/burke.ppt

and domains. This hypothesis is supported for user-based and partially accepted for SVD-based algorithms for the ML100K dataset. It is partially supported for SVD-based recommenders because In-Degree shows higher influence than Aggregated Similarity and equal influence with Number of Ratings.

DH-2: A significant percentage of synthetic user profiles generated from statistical characteristics of power users will be identified by selected power user selection techniques across multiple datasets and domains. This hypothesis is supported for In-Degree and Number of Ratings methods for the ML100K dataset; the hypothesis is not supported for the Aggregated Similarity power user selection method.

DH-3: Power user attack profiles generated from characteristics of InDegree-selected power users will result in more effective attacks (from the attacker's viewpoint) than attack profiles generated from characteristics of power users selected from other techniques across CF algorithms, datasets, and domains. This hypothesis is not supported for SVD-based and partially supported for user-based algorithms for the ML100K algorithm, i.e, In-Degree is a more superior power user selection method than Number of Ratings and Aggregated Similarity for user-based recommenders and there is no clear superior power user selection method for SVD-based recommenders.

DH-4: A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects on RS predictions and top-N lists of recommendations across multiple power user selection techniques, collaborative filtering algorithms, datasets, and domains. This hypothesis is supported for user-based and SVD-based algorithms for the ML100K dataset.

CHAPTER 7: POWER ITEM ATTACK MODEL AND EVALUATION

7.1 Introduction

Recommender systems that use model-based approaches such as item-based and SVD matrix factorization algorithms have been found to be robust to many types of attack [36, 31, 59, 21]. The conventional advice in designing for system robustness has thus been to employ model-based approaches [37]. 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* [54, 60, 61]. We found that Power User Attacks (PUAs) are able to successfully impact SVD-based and user-based recommenders [54, 60, 61]. However, we also confirmed previous research [26, 36, 61] that item-based systems remained fairly robust to attack. Because attackers continue to develop new approaches for biasing RS results, it is critically important for researchers to keep pace in analyzing potentially new attack vectors. Therefore, our challenge was to determine how to attack the item-based algorithm.

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 that power items would exhibit the same type of influence found with power users. To successfully attack item-based recommender systems [47], prior research showed that item-item similarities can be manipulated; e.g., this was demonstrated in the Bandwagon and

Segment attacks [6, 36]. And with this knowledge, it is possible that an attacker could generate attack user profiles that exploit this vulnerability in the item-based CF algorithm. Furthermore, it is conceivable that an attacker would want to attack multiple items, e.g., promoting a set of related items from a single supplier or promoting similar items from multiple suppliers. So, rather than mount multiple attacks each of which targets a single item, the attacker can more efficiently impact multiple items in a single attack. However, while an attack using multiple target items could be effective against the item-based CF algorithm, it may not be as effective against user-based CF recommenders: User-based recommenders would compute user-user similarities of attack user profiles (containing multiple target items) and form neighborhoods of users that have similar tastes not only with filler items but also with the multiple target items, effectively reducing the focus and effectiveness of the attack. To eliminate confounds between the filler/selected items (that are used to correlate with other users) and the target item, only single target item attacks have been used in the past [36, 9] against user-based recommender systems. Therefore, we believe that using influential power items as filler/selected items is particularly well-suited to attack the item-based collaborative filtering algorithm that generates recommendations based on the similarity between items (not users) and will not suffer from the confounds between power items and the multiple target items. Instead, the multiple target items become strongly associated with the power items in the attack user profile that are, in turn, used to correlate with other users in the dataset.

This Chapter 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 [47].

The research question for this analysis is:

RQ-1: Can a Power Item Attack successfully (from the attacker's viewpoint) impact item-based recommenders as measured with Hit Ratio, Prediction Shift, and Rank robustness metrics?[56]

The hypotheses to be tested are:

H-1: A PIA with relatively small number of Synthetic Power Item Profiles (SPIP's)[57], i.e., <=5% of all users, can have significant effects on RS predictions and top-N lists of recommendations, measured with robustness metrics.

H-2: 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.

7.2 Selecting Power Items

To select *power items* our initial study employs the same methods we used previously for power user selection [61]. 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 [57], 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

[56] See § 7.5 for description of robustness metrics.
[57] See § 7.3 for a description of SPIP's

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.[58] 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 [18]. 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.[59]

Number of Ratings (NumRatings or NR) – Power users were defined in [20] 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 [6, 36, 21].

Although PIA detection is beyond the scope of this dissertation, we should note that detailing the Power Item Model (§ 7.3) and the methods for selecting power items (§ 7.2) provides the basic information required for detection analysis.

7.3 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 pro-

[58] We used a divisor of 50 users as an analog to work done with co-rated items in user neighborhoods by [19] to optimize RS accuracy.

[59] Based on personal communication with the authors.

file filler items [36], 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 (§ 7.2) 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 [36]. 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 may have more impact, however, it is also more easily detectable. Previous work [36] 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 § 7.2. 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. [26, 36].

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 [26, 36]:

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 [36] 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 § 7.2. However, these attack models differ primarily in the contents of I_F and I_S as shown in Table 4. Furthermore, the PIA-MT differs radically from previously studied attacks using popular items, not only in the profile contents shown in Table 4 but also in that the PIA-MT uses multiple targets simultaneously rather than a single target item in order to mount the attack.

The PIM approach goes beyond prior research primarily in two areas: first, we utilize influence-based methods (§ 7.2) to select the power items for the attack user profile, and second, we utilize multiple rather than just single target items. We believe

Table 4: 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

that this combination can yield powerful attacks, especially against the item-based algorithm that has been resistant to attack in the past [26, 36].

7.4 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 [36] 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 ef-

fectiveness 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:

H-1: 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 [26] 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 [6, 36], i.e, Hit Ratio $> 11\%$ and Prediction Shift > 0.1.

H-2: 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 [57] that high InDegree centrality is indicative of nodes (users) that have strong influence over other users.

7.5 Experimental Design

Evaluation Metrics – Evaluations were performed before and after the attacks using the Apache Mahout 0.8 platform[60]. For robustness metrics [36, 9], 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

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

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 $NTPU = \frac{\sum_{i \epsilon U_T} NTPU_u}{|UH_T|}$. The range of values for NTPU is from one to the total number of target items used in the attack. Because we are averaging over all users with hits, if only one user had any hits and got all of them, the attack would be considered maximally effective. While this scenario is possible, it indicates that the attack is maximally effective for only one user, i.e., from an attacker's viewpoint, this attack would be effective only if that one user was being targeted. Such an attack would be costly to mount and not likely to be pursued. A more likely scenario is that a relatively small number of users garner many of the hits and NTPU is relatively high. In this case, the attack is considered effective for that subset of users. The objective of NTPU is to provide a measure that considers the level of "penetration" for a given attack in terms of the number of targets appearing in a given set of user top-N lists; the set of users ranging from one to the total number of users in the dataset. The value of the NTPU is limited by this objective. Other measures such as Recall (% relevant retrieved) could be used for this purpose, however, those measures do not directly indicate number of targets per user. 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[61] ML100K[62], ML1M[63], and ML10M[64] datasets. The RS algorithms used were provided in Apache Mahout and customized for this study. The CF user-based weighted algorithm (UBW) [13] 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 [19]. The item-based weighted algorithm (IBW) [47] uses Pearson Correlation 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.

Attack User Profiles – To mount the Power Item Attack, attack user profiles were generated as described in § 7.3 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 § 7.2.

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.

[61] http://www.grouplens.org
[62] nominal 100,000 ratings, 1,682 movies, and 943 users.
[63] nominal 1,000,209 ratings, 3,883 movies, 6,040 users.
[64] nominal 10,000,054 ratings, 10,676 movies, 69,878 users.

Table 5: 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

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 5. The Attack profiles were generated as described in § 7.3 and the target item rating was injected at run time.

Test Variations – For all three experiments, we used all three power item selection methods (§ 7.2). 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.

7.6 Experiments and Results

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

Single target item attacks have been used in the past [36, 9] 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 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 [6, 36].

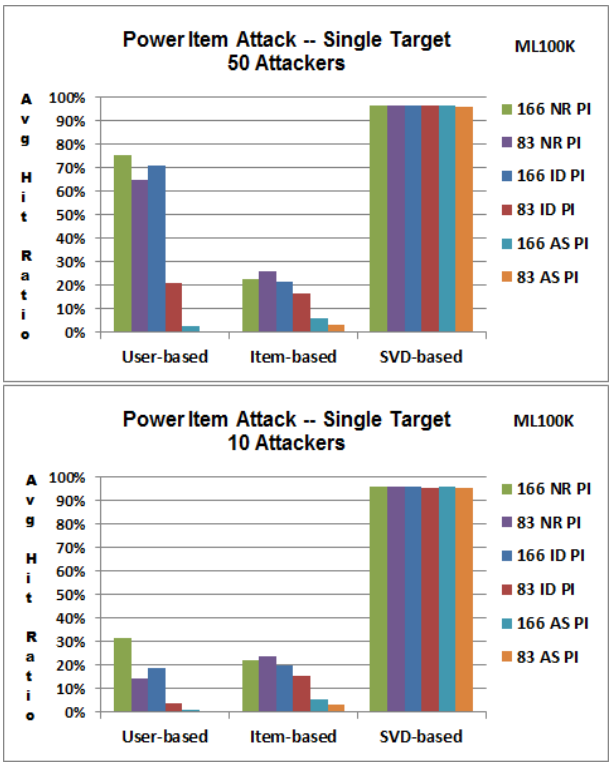

Figure 23: ML100K – Experiment 1 Hit Ratio results

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 § 7.5) 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 23. For the case with 50 attackers or 5% of user base (top of Figure 23), 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 23), 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) [26, 36, 60], showing that the item-based algorithm is resistant or robust to attack. Hypothesis H-1 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 H-2 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).

7.6.2 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 [26, 36, 9] 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 [26, 36, 9]. 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 [36, 6] 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.

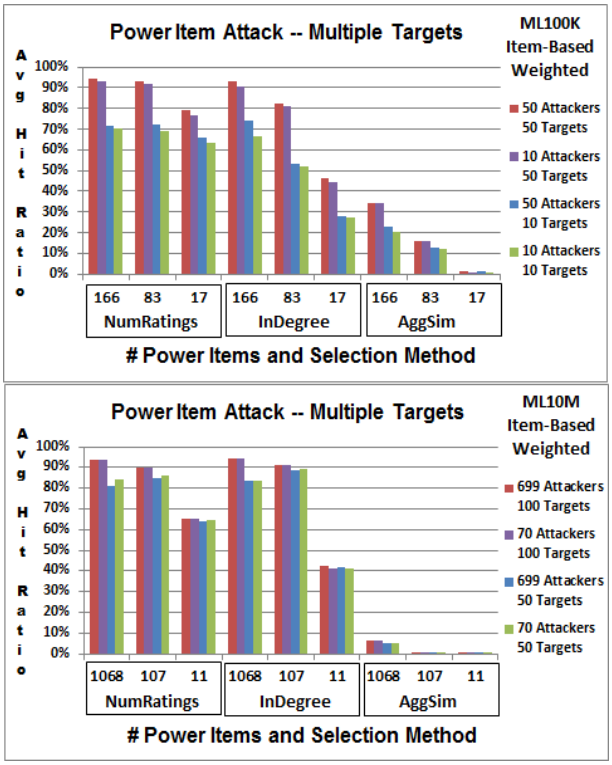

Figure 24: ML100K / ML10M – Experiment 2 Hit Ratio results

The effectiveness of Experiment 2 was measured using robustness metrics (see § 7.5). 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

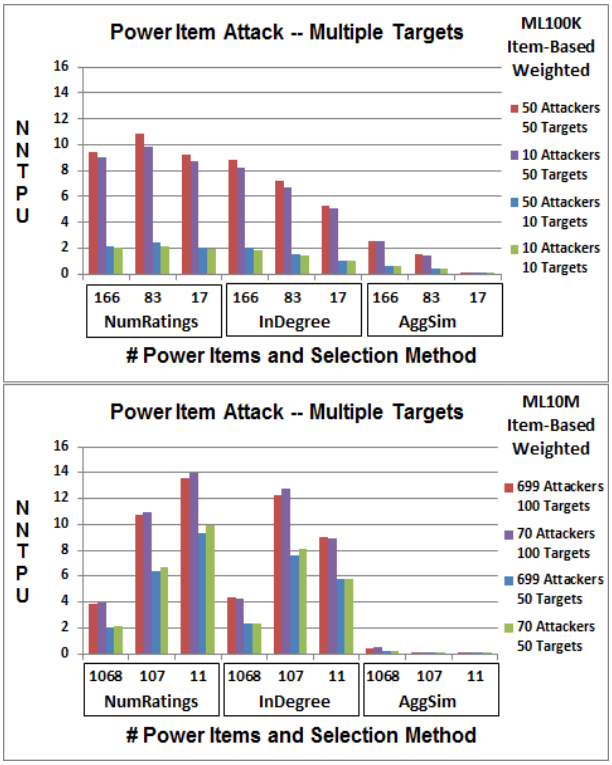

Figure 25: ML100K / ML10M – Experiment 2 Normalized Number of Targets Per User (NNTPU) results

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 5 for parameter settings. The ML100K $\overline{HR}$ results shown in the upper chart of Figure 24 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 24. 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 25. 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.

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 H-1 is accepted for the higher levels of attack size and number of targets for all power item selection methods and all three datasets. Hypothesis H-2 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.

7.6.3 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 5 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

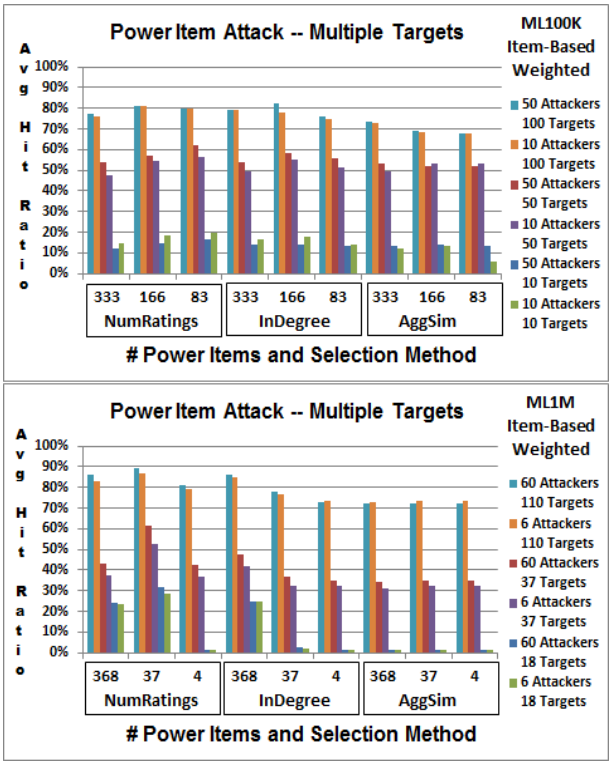

Figure 26: ML100K / ML1M – Experiment 3 Hit Ratio results

of Figure 26 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 26; 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).

To compare the attack effectiveness between Experiments 2 and 3, we used the

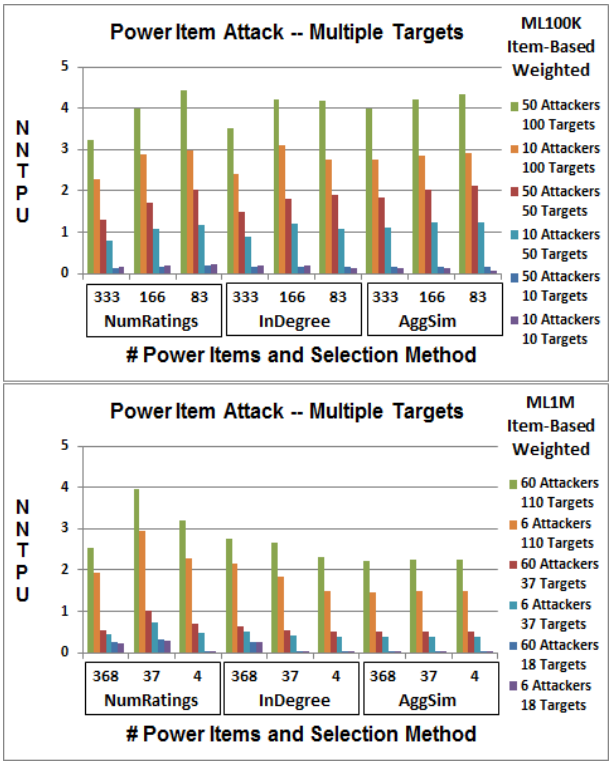

Figure 27: ML100K and ML1M – Experiment 3 Normalized Number of Targets Per User (NNTPU) results

NNTPU metric shown in Figure 27 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 27); this occurs consistently for all three power item selection methods. Reported in [36], 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 25).

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, [6, 36] 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 H-1 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 $\overline{PS}$. H-1 is partially accepted for ML10M for $\overline{HR}$. Hypothesis H-2 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.

7.7 Summary of this Chapter

In this chapter 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.

With respect to the Dissertation Hypotheses provided in Section 1.5.2, this chapter has indicated the following level of support for the applicable hypotheses; final acceptance/rejection of the Dissertation Hypotheses are provided in the Dissertation Summary, Section 10.1:

DH-3: Power user attack profiles generated from characteristics of InDegree-selected power users will result in more effective attacks (from the attacker's viewpoint) than attack profiles generated from characteristics of power users selected from other techniques across CF algorithms, datasets, and domains. For this chapter, the power user selection methods were used to select power items, therefore, the power user attack profiles were generated from characteristics of InDegree-selected *power items.* Nevertheless, this hypothesis is still valid and serves to determine how well the InDegree method can be used to generate influential power users. For the single-item (ST) Power Item Attacks, this hypothesis is not supported for user-based, item-based, and SVD-based algorithms for ML100K and ML1M. Although the InDegree and Number of Ratings perform well and at a high level, Number of Ratings is a slightly better method for selecting power items; they are both superior to Aggregated Similarity. For the multiple-item (MT) Power Item Attacks, this hypothesis is partially supported for the item-based algorithm for the ML100K, ML1M, and ML10M datasets. We find that the InDegree and Number of Ratings methods, on average, perform about the same at all levels of power items and both methods are superior to Aggre-

gated Similarity.

DH-4: A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects on RS predictions and top-N lists of recommendations across multiple power user selection techniques, collaborative filtering algorithms, datasets, and domains. This hypothesis is supported for the item-based algorithm for the ML100K, ML1M, and ML10M datasets.

CHAPTER 8: POWER USER ATTACK MITIGATION

8.1 Introduction

In previous work [61, 55], we have shown that attackers emulating power users are effective against user-based, item-based, and SVD-based recommenders. In the literature, mitigating RS attacks usually consists of detecting the attackers and either removing them from the dataset or ignoring them during the prediction calculations [10, 32]. While removing attack user profiles from recommendation calculations is a straightforward approach to eliminating the attacker's influence in a laboratory environment, using this approach in live RS environments could have unwanted side effects [32]. For instance, in cases where a legitimate power user is mistakenly identified as an attacker (false positive) and is removed, two issues could occur: (1) the removed legitimate power user would no longer receive recommendations, and (2) the users that rely on that legitimate power user's neighborhood influence may be impacted. These approaches also assume that all attackers will be detected, i.e., no provision is provided for attackers that are not detected (false negatives).

This study investigates the potential for more effective impact mitigation approaches against Power User Attacks (PUAs), as compared to 100% removal of identified power users. PUA mitigation walks a fine line between two key RS measures (see §8.4): accuracy (too many power user attack profiles are removed) and robustness (too few power user attack profiles are removed). In this work, removal of identified

power user attack profiles is considered a worst-case scenario for attack mitigation and may not be applicable in situations where system operators want to prevent attackers from knowing that they have been detected. As an alternative strategy for power user attack mitigation, we propose and investigate approaches that keep all the identified power users in the dataset and reduce the influence that those power users have on recommendations for other users. The influence reduction approaches consist of (1) attenuating the similarity (influence) that power users have with other users in their k-nearest neighborhood [27], and (2) reducing the number of power users that are allowed to participate in other users' k-nearest neighborhoods. We then evaluate these removal and influence reduction approaches to determine the approach that best balances RS accuracy and robustness measures.

The research questions for this analysis are:

RQ-1: What happens to RS accuracy and robustness when power user profiles are removed from recommendation calculations to mitigate the power user attack impacts?

RQ-2: What happens to RS accuracy and robustness when power user influence is reduced during similarity calculations?

RQ-3: What are the trade-offs between accuracy and robustness when power user attacks are mitigated?

The hypothesis to be tested is:

H-1: Reducing the influence of power users is a more effective and less impactful mitigation strategy than removing the profiles of identified power users.

8.2 Power User Attack Background

In order to study RS attacks based explicitly on measures of influence, we previously defined a *Power User Attack* model as a set of power user profiles with biased ratings that influence the results presented to other users [60]. The PUA relies critically on the method of power user identification/selection, so we developed and evaluated a novel use of degree centrality concepts from social network analysis for identifying influential RS power users for attack purposes [60]. In addition, we chose to use other heuristics because this would provide best-case and worst-case scenarios that we could use to compare with our degree centrality approach.

The power user selection methods used in this analysis are as follows:

- In-Degree Centrality: Users with the highest user-user in-degree values are selected as power users.
- Aggregated Similarity: Users with highest user-user similarity correlation values are selected as power users.
- Number of Ratings: Users with the most number of ratings are selected as power users.

For more details, please refer to Section 6.2.

8.3 Mitigation Strategies

Removing 100% of the power user attackers as a mitigation strategy could result in various issues: (1) reduced coverage for the "removed" users including legitimate users (false positives), 2) reduced accuracy for users whose similarity neighborhoods no longer benefit from the influence of the "removed" users including legitimate au-

thoritative users (false positives) [32], and (3) no provision for attackers that are not detected (false negatives) and assumes that all (true) power user attackers will be detected. In this analysis, removal of identified power user attack profiles is considered a worst-case scenario for attack mitigation and may not be applicable in situations where system operators want to prevent attackers from knowing that they have been detected. As an alternative strategy for power user attack mitigation, we investigate approaches that keep all the identified power users in the dataset and reduce the influence that those power users have on recommendations for other users. Therefore, the following mitigation strategies were initially evaluated in this study:

- Remove attackers incrementally from 0% to 100%.
- Reduce the similarity weighting factor of all attackers incrementally from 1.0 to 0.0.
- Combine removal and influence reduction.

Our analysis of these initial mitigation strategies determined the following:

When removing power user attackers incrementally from the dataset, removal sequence matters. From the attacker's standpoint, it would be better to remove starting from least influential to most influential; while from the system operator's standpoint, removing starting from most influential to least influential would be better. And we also analyzed the impacts when removing power user attackers randomly. Since this is a mitigation study, we decided to use a removal sequence that favored system operators. Comparative results (§ 8.5) indicate that removing power users starting from most influential to least influential improves robustness at a faster rate than the other two methods.

When mitigating the PUA, the type of target item matters. We used "New" target items (those with one rating) and "New and Established" target items (those with one or more ratings). From previous research [55], we knew that New target items are more vulnerable to attack than New and Established targets. We analyzed the impacts of the PUA and found that, for New targets, robustness metrics were relatively high until the weighting was set to zero (ignore power user influence). For New and Established targets we found that robustness measures were significantly lower between weightings from 1.0 to 0.1, indicating that the PUA was not as effective with these target items.

Combining power user attacker removal and influence reduction resulted in outcomes similar to the removal approach when the similarity weighting was greater than zero. Robustness was at a minimum only when the similarity weighting was zero. So, this approach did not provide additional information regarding power user attack mitigation (§ 8.5).

In our initial approach, all injected attackers were also considered to be power users (even when they were not) so that removing and/or reducing the influence of power users assumed a perfect power user attacker detection method. This was not a very realistic assumption so we decided to use our power user selection methods (§ 8.2) to allow for a mix of real and synthetic power users. Furthermore, only synthetic users were injected as attackers; this meant that a subset of synthetic users selected as power users were also attackers and the rest of the synthetic users were attackers but not power users. Consequently, the final mitigation strategies (MS) for this study are:

MS1: Remove selected power users incrementally from 0% to 100%, starting from most influential to least influential.

MS2: Reduce the similarity weighting factor of all selected power users incrementally from 1.0 to 0.0.

MS3: Reduce the number of power users that influence predictions. The percentage of power users selected is reduced incrementally from 100% to 0% and the similarity weighting is set to one if selected, zero otherwise.

To implement these mitigation strategies, the following methodology was used:

1. Generate power user lists from selected datasets using power user selection techniques, including InDegree, NumRatings, and AggSim (see § 8.2). This generates a list of real power users (RPUs).
2. Generate synthetic power user (SPU) attack profiles based on power user statistical characteristics [61] and insert into the dataset. Select power users from the updated dataset using the power user selection techniques described in § 8.2. A top-k list of power users will, therefore, be a combination of RPUs and SPUs.
3. Select target items from a given dataset: New Items (items with one rating), New and Established Items (randomly-selected items with a range of popularity and likeability values).
4. Create incremental datasets with most-to-least-influential power users removed i.e., from the top of the top-k list of power users.
5. Execute attacks for each mitigation strategy for the selected target items and calculate averaged metrics over all target items. Only SPUs will be used for attack purposes, leaving RPUs to provide their influence but not be part of the

attack. Note that some SPU attack profiles will remain in the dataset after the top-k power users are removed during the experiments as described in § 8.4.

6. Compare accuracy, coverage, and robustness metrics for variations of the mitigation strategy to determine impacts of removing and reducing influence of power users.

Other parameters such as recommender algorithm, datasets, and metrics were also specified (see § 8.4).

8.4 Experimental Design

To address our research questions and hypothesis, we conducted three main experiments to correspond with the three final mitigation strategies (*MS1-MS3*) described in § 8.3:

- E1: Power User Removal
- E2: Power User Influence Reduction: All power users
- E3: Power User Influence Reduction: Selected power users

Evaluation Metrics – Evaluations were performed before and after the attacks. We use Mean Absolute Error (MAE) for accuracy and prediction coverage [20, 56] using a holdout-partitioned 70/30 train/test dataset. To compare MAE before and after attacks, we use $\delta MAE = MAE_{after} - MAE_{before}$. For robustness metrics [36, 9], we use Hit Ratio (HR), Average HR ($\overline{HR}$), Prediction Shift (PS), Average PS ($\overline{PS}$), Rank (R), and Average R ($\overline{R}$), where a high Hit Ratio and a low Rank after the attack indicate that the attack was successful (from the attacker's standpoint) assuming that the target item had a lower Hit Ratio and higher Rank before the attack. The top-N size for Hit Ratio calculations is N=40. To compare the effectiveness of the

mitigation strategies and to assess the trade-offs between accuracy and robustness, we developed a new metric called the Accuracy/Robustness/Mitigation measure (ARM), $ARM = (2 * \frac{MAE_{after} * \overline{HR}}{MAE_{after} + \overline{HR}}) * (1 - \rho)$, where ρ is the percentage of power users or influence being evaluated. ARM varies between 0 and 1 and a higher ARM value indicates a more effective mitigation for a given experiment.

The major motivation behind the ARM metric is to find a measure that determines the level of power removal or influence reduction that is best for mitigating the PUA. We know from power user ablation studies [60, 51, 61], that MAE tends to increase (get worse) as power users (or their influence) are removed from the RS dataset. Those same studies show that as power user attackers (or their influence) are removed from the dataset, Hit Ratio decreases (the attack becomes less effective). And there is evidence in this study that as power user attackers (or their influence) are removed/reduced, MAE tends to increase (get worse). So, we need a metric that can indicate how much power user removal or influence reduction can be used to mitigate a power user attack and still leave MAE and Hit Ratio as low as possible. The ARM metric combines MAE and Hit Ratio in such a way that it balances the increase in MAE with the reduction in Hit Ratio as power user influence is removed or reduced. It should also be noted that when the PUA being evaluated uses "New" target items (items with 1 rating), the Prediction Shift is expected to be close to the maximum rating as defined by the RS. For "New and Established" target items, the Prediction Shift may also be high because some of the SPUs may fall below the threshold of power users to be removed or have their influence reduced; the SPUs that remain after removal or influence reduction are still used in the attack and may contribute

to the high Prediction Shift.

Datasets and Algorithms – We used MovieLens[65] ML100K[66] and ML1M[67] datasets. The CF user-based weighted algorithm (UBW) [13] 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 [19]. We used UBW from Apache Mahout[68] and added functionality to implement the MS2 and MS3 strategies (see § 8.3).

Power User Selection – The InDegree (ID), NumRatings (NR), and AggSim(AS) methods described in § 8.2 were used.

Target Item Selection – Fifty target items with no more than one rating, regardless of their rating value, were selected randomly as "New" target items. We also used 50 "New and Established" target items, i.e., target items were selected randomly and had the following average number of ratings, average rating, and average rating entropy, respectively: ML100K (73.780, 3.133, 1.769), ML1M (280.399, 3.296, 1.883).

Attack Parameter Selection – The Attack Intent is Push, i.e., target item rating is set to max (= 5). The Attack Size or number of SPUs in each attack varied by dataset: 50 for ML100K and 300 for ML1M. Attack sizes, also expressed as $(\frac{\#attackers}{\#users} * 100)\%$, were selected based on previous research [36, 9], where a 5-10% attack size was shown to be effective; we use a 5% attack size for each dataset. Power user attack profiles were generated as described in [61] and target item ratings were injected at run time.

[65] http://www.grouplens.org
[66] nominal 100,000 ratings, 1,682 movies, and 943 users.
[67] nominal 1,000,209 ratings, 3,883 movies, 6,040 users.
[68] http://mahout.apache.org

Test Variations – To evaluate the final mitigation strategies, we used three experiments, one prediction algorithm, two datasets, three power user selection methods, two target item types, and eight attack sizes.

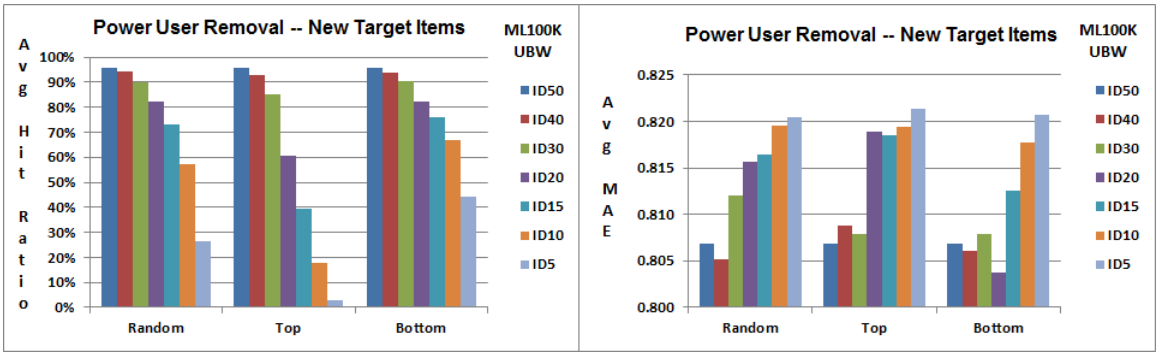

Figure 28: Hit Ratio and MAE as InDegree synthetic power users decrease from 50 to 5 using ML100K and UBW

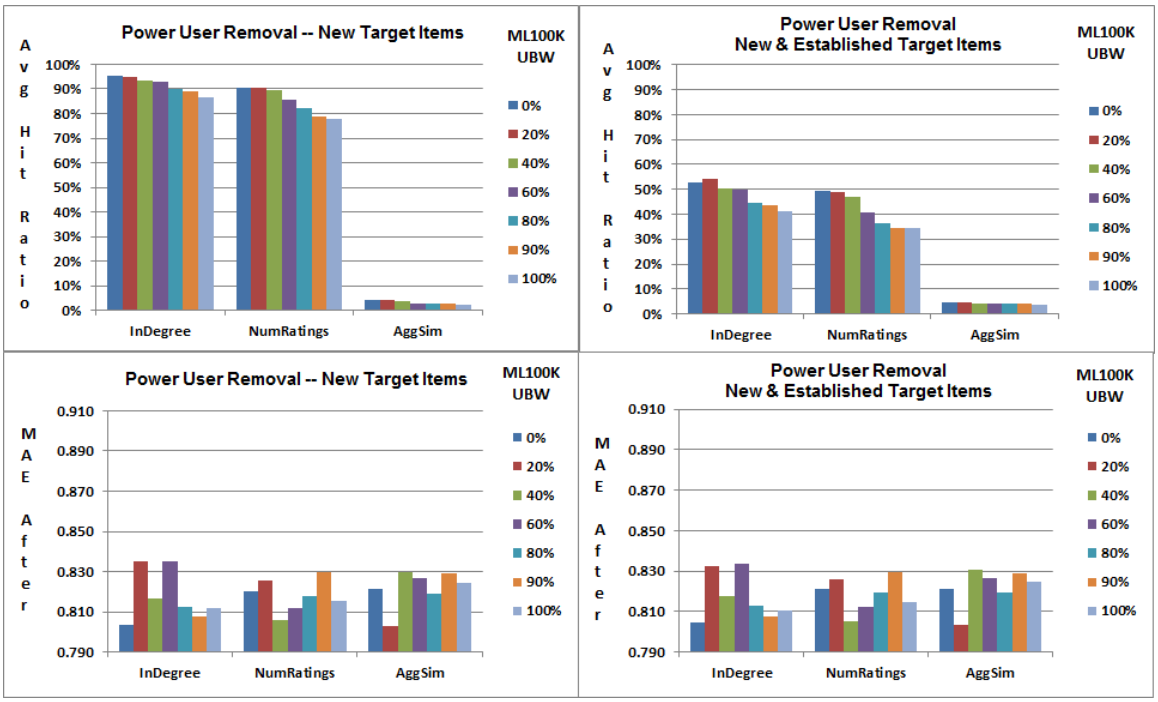

Figure 29: E1 – Hit Ratio and MAE as 0% to 100% of power users (real and synthetic) are removed using ML100K

Figure 30: E2 – Hit Ratio and MAE as power users' (real and synthetic) influence reduced from 1.0 to 0.0 using ML1M

8.5 Results and Discussion

(E1) Power User Removal – Consisted of removing power users from the dataset (incrementally from 0% to 100%) prior to recommendation calculations (similarity and prediction). We conducted a series of PUA's against the user-based CF algorithm. Each PUA in this experiment uses a dataset with a specified number of injected SPU attackers (§ 8.4). SPUs are generated based on three power user selection methods: InDegree (ID), NumRatings (NR), and AggSim (AS). Generated SPUs are added to the dataset for attack purposes and then the top-k power users (a mix of RPUs and a subset of the SPUs) selected by the three power user selection methods, are incrementally removed from the dataset; top-k = 50 for ML100K and 300 for ML1M. The SPUs are injected with either 50 New or 50 New and Established target items

at runtime to evaluate the PUA in separate trials (one target item attack at a time) and the HR/Rank/PS metrics are averaged across all 50 target items. Initially, we analyzed various SPU removal approaches: most-to-least-influential (Top), least-to most-influential (Bottom), and Random, see Figure 28. For E1, however, we chose the most-to-least-influential (Top) approach since that would better mitigate the attack effectiveness based on Hit Ratio, from a system operator's perspective.

Figure 29 shows the results for ML100K as the percentage of power users removed increases from 0% to 100% (0-50 power users). $\overline{HR}$ before the attack (not shown) is 0% for New target items and 2% for New and Established target items for ML100K across all power user removal levels. These values serve as the $\overline{HR}$ baseline and indicate that without attackers, the target items do not appear in any top-N lists of recommendations. The drop in $\overline{HR}$ is not as dramatic compared to results in Figure 28 because some SPU attackers remain in the dataset, i.e, they were below that power user selection threshold, and contribute to increasing the $\overline{HR}$. New and Established target items are more difficult to attack compared to New targets as evidenced by a lower $\overline{HR}$ in the right hand side of Figure 29. Removing 100% of the power users still leaves SPUs in the dataset, hence $\overline{HR}$ remains high; $\overline{HR}$ for the AS attack is not significant at any level of removal. The PUA is not as effective when using New and Established target items (this was expected). $\overline{PS}$ for ML100K and New target items was 4.9 for all power user selection methods across all removal levels; for New and Established target items $\overline{PS}$ was 4.5 for InDegree and NumRatings, across all removal levels; 3.0 for AggSim. $\overline{R}$ for ML100K and New target items varied between 3 and 4 for all power user selection methods across all removal levels; for

New and Established target items $\overline{R}$ varied between 13 and 15 across all selection methods and removal levels, indicating a less effective attack.

We observed similar results with ML1M (not shown) except that for ID, $\overline{HR}$ increased slightly as power users were removed, most likely due to the influence characteristics of the SPUs. $\overline{R}$ for New target items varied between 7 and 9, and for New and Established target items varied between 16 and 21 for all power user selection methods across all removal levels. The ARM measure (not shown) indicated that 100% removal is the best mitigation for all E1 attacks in Figure 29.

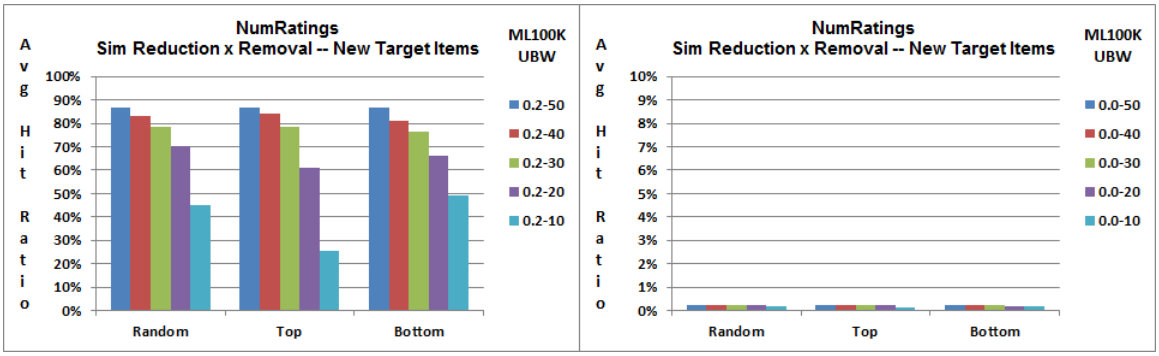

Figure 31: Examples of Hit Ratio impacts as SPU influence is reduced (0.2 & 0.0) and removed (50 to 10) using ML100K

(E2) Influence Reduction, all Power Users – Consisted of varying the similarity weighting (incrementally from 0.0 to 1.0) applied to power users (selected RPUs and SPUs) who are nearest neighbors during the prediction calculation. We conducted a series of PUA's against the user-based CF algorithm. Each PUA in this experiment uses a dataset with the same number of power users, i.e., there is no removal of power users in this experiment. $\overline{HR}$ before the attack (not shown) is 0% for New target items and 1% for New and Established target items for ML1M across all power user influence reduction levels. New target item results in Figure 30 for ML1M (left side of

Figure 32: E3 – Hit Ratio and MAE as 100% to 0% of power users' (real and synthetic) influence is applied using ML1M

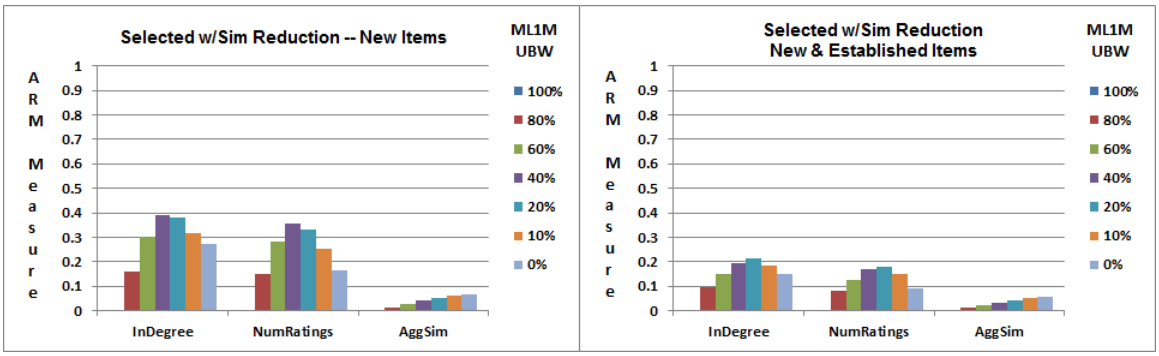

Figure 33: E3 – ARM Measure as 100% to 0% of power users' (real and synthetic) influence is applied using ML1M

charts) indicate that as similarity weighting is reduced from 1.0 to 0.1, $\overline{HR}$ remains flat for ID (81%), NR (74%), and AS (4%). When similarity weighting is set to zero, $\overline{HR}$ drops significantly for ID (to 16%) and for NR (to 9%), while remaining flat for AS (4%). And as similarity weighting is reduced from 1.0 to 0.1, MAE increases for ID and NR,and remains flat for AS. $\overline{R}$ (not shown) ranges between 9-14 over

all similarity weightings and selection methods. The interpretation for the small reduction in $\overline{HR}$ for ID and NR as similarity weighting is reduced from 1.0 to 0.1, can be attributed to the fact that the SPUs are, in most cases, the only users that have rated the New target items; therefore, the SPUs dominate the influence within the neighborhoods keeping $\overline{HR}$ high and $\overline{R}$ low. New and Established target results in Figure 30 for ML1M (right side) indicate that $\overline{HR}$ begins at a much lower level (40% for ID and NR, 4% for AS) and remains flat until similarity weighting drops to 0.0, mainly because SPU influence is not very dominant within the neighborhoods as many other users have rated established items. $\overline{R}$ (not shown) ranges between 14-22 over all similarity weightings and selection methods. The influence of power users (a mix of RPUs and some SPUs) can be observed in the significantly higher MAE results (less accuracy) when similarity weighting is set to zero, i.e., without the power user influence, accuracy becomes much worse. The low $\overline{HR}$ when similarity weighting is set to zero indicates that not all attackers (SPUs) have been removed from the prediction calculations. $\overline{PS}$ for ML1M and New target items was 4.9 for all power user selection methods across all reduction levels; for New and Established target items this averages 4.4 for InDegree, 4.2 for NumRatings, 3.8 for AggSim across all reduction levels.

We observed very similar results with ML100K (not shown) except that $\overline{HR}$ starts slightly higher (90%) and $\overline{R}$ ranges lower (4-6 for New, 15-17 for New and Established targets) for ID and NR. For all attacks in E2 (ML100K and ML1M), a similarity weighting reduction of 0.0 is required to significantly reduce $\overline{HR}$, at the expense of a higher MAE (lower accuracy). However, the ARM measure (not shown) indicated

that a similarity weighting reduction setting of 0.1 is the best mitigation for ID and NR, avoiding the spike in MAE_{after} albeit with high $\overline{HR}$, and 0.0 for AS.

In our initial analysis, we developed a hybrid mitigation approach that combined power user removal (E1) and similarity reduction (E2). We found that $\overline{HR}$ followed the E1 results across all similarity weighting values > 0.0 and we provide an example of results with 0.2 similarity weighting in Figure 31 (left); at similarity weighting $= 0.0$, $\overline{HR}$ was near zero in Figure 31 (right). Based on these results, a hybrid approach was not pursued any further. We also were concerned about coverage, i.e, % of items for which recommendations can be formed, as power users had their influence reduced. Our results indicated that coverage remained flat over the course of influence reduction and varied according to selection method used, ID (73%), NR (77%), and AS (65%).

(E3) Influence Reduction, selected Power Users – Each PUA in the experiment uses a dataset with the same number of injected SPUs (there is no power user removal in this experiment) and will only allow a percentage (incrementally from 0% to 100%) of them to be involved in the prediction calculation. The power users are selected randomly and will have a similarity weighting of 1.0 if selected and 0.0 if not selected, during the prediction calculation. $\overline{HR}$ before the attack (not shown) is 0% for New target items and 1% for New and Established target items for ML1M across all power user influence reduction levels. For ML1M with New target items shown in Figure 32 (left side), as the percentage of power users is reduced from 100% to 10%, $\overline{HR}$ decreases from 82% to 22% for ID and 74% to 17% for NR; for AS, $\overline{HR}$ remains flat at about 4%. When the percentage of power users is set to zero, $\overline{HR}$ goes to 16%

and 9% for ID and NR, and no change for AS. As the percentage of power users is reduced from 100% to 10%, MAE increases for ID and NR, remains flat (0.79) for AS. $\overline{PS}$ for ML1M and New target items was 4.9 for all power user selection methods across all reduction levels. $\overline{R}$ ranges between 9 and 17 over all power user percentages and selection methods; when the percentage of SPUs is 0.0, $\overline{R}$ is 14 for ID and NR, 11 for AS. For ML1M with New and Established targets, we observed a similar set of results except that $\overline{HR}$ begins at a lower level. As the percentage of power users is reduced from 1.0 to 0.1, $\overline{HR}$ decreases from about 40% to 10% for both ID and NR; for AS, $\overline{HR}$ remains flat at 3%. When the percentage of power users is set to zero, $\overline{HR}$ goes to 8% and 5% for ID and NR, and 3% for AS. As the percentage of power users is reduced from 1.0 to 0.1, MAE increases for ID and NR, remains flat (0.79) for AS. $\overline{PS}$ for ML1M and New and Established target items averages 4.4 for InDegree, 4.2 for NumRatings, 3.8 for AggSim across all reduction levels. $\overline{R}$ ranges between 13 and 25 over all power user percentages for ID and NR, and between 15 and 16 for AS; when the percentage of power users is 0.0, $\overline{R}$ is 14 for ID and NR, 15 for AS.

We observed very similar results with ML100K except that $\overline{HR}$ starts slightly higher (90%) and $\overline{R}$ ranges lower (3-6 for New, 13-18 for New and Established targets) for ID and NR. For New and New and Established target items, the ARM measure (see Figure 33) indicated that a percentage of power user reduction of 20-40% is the best mitigation for ID and NR (avoiding the larger values of MAE_{after}) and 0.0 for AS. ARM results for ML100K were very similar (not shown).

Based on our results and using the ARM metric, the mitigation strategy that best

balances accuracy and robustness for ID and NR PUAs is MS3; the AS PUA was not an effective attack in this study and did not require mitigation. For MS1, the ARM metric indicates 100% removal which leaves a very high Hit Ratio. And MS2 is marginally better than MS1, with the ARM metric indicating a similarity weighting of 0.1. Our hypothesis is accepted for MS3, partially accepted for MS2, and rejected for MS1.

8.6 Summary of this Chapter

This chapter evaluated power user attack mitigation approaches to address issues encountered when legitimate influential users (false positives) are removed along with attackers. We have shown that reducing similarity weighting during prediction calculation is an improvement over removal. We showed that there is a trade-off between accuracy (MAE) and robustness (Hit Ratio) when implementing mitigation strategies and have developed a metric to assist in evaluating this trade-off. Consistent with our previous work using user-based recommenders, we also showed that reducing the influence of power users contributes to a reduction in recommender system accuracy indicated by an increase in MAE; this shows how power users can impact recommendations.

With respect to the Dissertation Hypotheses provided in Section 1.5.2, this chapter has indicated the following level of support for the applicable hypotheses; final acceptance/rejection of the Dissertation Hypotheses are provided in the Dissertation Summary, Section 10.1:

DH-3: Power user attack profiles generated from characteristics of InDegree-selected power users will result in more effective attacks (from the attacker's viewpoint) than

attack profiles generated from characteristics of power users selected from other techniques across CF algorithms, datasets, and domains. Although this hypothesis was not tested explicitly, the E1 results reported in Section 8.5 indicate that the InDegree-based PUA shows higher Hit Ratio metrics than those of NumRatings and AggSim. Therefore, this hypothesis is supported for ML100K and ML1M using the UBW algorithm.

DH-4: A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects on RS predictions and top-N lists of recommendations across multiple power user selection techniques, collaborative filtering algorithms, datasets, and domains. Although this hypothesis was not tested explicitly, the E1 results reported in Section 8.5 indicate that the InDegree-based and NumRatings-based PUA, show significantly high Hit Ratio metrics $> 90\%$ for ML100K and $> 80\%$ for ML1M (not shown); on the other hand, results for AggSim indicate very little impact on robustness metrics . Therefore, this hypothesis is supported for InDegree and NumRatings using UBW algorithm and ML100K and ML1M.

DH-5: Reducing the influence of power users is a more effective and less impactful mitigation strategy than completely removing power users from the dataset. This hypothesis is supported for the "Power User Influence Reduction: Selected power users" strategy and was tested with the user-based algorithm on the ML100K and ML1M datasets.

CHAPTER 9: EVALUATING POWER USER ATTACKS ON COLLABORATIVE RECOMMENDER SYSTEMS USING YAHOO! MUSIC DATA

9.1 Introduction

In prior work, we have developed, implemented, and evaluated the Power User Model and the Power User Attack (PUA) across popular collaborative recommender algorithms and movie domain datasets (see Chapters 5, and 6). We have also studied the mitigation of power user attacks on user-based collaborative recommenders using movie domain datasets (see Chapter 8). Results from those studies indicate that the PUA is effective in negatively impacting the accuracy and robustness of collaborative recommender systems.

The research question motivating this Chapter is whether the PUA can also be successful (from the attacker's viewpoint) using data from another domain. Therefore, this analysis investigates the impacts of power user attacks on user-based collaborative recommenders using a music domain dataset from Yahoo! Music[69].

The research question for this analysis is:

RQ-1: Can the Power User Attack be successful (from the attacker's viewpoint) using data from a domain other than movies?

The hypotheses to be tested are:

H-1: The use of In-Degree Centrality to select a set of power users results in power

[69]R3 at http://webscope.sandbox.yahoo.com/catalog.php?datatype=r

users with higher influence than other selection techniques.

H-2: A significant percentage of synthetic user profiles generated from statistical characteristics of power users will be identified by selected power user selection techniques.

H-3: Power user attack profiles generated from characteristics of InDegree-selected power users will result in more effective attacks (from the attacker's viewpoint) than attack profiles generated from characteristics of power users selected from other techniques.

H-4: A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects on RS predictions and top-N lists of recommendations across multiple power user selection techniques.

H-5: Reducing the influence of power users is a more effective and less impactful mitigation strategy than completely removing power users from the dataset.

9.2 Power User Attack Background

In order to study RS attacks based explicitly on measures of influence, we previously defined a *Power User Attack* model as a set of power user profiles with biased ratings that influence the results presented to other users (see Chapter 5). The PUA relies critically on the method of power user identification/selection, so we also developed and evaluated a novel use of degree centrality concepts from social network analysis for identifying influential RS power users for attack purposes. In addition, we chose to use other heuristics because this would provide best-case and worst-case scenarios that we could use to compare with our degree centrality approach.

The power user selection methods used in this analysis are as follows:

- In-Degree Centrality: Users with the highest user-user in-degree values are se-

lected as power users.

- Aggregated Similarity: Users with highest user-user similarity correlation values are selected as power users.
- Number of Ratings: Users with the most number of ratings are selected as power users.

For more details on power user selection methods, please refer to the summary in Section 6.2.

9.3 Mitigation Strategies

Removing 100% of the power user attackers as a mitigation strategy results in (1) reduced coverage for the "removed" users, some of which could be legitimate users (false positives), and (2) reduced accuracy for users whose similarity neighborhoods no longer enjoy the influence of the "removed" users, some of which could be legitimate users (false positives) [32]. For live RSs, this could lead to dissatisfaction problems in both cases. This 100% removal approach also assumes that all (true) power user attackers will be detected, i.e., no provision is provided for attackers that are not detected (false negatives).

The mitigation strategies (MSs) for this study were described in Section 8.3 and consist of:

MS1: Remove selected power users incrementally from 0% to 100%, starting from most influential to least influential.

MS2: Reduce the similarity weighting factor of all selected power users incrementally from 1.0 to 0.0.

MS3: Reduce the similarity weighting of a percentage of all selected power users; the percentage of power users is varied incrementally from 100% to 0%. The similarity weighting is set to one if selected, zero otherwise.

To implement these mitigation strategies, the following methodology was used (see § 8.3):

1. Generate power user lists from selected datasets using power user selection techniques, including InDegree, NumRatings, and AggSim (see § 9.2). This generates a list of real power users (RPUs).
2. Generate synthetic power user (SPU) attack profiles based on power user statistical characteristics (see 6.3) and insert into the dataset. Select power users from the updated dataset using the power user selection techniques described in § 9.2. A top-N list of power users will, therefore, be a combination of RPUs and SPUs.
3. Select target items from a given dataset: New Items (items with one rating), New and Established Items (randomly-selected items with a range of popularity and likeability values).
4. Create incremental datasets with most-to-least-influential power users removed i.e., from the top.
5. Execute attacks for each mitigation strategy for the selected target items and calculate averaged metrics over all target items. Only SPUs will be used for attack purposes, leaving RPUs to provide their influence but not be part of the attack. Note that some SPU attack profiles will remain in the dataset after power users are removed during our experiments as described in § 9.4.

6. Compare accuracy, coverage, and robustness metrics for variations of the mitigation strategy to determine impacts of removing and reducing influence of power users.

Other parameters such as recommender algorithm, datasets, and metrics were also specified (see § 9.4).

9.4 Experimental Design

We conducted four experiments to test the five hypotheses (*H-1* through *H-5)* described in § 9.1:

- E1: Power User Ablation
- E2: Synthetic Power User Identification
- E3: Power User Attack Effectiveness
- E4: Power User Attack Mitigation
  - E4-M1: Power User Removal (MS1)
  - E4-M2: Power User Influence Reduction: All power users (MS2)
  - E4-M3: Power User Influence Reduction: Selected power users (MS3)

Evaluation Metrics – Evaluations were performed before and after the attacks. We use Mean Absolute Error (MAE) for accuracy and prediction Coverage, i.e., the percentage of users for which the system can form a recommendation [20, 56], using a holdout-partitioned 70/30 train/test dataset. To compare MAE before and after attacks, we use $\delta MAE = MAE_{after} - MAE_{before}$. For robustness metrics [36, 9], we use Hit Ratio (HR), Average HR ($\overline{HR}$), Prediction Shift (PS), Average PS ($\overline{PS}$), Rank (R), and Average R ($\overline{R}$), where a high Hit Ratio and a low Rank indicate that the attack was successful (from the attacker's standpoint). To compare the effective-

ness of the mitigation strategies and to assess the trade-offs between accuracy and robustness, we developed a new metric called the Accuracy/Robustness/Mitigation measure (ARM),

$ARM = (2 * \frac{MAE_{after} * \overline{HR}}{MAE_{after} + \overline{HR}}) * (1 - \rho)$, where ρ is the percentage of power users or influence being evaluated. Higher ARM indicates a more effective mitigation.

Datasets and Algorithms – We used the Yahoo! Music R3 dataset and called it Y365K[70] in this study. The CF user-based weighted algorithm (UBW) [13] 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 [19]. We used UBW from Apache Mahout[71] and added functionality to implement the MS2 and MS3 strategies (see § 9.3). We also added functionality to Mahout to implement the user-based mean-centered prediction algorithm (UMCP) [45].

Power User Selection – The InDegree (ID), NumRatings (NR), and AggSim(AS) methods described in § 9.2 were used. For experiments E1 and E2, we also used a randomized method (Rand) of selecting power users, for comparison purposes. In all cases, for top-N users and neighbors, we use N=50.

Target Item Selection – In previous work (see Chapters 5 and 6) we have randomly selected 50 target items with no more than one rating, regardless of their rating value, as "New" item targets. The rationale for using new items as targets is that they are easier to attack and, hence, produce a better signal. However, the Y365K dataset has a minimum of 66 ratings per item, so we used 50 "New and Established" items, i.e.,

[70] 365,704 ratings, 1,000 songs, 15,400 users, and 97.62% sparsity.
[71] http://mahout.apache.org

target items were selected randomly and had the following average number of ratings, average rating, and average rating entropy, respectively: 350.38, 2.522, 1.086. Our expectation was that attacking "New and Established" target items will produce a weaker signal. For experiments with power user attacks (E3 and E4), each test variation was executed once for each of the 50 target items and data results were averaged over the 50 target items.

Attack Parameter Selection – The Attack Intent is Push, i.e., target item rating is set to max (= 5). The Attack Size or maximum number of SPUs in each attack was 770 for Y365K. The attack size was selected based on previous research [36, 9], where a 5-10% attack was shown to be effective; we chose a 5% attack size for each dataset, where 770 is 5% of 15,400 users. The power user attack profiles were generated as described in Section 6.3 and the target item rating was injected at run time. For experiments with power user attacks (E3 and E4), each test variation was executed with a percentage of the maximum number of power users, incrementally from 0% to 100%.

9.5 Results and Discussion

9.5.1 Initial Investigation

Before the start of this experimentation, we investigated the use of various Yahoo! Music datasets. We began with a large dataset[72] consisting of 717 million ratings of 136 thousand songs given by 1.8 million users of Yahoo! Music services. We extracted two smaller datasets from R2, Y9M[73] and Y1M[74] to match the sizes of MovieLens

[72] R2 at http://webscope.sandbox.yahoo.com/catalog.php?datatype=r
[73] 8,846,899 ratings, 136,736 songs, 23,179 users, and 99.72% sparsity.
[74] 1,002,415 ratings, 126,038 songs, 2,717 users, and 99.71% sparsity.

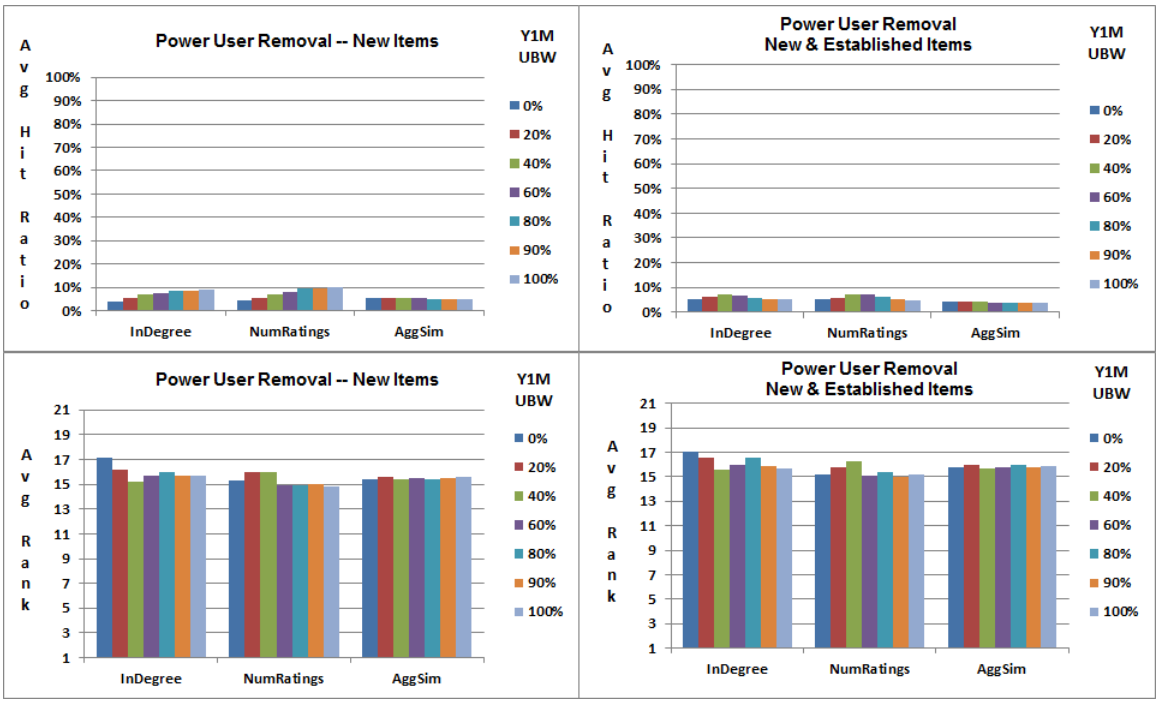

Figure 34: Hit Ratio and MAE as 0% to 100% of real power users are removed using UBW and Y1M

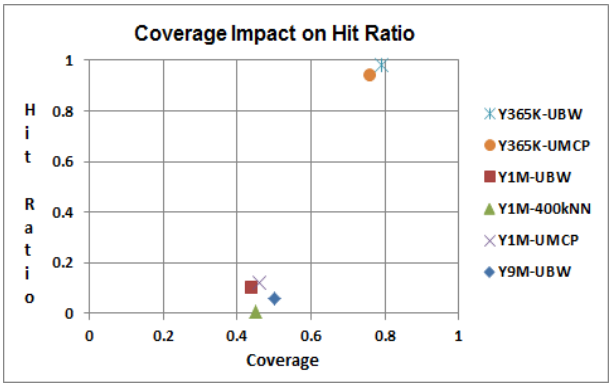

Figure 35: Prediction Coverage and Hit Ratio correlation for power user attacks on Yahoo! Music datasets

datasets of similar ratings size. We set up PUAs using synthetic power users generated with the power user selection methods described in § 9.2 and user-based collaborative algorithms (UBW and UMCP). Even in the best-case scenarios (from an attacker's standpoint), the attacks resulted in $\overline{HR} < 12\%$ and $\overline{R} >= 15$, e.g., see Figure 34. We were concerned that the Yahoo! Music datasets would not have the same accu-

racy/robustness profiles exhibited during a PUA that we had seen with MovieLens datasets. We conjectured that, because of the 99%+ sparsity of the datasets, we would need larger neighborhood sizes and experimented with various settings from 50 to 400; however, even with kNN=400 (labeled "Y1M-400kNN" in Figure 35) for UBW, we did not see a change in the attack metrics. We then turned our attention to the datasets' prediction Coverage and its impact on Hit Ratio. We observed a correlation between a given level of Coverage and the corresponding Hit Ratio, i.e., low Coverage ($< 50\%$) correlated with low Hit Ratio ($< 12\%$) and determined that we need to use a dataset with higher Coverage values. We then selected the smaller R3 dataset from Yahoo! Music and called it Y365K in our experiments. After executing initial power user attacks using Y365K, we found that higher Coverage (>70%) correlated with higher Hit Ratio (>90%) results, as shown in Figure 35. Although we are not claiming this as a general rule, it appears that a minimum amount of prediction coverage is required (between 50% and 70%) in order to have successful power user attacks.

9.5.2 E1: Power User Ablation

In previous work (see Chapters 5 and 6) we described how power user influence can be illustrated by observing the change in system accuracy (MAE) as power users are removed from the system. Our ablation studies showed that as power users were removed from the dataset, MAE increased; i.e., the system became less accurate. This was the case using our chosen power user selection methods, InDegree, NumRatings, and AggSim (see § 9.2), with multiple collaborative algorithms and MovieLens datasets. In this experiment, we repeat the ablation analysis using the

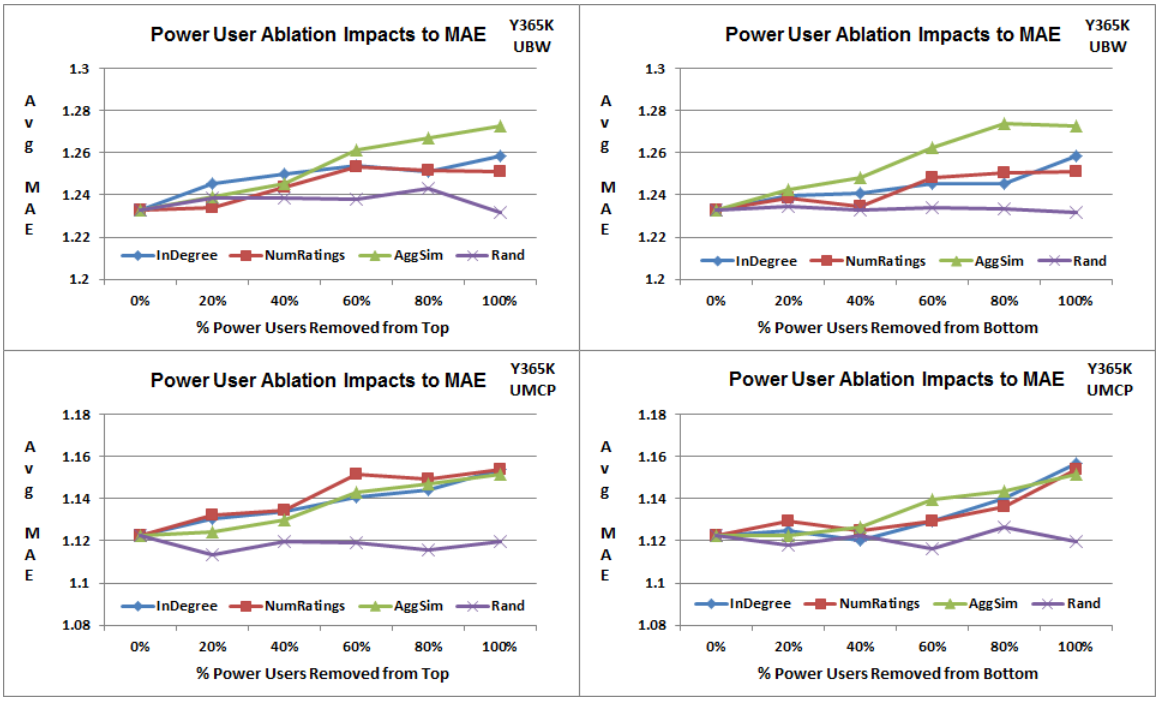

Figure 36: E1 - MAE as 0% to 100% of real power users are removed using UBW, UMCP, and Y365K

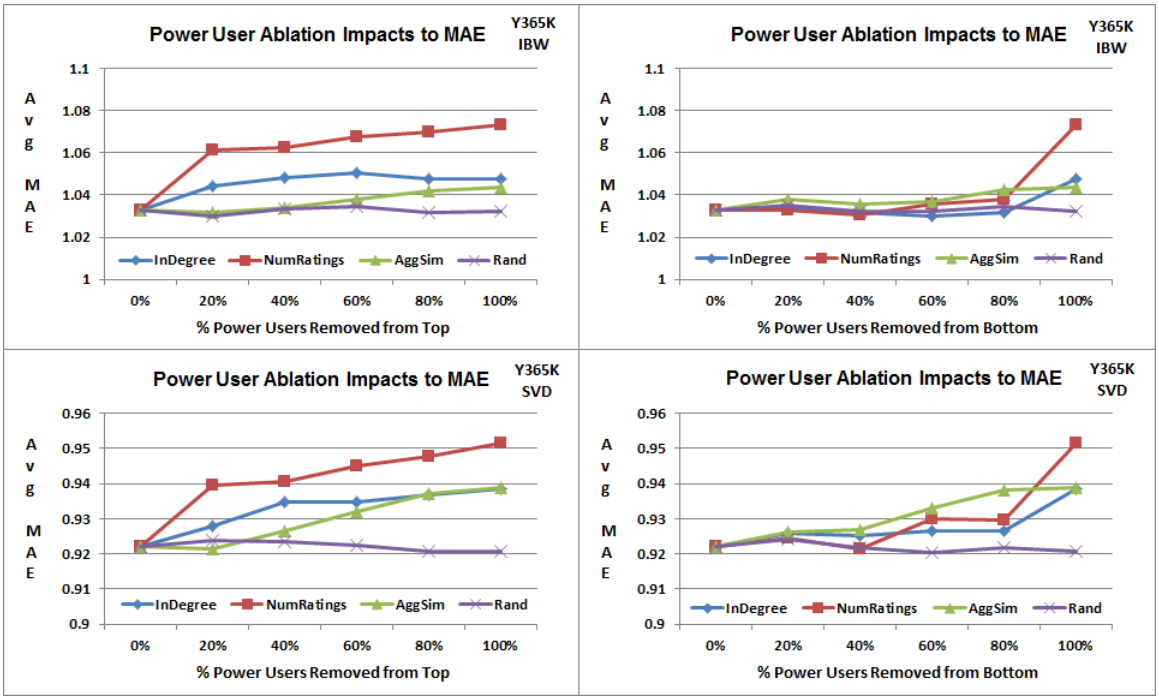

Figure 37: E1 - MAE as 0% to 100% of real power users are removed using IBW, SVD, and Y365K

Y365K dataset and four popular collaborative filtering algorithms (§ 9.4); we added a fourth selection method called Random (Rand) to compare with the other three methods. In addition, we ran the ablation by removing power users "from the top" (most influential to least influential) and "from the bottom" (least influential to most influential) to see the accuracy impacts this would produce. Results are shown in Figures 36 and 37.

The top two charts in Figure 36 show results for UBW and the bottom two charts show results for UMCP; the left two charts show results as power users are removed "from the top" and the right two charts show results as power users are removed "from the bottom". For UBW, AggSim MAE is significantly higher than InDegree, NumRatings, and Rand at 100% removal ($p < 0.01$) indicating that the AggSim influence is superior. There is no significant difference between InDegree and NumRatings at 100% removal, indicating that these two selection methods select power users of equal influence; also, InDegree and NumRatings MAE at 100% removal is significantly higher than Rand ($p < 0.01$) indicating higher influence impact. For UMCP, there is no difference between AggSim, InDegree, and NumRatings at 100% removal indicating that all three methods have the same level of influence; results also indicate that the AggSim, InDegree, and NumRatings methods perform significantly better than Rand at 100% removal ($p < 0.01$) indicating that selecting power users with the AggSim, InDegree, and NumRatings methods results in more influential users than selecting users at random. For UBW and UMCP, for each power user selection method except Rand, there is a significant increase in MAE comparing 0% to 100% removal ($p < 0.01$); for Rand, there is no significant difference indicating

that randomly selected power users are removed, they not influential enough to vary the system accuracy. And as expected for each algorithm, MAE for 0% power users removed is the same regardless of whether removal is from the top or bottom, 1.23 for UBW and 1.12 for UMCP. This difference indicates that the accuracy of the UMCP algorithm is significantly better ($p < 0.01$) than UBW.

The top two charts in Figure 37 show results for IBW and the bottom two charts show results for SVD; the left two charts show results as power users are removed "from the top" and the right two charts show results as power users are removed "from the bottom". For IBW, NumRatings MAE is significantly higher than InDegree, AggSim, and Rand at 100% removal ($p < 0.01$) indicating that the NumRatings influence is superior. There is no significant difference between InDegree and AggSim at 100% removal, indicating that these two selection methods select power users of equal influence. For SVD, NumRatings MAE is significantly higher than InDegree, AggSim, and Rand at 100% removal ($p < 0.01$) indicating that the NumRatings influence is superior. There is no significant difference between InDegree and AggSim at 100% removal, indicating that these two selection methods select power users of equal influence. Results also indicate that the AggSim, InDegree, and NumRatings methods perform significantly better than Rand at 100% removal ($p < 0.01$) indicating that selecting power users with the AggSim, InDegree, and NumRatings methods results in more influential users than selecting users at random. For IBW and SVD, for each power user selection method except Rand, there is a significant increase in MAE comparing 0% to 100% removal ($p < 0.01$); for Rand, there is no significant difference indicating that when randomly selected power users are removed, they are

not influential enough to vary the system accuracy. And as expected for each algorithm, MAE for 0% power users removed is the same regardless of whether removal is from the top or bottom, 1.032 for IBW and 0.922 for SVD. This difference indicates that the accuracy of the SVD algorithm is significantly better ($p < 0.01$) than IBW.

The H-1 hypothesis is rejected for user-based, item-based, and SVD-based collaborative recommenders using the Y365K dataset because AggSim and NumRatings were able to select a better set of power users than InDegree. NumRatings had the best performance for item-based and SVD-based recommenders and AggSim had the best performance for user-based weighted recommenders. All three of these power user selection methods had the same level of performance for user-based mean-centered recommenders. In all cases, InDegree, NumRatings, and AggSim provided a more influential power user selection than randomly selected power users.

9.5.3 E2: Synthetic Power User Identification

To evaluate the SPU profiles (before the attack), we removed the top 770 RPUs from the original Y365K dataset using each of the selection methods (InDegree, NumRatings, AggSim, and Rand) and replace them with 770 SPU profiles to create modified Y365K datasets.[75] We remove the RPUs to see how well the SPUs would replace them. Then, we identify/select the top 770 power users from the modified datasets using each of the three selection methods. In all cases, the top power users consisted of a mix of RPUs and SPUs

To determine the extent to which the 770 SPU's are actually selected by each

[75]NB: The desired attack size (5% of users in the dataset) is equivalent to 770 SPUs for Y365K; the same number of SPU profiles are evaluated before and after the attack.

method, we use precision and recall metrics. We observe that the power user model generated SPU profiles with varying degree of success based on the power user selection method used. For InDegree, 59.9% of the SPU's were identified and NumRatings achieved 78.1% precision and recall scores, while AggSim was only able to achieve a 13.1% precision and recall score and Rand achieved a 6.0% score. As a precedent for comparison, in Chapter 6 we achieved precision/recall scores of 70%, 83%, and 32% for InDegree, NumRatings, and AggSim, respectively. To evaluate the hypothesis for this experiment, we expect that a majority of the SPU profiles injected into a given dataset will be successfully identified by the same power user selection method used to identify the respective RPU profiles, i.e., precision and recall scores will be $>$ 50%. Therefore, Hypothesis H-2 is accepted for InDegree and NumRatings, rejected for AggSim and Rand methods, meaning that the power user model generated an acceptable number of SPUs that were successfully identified/selected by the InDegree and NumRatings methods and not the AggSim or Rand methods.

9.5.4 E3: Power User Attack Effectiveness

Two types of power user attacks were used in this experiment, based on previous PUA analyses (see Chapters 5 and 6). In one case, we select RPUs using power user selection techniques (see § 9.2); in the second case, we generate SPUs, replace the RPUs with the SPUs, and then select power users using the same power user selection techniques (see § 9.2) as in the first case. In both cases, the PUA is mounted with a varying percentage, from 0% to 100%, of power user attackers removed from the dataset; in the first case, RPUs are the attackers and in the second case, SPUs are the attackers. We also use New and Established item targets for both attacks. These

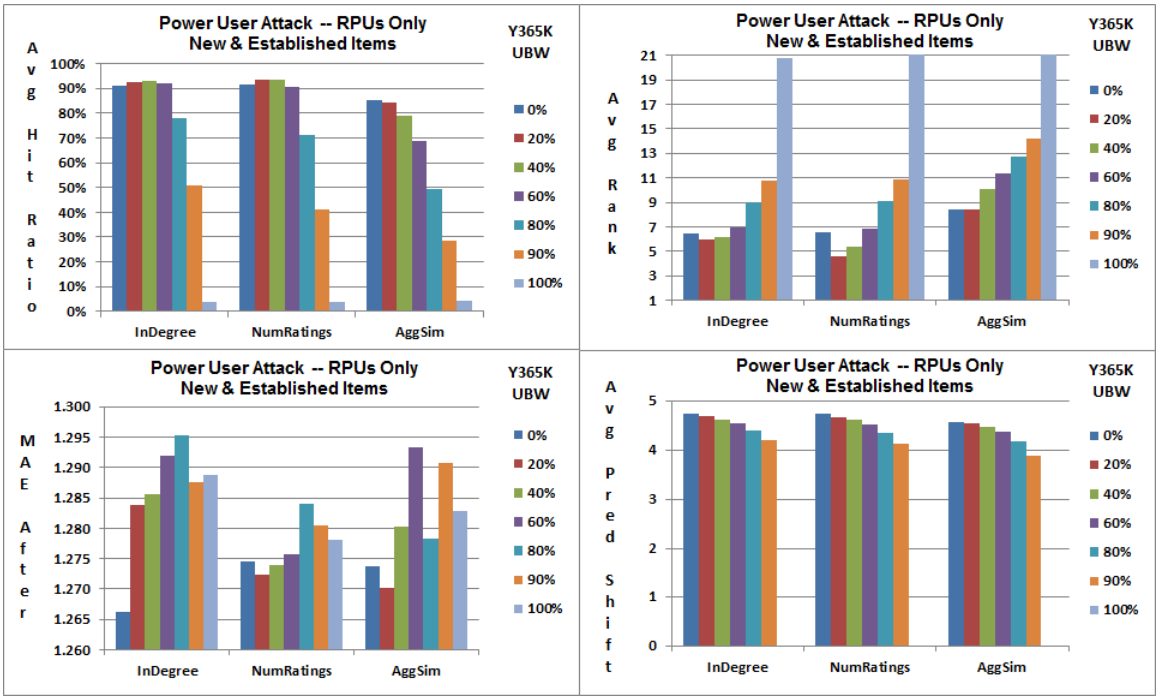

Figure 38: E3 - accuracy and robustness metrics for an RPU-based PUA using UBW and Y365K

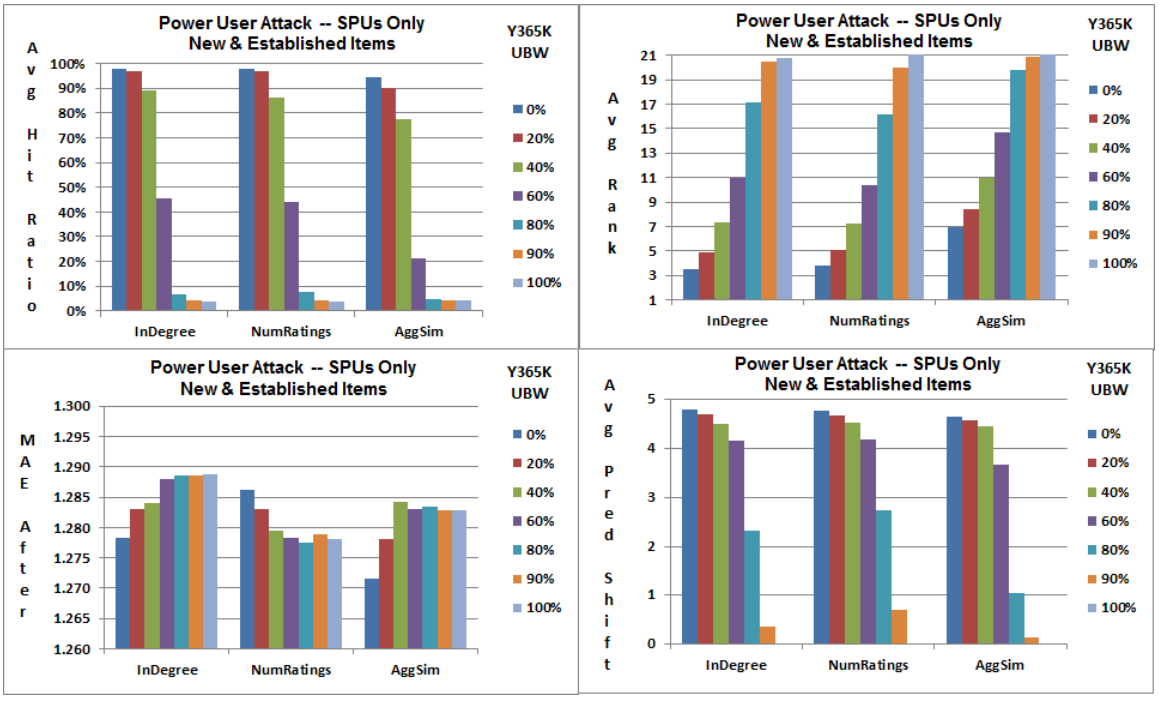

Figure 39: E3 - accuracy and robustness metrics for an SPU-based PUA using UBW and Y365K

two approaches allow us to more broadly explore the impacts of the PUA on the Y365K dataset and to compare with previously observed results. The results of these analyses are shown in Figures 38 and 39.

In Figure 38, we see that for InDegree and NumRatings, $\overline{HR}$ (upper left) remains at a 90% level with up to 60% of the power users removed and $\overline{R}$ is no higher than 7. For AggSim, these metrics are slightly more subdued. In many cases, MAE_{after} increases from its value at 0% removal indicating the impact of power user influence and $\overline{PS}$ remains at a high level (> 4) indicating the high level of influence of this attack. At 100% power user removal, $\overline{HR}$ is at 4%, $\overline{R}$ at 21, and $\overline{PS}$ drops to zero indicating that without the power users attacking, there still is an impact on the target items from other users, albeit minimal.

In Figure 39, we see that for InDegree and NumRatings, $\overline{HR}$ (upper left) remains at about a 90% level with up to 40% of the power users removed and $\overline{R}$ is no higher than 11. For AggSim, these metrics are slightly more subdued. In some cases, MAE_{after} increases from its value at 0% removal indicating a reduction of power user influence and $\overline{PS}$ remains at a high level (> 4) indicating the high level of influence of this attack. For NumRatings, we see that both $\overline{HR}$ and MAE_{after} decrease which would indicate that removing NumRatings power users improves the accuracy and robustness characteristics of the system. At 100% power user removal, $\overline{HR}$ is at 4%, $\overline{R}$ at 21, and $\overline{PS}$ drops to zero indicating that without the power users attacking, there still is an impact on the target items from other users, albeit minimal.

Based on these results, Hypothesis H-3 is partially accepted for InDegree since it is better than AggSim albeit virtually tied with NumRatings in terms of attack

effectiveness.

Given the high levels of $\overline{HR}$ and low levels of $\overline{R}$ in Figures 38 and 39, Hypothesis H-4 is accepted for InDegree and NumRatings. For the RPU-based attack, an attack size of 2% shows major impacts to accuracy and robustness; for an SPU-based attack, a 3% attack shows major impacts.

9.5.5 E4: Power User Attack Mitigation

(E4-M1) Power User Removal: This experiment consisted of removing power users from the dataset (incrementally from 0% to 100%) prior to recommendation calculations (for similarity and prediction). We conducted a series of PUA's against the user-based CF algorithm. Each PUA in this experiment uses a dataset with a specified number of injected SPUs (§ 9.4); for Y365K, we use up to 770 SPUs. The SPUs are generated based on three power user selection methods described in § 9.2: InDegree, NumRatings, and AggSim. The SPUs are injected with target items at runtime; the 50 targets are evaluated one at a time for the PUA and the HR/Rank/PS metrics are averaged across all 50 target items.

In previous work (see Chapter 8), we analyzed various SPU removal approaches: most-to-least-influential (Top), least-to most-influential (Bottom), and Random. For E4-M1, power users are removed from most-to-least-influential (from the top) since that would better mitigate the attack effectiveness based on Hit Ratio, from a system operator's perspective. Generated SPUs are added to the dataset for attack purposes and then the top-N RPUs/SPUs, as selected by the three power user selection methods, are incrementally removed from the dataset; top-N = 770 for Y365K. The SPUs are injected with either 50 "New" or "New and Established" target items at

runtime to evaluate the PUA in separate trials (one target item attack at a time) and the HR/Rank/PS metrics are averaged across all 50 target items. Figure 40 shows the results for Y365K as the percentage of power users removed increases from 0% to 100% (0-50 power users); the left side shows power user removal for "New" item targets and the right side shows "New and Established" targets. The reason they are similar is that the "New" targets in the Y365K dataset have a minimum of 66 ratings, so the contrast with "New and Established" targets is not as significant as what we had observed in previous work using target items with only 1 rating (see Chapter 8).

Removing 100% of the RPUs/SPUs still leaves SPUs in the dataset, hence $\overline{HR}$ for InDegree remains relatively high and is reduced to 30% for NumRatings. $\overline{HR}$ for the AggSim attack remains high for all levels of removal because the AggSim power user selection method did not find any SPUs in the top 770 power users, so none are available for removal. The ARM measure indicated that 100% removal is the best mitigation for InDegree and AggSim power user selection methods, and 80% for NumRatings.

(E4-M2) Influence Reduction, all Power Users – Consisted of varying the similarity weighting (incrementally between 0.0 to 1.0) applied to power users (selected RPUs and SPUs) who are nearest neighbors during the prediction calculation. We conducted a series of PUA's against the UBW algorithm. Each PUA in this experiment uses a dataset with the same number of injected RPUs/SPUs; there is no removal of power users in this experiment. Furthermore, because we did not observe much difference between New targets and New and Established targets in E4-M1 experiment, we decided to only test with New and Established targets in E4-M2 and E4-M3.

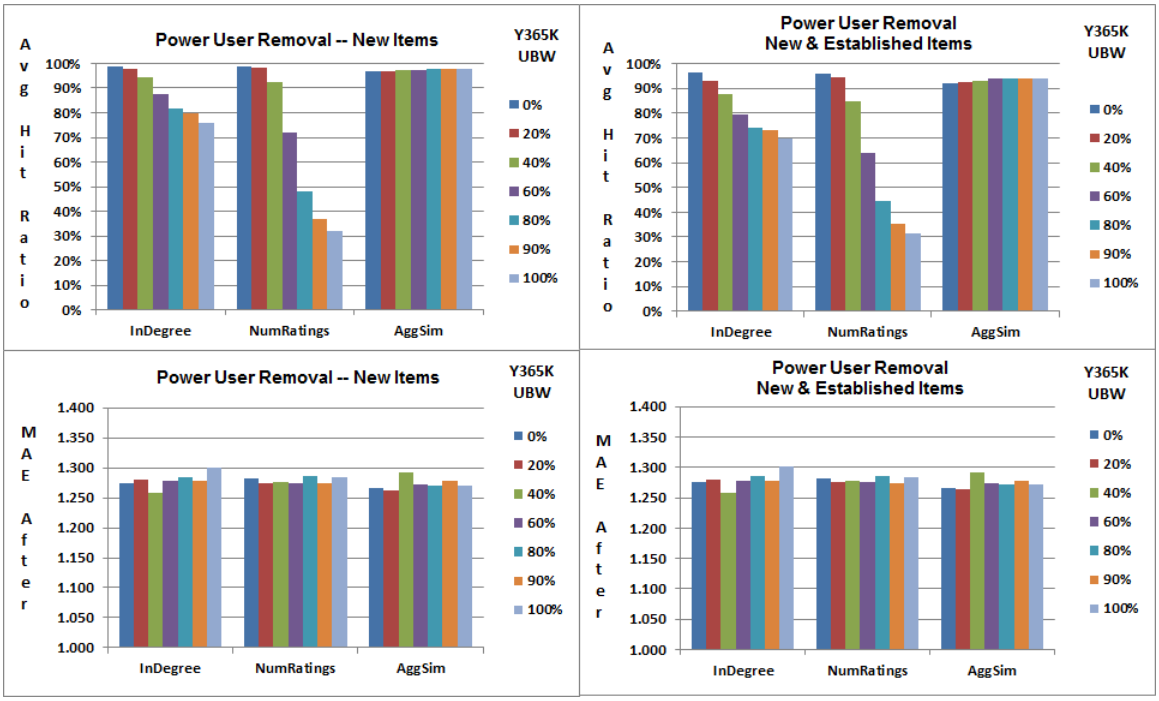

Figure 40: E4-M1 – Hit Ratio and MAE as 0% to 100% of power users (real and synthetic) are removed using UBW and Y365K

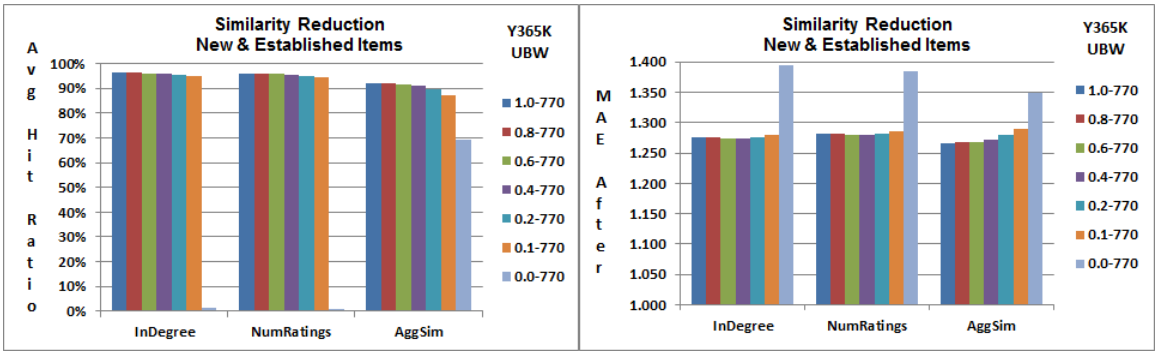

Figure 41: E4-M2 – Hit Ratio and MAE as power users' (real and synthetic) influence reduced from 1.0 to 0.0 using Y365K

New and Established target item results in Figure 41 for Y365K indicate that AggSim similarity weighting is reduced from 1.0 to 0.1, $\overline{HR}$ remains flat for InDegree (94-96%), NumRatings (94-96%), and AggSim (88-92%). When similarity weighting is set to zero, $\overline{HR}$ drops significantly for InDegree and NumRatings (to 1%), while remaining high for AggSim (69%). The $\overline{HR}$ for AggSim remains high because the

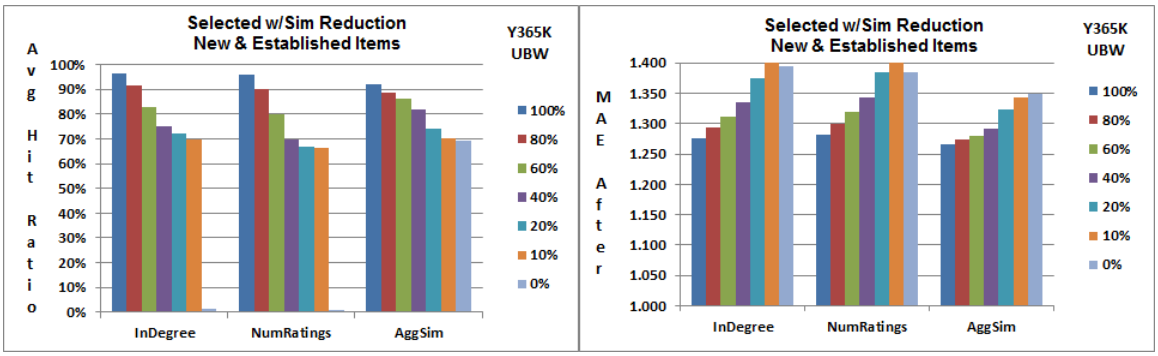

Figure 42: E4-M3 – Hit Ratio and MAE as 100% to 0% of power users' (real and synthetic) influence is applied using Y365K

AggSim power user selection method did not find any SPUs in the top 770 power users that were removed; this result indicates that the RPUs contribute somewhat to the attack, perhaps by correlating well with the SPU attackers. As similarity weighting is reduced from 1.0 to 0.1, MAE_{after} increases slightly for InDegree, NumRatings, and AggSim; MAE_{after} increases (accuracy gets worse) when all the power users are removed as observed in E1. The $\overline{R}$ ranges between 6-9 over all similarity weightings except 0.0 and selection methods; for a similarity weighting of 0.0, $\overline{R}$ increases to 13 for InDegree and AggSim, and to 17 for NumRatings. For New and Established target items, the ARM measure indicated that a similarity weighting reduction setting of 0.1 is the best mitigation for all three power user selection methods, avoiding the spike in MAE_{after}; however, the $\overline{HR}$ results remain high at 90% and provides little attack mitigation from a robustness perspective.

(E4-M3) Influence Reduction, selected Power Users – Each PUA in the experiment uses a dataset with the same number of injected SPUs (there is no power user removal in this experiment) and will only allow a percentage (incrementally from 0% to 100%) of them to be involved in the prediction calculation. The RPUs/SPUs are selected

randomly and will have a similarity weighting of 1.0 if selected and 0.0 if not selected, during the prediction calculation. Furthermore, because we did not observe much difference between New targets and New and Established targets in E4-M1 experiment, we decided to only test with New and Established targets in E4-M2 and E4-M3.

For Y365K with New and Established target items shown in Figure 42, as the percentage of RPUs/SPUs is reduced from 100% to 10%, $\overline{HR}$ decreases 96-69% for InDegree, 96-66% for NumRatings, and 92-70% for AggSim. When the percentage of RPUs/SPUs is set to zero, $\overline{HR}$ goes to 1% for InDegree and NumRatings, and to 69% for AggSim. As the percentage of RPUs/SPUs is reduced from 100% to 10%, MAE_{after} increases 1.28-1.4 for InDegree and NumRatings, and 1.27-1.34 for AggSim. The $\overline{R}$ ranges between 7 and 20 over all RPU/SPU percentages greater than 0.0 and all selection methods; when the percentage of RPUs/SPUs is 0.0, $\overline{R}$ is 13 for InDegree and AggSim, 17 for NumRatings. The ARM measure indicated that a similarity weighting reduction setting of 10% is the best mitigation for InDegree and NumRatings (avoiding the larger values of MAE_{after}) and 0.0 for AggSim; however, the $\overline{HR}$ results remain high at 70% and provides a small measure of attack mitigation from a robustness perspective.

Based on these results, it appears that the mitigation strategy that best balances accuracy and robustness is MS1 for NumRatings; MS3 for all power user selection methods can be considered as a distant second choice. For MS3, the ARM metric indicates a solution which leaves a relatively high Hit Ratio and vulnerability to attack. Our hypothesis cannot be accepted for MS2 and MS3 strategies.

9.6 Summary of this Chapter

This chapter evaluated power user selection methods, power user attacks, and power user attack mitigation approaches with a Yahoo! Music ratings dataset to contrast with previous evaluation results obtained using MovieLens movie ratings datasets (see Chapters 5 and 6). Except for the findings in E1, we observed similar results between the movie and music domain datasets.

We showed, through an ablation approach, that removing the influence of power users causes a reduction in recommender system accuracy indicated by an increase in MAE. This means that power users contribute positively to the accuracy of the system. Unlike previous work, the InDegree method did not perform as well as the NumRatings and AggSim methods for power user selection effectiveness. For InDegree and NumRatings power user selection methods, we showed that our synthetic power user generation method is effective not only from a selection perspective but also from an attack perspective. We determined, as we have in previous work, that a small number of power user attackers (less than 5% of all users) can have significant effects on the RS recommendations. We evaluated power user attack mitigation approaches to address issues encountered when legitimate influential users (false positives) are removed along with attackers. We have shown that reducing similarity weighting during prediction calculation is an improvement over removal. We showed that there is a trade-off between accuracy (MAE) and robustness (Hit Ratio) when implementing mitigation strategies and have developed a metric to assist in evaluating this trade-off. Consistent with our previous work using user-based recommenders, we also showed

that reducing the influence of power users contributes to a reduction in recommender system accuracy indicated by an increase in MAE; this shows how the influence of power users can impact recommendation accuracy.

With respect to the Dissertation Hypotheses provided in Section 1.5.2, this chapter has indicated the following level of support for the applicable hypotheses; final acceptance/rejection of the Dissertation Hypotheses are provided in the Dissertation Summary, Section 10.1:

DH-1: The use of In-Degree Centrality to select a set of power users results in power users with higher influence than other selection techniques, across multiple datasets and domains. This hypothesis is not supported for user-based, item-based, and SVD-based collaborative recommenders using the Y365K dataset because AggSim and NumRatings were able to select a better set of power users than InDegree. NumRatings had the best performance for item-based and SVD-based recommenders and AggSim had the best performance for user-based weighted recommenders. All three of these power user selection methods had the same level of performance for user-based mean-centered recommenders. In all cases, InDegree, NumRatings, and AggSim provided a more influential power user selection than randomly selected power users.

DH-2: A significant percentage of synthetic user profiles generated from statistical characteristics of power users will be identified by selected power user selection techniques across multiple datasets and domains. This hypothesis is supported for InDegree and NumRatings, and rejected for AggSim using Y365K.

DH-3: Power user attack profiles generated from characteristics of InDegree-selected power users will result in more effective attacks (from the attacker's viewpoint) than

attack profiles generated from characteristics of power users selected from other techniques across CF algorithms, datasets, and domains. This hypothesis is partially supported because InDegree had more effective attacks than AggSim, however, the results showed that InDegree and NumRatings were equally effective for Y365K using user-based recommenders.

DH-4: A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects on RS predictions and top-N lists of recommendations across multiple power user selection techniques, collaborative filtering algorithms, datasets, and domains. This hypothesis is supported for InDegree, NumRatings, and AggSim given that they all had Hit Ratio values greater than 80%; in some cases, we observed significant impacts to robustness metrics with attack sizes that were less than 5%. These tests were conducted with Y365K and user-based recommenders.

DH-5: Reducing the influence of power users is a more effective and less impactful mitigation strategy than completely removing power users from the dataset. This hypothesis is not supported for the "Power User Influence Reduction" strategies tested with the user-based algorithm on the Y365K dataset.

CHAPTER 10: DISSERTATION SUMMARY

The problem with attacks on recommender systems is that they bias the underlying data and cause the system to deliver erroneous or misleading recommendations to online users. This can cause users to lose trust in the system and either (1) shop elsewhere, negatively impacting the sales of the attacked service/product provider, or (2) purchase the product only to find out that it does not meet their needs, negatively impacting user satisfaction with the online recommender. With the significant growth in e-commerce in the last few years, this is a major problem that needs to be studied. There is abundant evidence in the media regarding the negative impacts that attacks on recommender systems can have on consumer behavior and the concomitant negative effects on system and service/product providers.

The research effort in this dissertation was focused on analyzing recommender system power users, how they are identified, selected and evaluated; how they are characterized; how a novel Power User Attack is configured, executed, and evaluated; and how power user attacks can be mitigated. In addition, we investigated a new, complementary attack model, the Power Item Attack, that uses *influential items* to successfully attack RSs. We showed that the Power Item Attack 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. Furthermore, these evaluations were conducted using user-based, item-based, and SVD-based collabo-

rative filtering algorithms using a production-level platform (Mahout) and publicly-available synthetic datasets in the movie and music domains.

This research is motivated by the concern that recommender systems continue to be vulnerable to attack and that, although several attack models have been developed in the past, users with malicious intent continue to find new ways to bias predictions and disrupt the system. The novel Power User Attack, presented here and inspired by social network analysis, is a new attack model that was shown to be capable of impacting the accuracy of the system's recommendations. The novel Power Item Attack, presented here and complementary to the Power User Attack, was shown to successfully attack the robust item-based algorithm using a multiple-target approach.

10.1 Dissertation Hypotheses

A summary of the dissertation hypotheses (DH) provided in Section 1.5.2 and tested by this research is as follows:

DH-1: The use of In-Degree Centrality to select a set of power users results in power users with higher influence than other selection techniques, across multiple datasets and domains. Overall, this hypothesis is accepted only for user-based recommenders in the movie domain tested. However, it must be rejected for user-based, item-based, and SVD-based recommenders in the music domain tested. This hypothesis is not supported for user-based, item-based, and SVD-based collaborative recommenders using the Y365K dataset because AggSim and NumRatings were able to select a better set of power users than InDegree across all three collaborative algorithms.

This hypothesis serves to answer the following research question, *DRQ-1: Does the use of Social Network Analysis identify more influential Power Users than other*

methods?

DH-2: A significant percentage of synthetic user profiles generated from statistical characteristics of power users will be identified by selected power user selection techniques across multiple datasets and domains. This hypothesis is accepted for InDegree and NumRatings in the movie and music domains tested. This hypothesis serves to answer the following research question, *DRQ-2: Will synthetic Power User profiles generated from power user characteristics retain the same level of influence of real Power Users?*

DH-3: Power user attack profiles generated from characteristics of InDegree-selected power users will result in more effective attacks (from the attacker's viewpoint) than attack profiles generated from characteristics of power users selected from other techniques across CF algorithms, datasets, and domains. This hypothesis is partially accepted. The results are mixed in the movie and music domains tested, i.e., we consistently found InDegree to generate power user attack profiles that were more effective then those generated with AggSim but not always more effective than those generated with NumRatings. If one considers the "cost" of attack, NumRatings is the better choice because it has a very low cost to mount. This hypothesis serves to answer the following research questions, *DRQ-1: Does the use of Social Network Analysis identify more influential Power Users than other methods?* and *DRQ-3: What happens to Recommender System accuracy and robustness when power users attack?*

DH-4: A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects on RS predictions and top-N lists of rec-

ommendations across multiple power user selection techniques, collaborative filtering algorithms, datasets, and domains. The results overwhelmingly support accepting this hypothesis across user-based, item, based, and SVD-based collaborative recommenders using the movie and music domain datasets tested. This hypothesis serves to answer the following research questions, *DRQ-3: What happens to Recommender System accuracy and robustness when power users attack?* and *DRQ-4: Can a novel attack be crafted to achieve power user capability with strong influence and "low" cost of attack?*

DH-5: Reducing the influence of power users is a more effective and less impactful mitigation strategy than completely removing power users from the dataset. This hypothesis is accepted for the "Power User Influence Reduction" strategies for user-based recommenders for the movie domain tested. This hypothesis serves to answer the following research question, *DRQ-5: What is the most effective method of mitigating power user attacks?*

10.2 Contributions

The main contributions made by this research are:

- Power User Attack Model: This is a novel attack model based on influential power users. The model specifies how power users are selected from a dataset and how the power user profiles are configured for the attack. Different techniques for power user selection were evaluated and alternative methods of evaluating power user selection were analyzed in the context of power user attacks. See Chapters 5 and 6.
- Power User Model: This model specifies the statistical characteristics of power

users in sufficient detail so that synthetic power user attack profiles can be generated for attack purposes. This effort mainly involved characterizing power users according to their statistical properties and generating synthetic power user profiles. The degree to which synthetic power user profiles resemble actual power user profiles was evaluated across multiple power user selection techniques. See Chapter 6.

- Evaluation Approach for Power User Selection and Power User Attacks: The approach consists of metrics collected before and after the power user attack and is used to evaluate both the power user selection and the power user attack. A power user evaluation process that combines impacts to accuracy metrics before an attack and impacts to accuracy and robustness metrics after an attack was analyzed. The use of In-Degree Centrality to select a set of power users compared to other power user selection techniques was evaluated across multiple collaborative filtering algorithms, datasets, and domains. The degree to which power user attacks using synthetic profiles can impact RS recommendations across multiple power user selection techniques and collaborative filtering algorithms was also evaluated. See Chapters 5 and 6.
- Power Item Attack Model and Power Item Model: The novel power item attack model uses synthetic power user profiles populated with power items in a novel attack configuration using multiple targets. The power item model describes how synthetic power users are generated using characteristics of influential (power) items. The power item attack was evaluated across multiple power user selection techniques and collaborative filtering algorithms. See Chapter 7.

- Mitigation Approach for Power User Attacks: The approach is to reduce the impact of the power user attack without having to remove 100% of the power users because of the important role that power users play in maintaining a higher level of recommender system accuracy. The approach reducing the influence of power users is a more effective and less impactful mitigation strategy than completely eliminating the influence of power users was evaluated across multiple power user selection techniques and collaborative filtering algorithms. See Chapter 8.
- New Evaluation Metrics: Throughout this dissertation, several metrics were developed when evaluating accuracy and robustness measures. The AC metric discussed in Chapter 4 was used to show the trade-offs between accuracy and coverage when evaluating collaborative filtering algorithms, the NTPU and NNTPU metrics discussed in Chapter 7 were used to determine the effectiveness of the Power Item Attack within and between experiments, and the ARM metric discussed in Chapters 8 and 9 was used to evaluate the trade-offs between accuracy and robustness when evaluating power user attack mitigation strategies.

10.3 Findings

The major findings from this dissertation were:

- A relatively small number of power users (5% or less of the user base on selected datasets) can have significant effects (from the attacker's viewpoint) on recommender system predictions and top-N lists of recommendations across multiple

power user selection methods, collaborative filtering algorithms, and the movie and music domains tested.

- Power user attack profiles generated from characteristics of In-Degree and Number of Ratings power users result in more effective attacks (from the attacker's viewpoint) than attack profiles generated from characteristics of Aggregated Similarity power users across collaborative filtering algorithms and the movie and music domains tested.
- The use of In-Degree Centrality to select a set of power users results in power users with higher influence than other selection techniques for user-based and SVD-based recommenders in the movie domain tested.
- A significant percentage of synthetic user profiles generated from statistical characteristics of power users were identified by the In-Degree and Number of Ratings power user selection methods in the movie and music domains tested.
- Item-based collaborative recommenders, previously considered robust to attack, are vulnerable to the novel Power Item Attack using the novel Multiple-Target design approach.
- Reducing the influence of power users is a more effective and less impactful mitigation strategy than completely removing power users from the dataset for user-based recommenders in the movie and music domains tested.

The implications of these findings are that malicious users can use synthetic power users to mount efficient and effective attacks on popular collaborative recommenders. User-based, item-based, and SVD-based collaborative recommenders have been shown to be vulnerable to attack. These attacks may not be detectable using methods based

solely on statistical characteristics of "average" users; furthermore, the small attack size required for effective power user/item attacks may also elude current detection approaches. To defend against such attacks, system operators should consider using "influence reduction" mitigation strategies rather than removal.

10.4 Limitations and Future Work

There are several areas that need to be studied beyond the scope of this dissertation in order to further generalize the current findings. These limitations should be considered as future work and consist of the following:

- An in-depth power user attack detection study. In the literature, attack detection is limited to known models such as Random, Average, etc., and should be extended to include new attack models such as the Power User Attack and Power Item Attack. It would be interesting to know if existing attack detection methods could be used to detect new attack models.
- Analysis of power user selection methods based on other Social Network Analysis techniques as well as methods based other heuristic and statistical approaches. For example, methods such as eigenvector centrality, page rank, most entropy, most average user, and others should be investigated to determine if they are able to generate synthetic attack user profiles that can significantly impact recommender system accuracy and robustness.
- A replication of the Power User Attack and Power Item Attack experiments using larger, production-sized datasets with millions of users and items across several collaborative filtering algorithms and domains. For example, the use of

Netflix Prize[76] and Yahoo! Music[77] datasets would be a good starting point.

Furthermore, attacks on content-based recommenders also need to be studied in order to mitigate the numerous "opinion spam" attacks being perpetrated on today's online systems.

10.5 Conclusions

The principal conclusions were:

- Power user attacks can have significant impact on the predictions generated by popular collaborative recommender algorithms across the movie and music domains tested, i.e., these attacks can efficiently and effectively bias the recommender predictions as measured by accuracy and robustness metrics.
- Synthetic power user profiles generated from the In-Degree and Number of Ratings power user selection methods result in effective power user attacks.
- 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. Due to its low "cost" of attack, the simple Number of Ratings method appears to be the most efficient approach for selecting and generating power user profiles. Furthermore, 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. And 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-

[76] http://www.netflixprize.com/
[77] http://webscope.sandbox.yahoo.com/catalog.php?datatype=r

available information; this gives NumRatings an edge over the other selection methods, costs being equal.

- When implementing mitigation strategies, reducing similarity weighting during prediction calculation is an improvement over removal of detected power users. Furthermore, an influence reduction strategy also helps to optimize the trade-off between recommender system accuracy and robustness.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6), 2005.

[2] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol. Data mining methods for recommender systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*. Springer, 2011.

[3] S. S. Anand and N. Griffiths. A market-based approach to address the new item problem. In *Proceedings of the 5th ACM Recommender Systems Conference (RecSys '11)*, 2011.

[4] P. Bonacich. Power and centrality: A family of measures. *The American Journal of Sociology*, 92(5), 1987.

[5] R. Burke. Evaluating the dynamic properties of recommendation algorithms. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys 2010)*, 2010.

[6] 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 (ICDM 2005)*, 2005.

[7] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik. Classification features for attack detection in collaborative recommender systems. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006.

[8] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik. Detecting profile injection attacks in collaborative recommender systems. In *The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services*, 2006.

[9] R. Burke, M. P. O'Mahony, and N. J. Hurley. Robust collaborative recommendation. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*. Springer, 2011.

[10] P. A. Chirita, W. Nejdl, and C. Zamfir. Preventing shilling attacks in online recommender systems. In *WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management*. ACM, 2005.

[11] P. de Pechpeyrou. How consumers value online personalization: a longitudinal experiment. *Direct Marketing*, 3(1):35–51, 2009.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1), 1977.

[13] 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.

[14] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01. ACM, 2001.

[15] M. D. Ekstrand, M. Ludwig, J. A. Konstan, and J. T. Riedl. Rethinking the recommender research ecosystem: Reproducibility, openness, and lenskit. In *Proceedings of the 5th ACM Recommender Systems Conference (RecSys '11)*, October 2011.

[16] M. Ge, C. Delgado-Battenfeld, and D. Jannach. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the 4th ACM Recommender Systems Conference (RecSys '10)*, September 2010.

[17] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99)*, July 1999.

[18] A. Goyal and L. V. S. Lakshmanan. Recmax: Exploiting recommender systems for fun and profit. In *Proceedings of the 18th ACM Knowledge Discovery and Data Mining Conference (KDD '12)*, 2012.

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

[20] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 2004.

[21] N. Hurley, Z. Cheng, and M. Zhang. Statistical attack detection. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 149–156, New York, NY, USA, 2009. ACM.

[22] N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 219–230, New York, NY, USA, 2008. ACM.

[23] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03. ACM, 2003.

[24] Y. Koren and R. Bell. Advances in collaborative filtering. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*. Springer, 2011.

[25] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, pages 42–49, 2009.

[26] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*. ACM, 2004.

[27] N. Lathia, S. Hailes, and L. Capra. knn cf: A temporal social network. In *Proceedings of the 2nd ACM Recommender Systems Conference (RecSys '08)*, 2008.

[28] C. Latulipe, N. B. Long, and C. E. Seminario. Structuring flipped classes with lightweight teams and gamification. In *Proceedings of ACM SigCSE Technical Symposium on Computer Science Education*, SigCSE. ACM, March 2015.

[29] S. Mcnee, J. Riedl, and J. Konstan. Accurate is not always good: How accuracy metrics have hurt recommender systems. In *Proceedings of the Conference on Human Factors in Computing Systems(CHI 2006)*, April 2006.

[30] B. Mehta and T. Hofmann. A survey of attack-resistant collaborative filtering algorithms. *IEEE Data Engineering Bulletin*, 31(2), 2008.

[31] 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.

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

[33] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial Intelligence*, pages 187–192, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[34] B. Mobasher, R. Burke, R. Bhaumik, and J. Sandvig. Attacks and remedies in collaborative recommendation. *Intelligent Systems, IEEE*, 22(3):56 –63, May-June 2007.

[35] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams. Effective attack models for shilling item-based collaborative filtering systems. In *Proceedings of the 2005 WebKDD Workshop (KDD'2005)*, 2005.

[36] 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.*, 7(4):23, 2007.

[37] B. R. S. J. Mobasher, B. 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.

[38] M. O'Mahony, N. Hurley, N. Kushmerick, and G. Silvestre. Collaborative recommendation: A robustness analysis. *ACM Trans. Internet Technol.*, 4(4), 2004.

[39] M. P. O'Mahony, N. Hurley, and G. C. M. Silvestre. Promoting recommendations: An attack on collaborative filtering. In *DEXA '02: Proceedings of the 13th International Conference on Database and Expert Systems Applications*, London, UK, 2002. Springer-Verlag.

[40] M. P. O'Mahony, N. Hurley, and G. C. M. Silvestre. Recommender systems: Attack types and strategies. In *Proceedings of the 20st National Conference on Artificial Intelligence (AAAI-05)*, 2005.

[41] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre. An evaluation of neighbourhood formation on the performance of collaborative filtering. *Artif. Intell. Rev.*, 21(3-4):215–228, 2004.

[42] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, June 2011.

[43] 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.

[44] A. Rashid and J. Karypis, G.and Riedl. Influence in ratings-based recommender systems: An algorithm-independent approach. In *Proceedings of the SIAM International conference on Data Mining, 2005*, 2005.

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

[46] P. Resnick and R. Sami. The influence limiter: Provably manipulation-resistant recommender systems. In *Proceedings of the 1st ACM Conference on Recommender Systems (RecSys 2007)*, 2007.

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

[48] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender system – a case study. In *ACM WEBKDD WORKSHOP*, 2000.

[49] B. M. Sarwar, J. A. Konstan, A. Borchers, J. Herlocker, B. Miller, and J. Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work (CSCW '98)*, November 1998.

[50] M. Saveski and A. Mantrach. Item cold-start recommendations: Learning local collective embeddings. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14, pages 89–96, New York, NY, USA, 2014. ACM.

[51] C. E. Seminario. Accuracy and robustness impacts of power user attacks on collaborative recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*, RecSys '13. ACM, October 2013.

[52] C. E. Seminario and D. C. Wilson. Case study evaluation of mahout as a recommender platform. In *CEUR Proceedings series for the Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2012), held in conjunction with the 6th ACM Conference on Recommender Systems (RecSys '12)*, September 2012.

[53] C. E. Seminario and D. C. Wilson. Robustness and accuracy tradeoffs for recommender systems under attack. In *Proceedings of the 25th Florida Artificial Intelligence Research Society Conference*, FLAIRS-25. AAAI, May 2012.

[54] 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 Conference*, FLAIRS-27. AAAI, May 2014.

[55] C. E. Seminario and D. C. Wilson. Attacking item-based recommender systems with power items. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14. ACM, October 2014.

[56] G. Shani and A. Gunawardana. Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*. Springer, 2011.

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

[58] C. Williams, B. Mobasher, R. Burke, R. Bhaumik, and J. Sandvig. Detection of obfuscated attacks in collaborative recommender systems. In *ECAI'06: Proceedings of the 17th European Conference on Artificial Intelligence Workshop on Recommender Systems*, August 2006.

[59] C. Williams, B. Mobasher, and R. D. Burke. Defending recommender systems: detection of profile injection attacks. *Service Oriented Computing and Applications*, 1(3):157–170, 2007.

[60] 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, October 2013.

[61] 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*, UMAP '14, July 2014.

[62] D. C. Wilson and C. E. Seminario. Mitigating power user attacks on a user-based collaborative recommender system. In *(To Appear) Proceedings of the 28th Florida Artificial Intelligence Research Society Conference*, FLAIRS-28. AAAI, May 2015.

[63] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World Wide Web*, WWW '11. ACM, 2011.

APPENDIX A: PUBLICATIONS RELATED TO THIS DISSERTATION

The research for this dissertation has focused on various areas: recommender system evaluation, investigation of power user selection, power user modeling, power user attacks, and mitigation of power user attacks. The following are the papers related to this dissertation, presented in the order they were published:

[53] C. E. Seminario and D. C. Wilson. Robustness and accuracy tradeoffs for recommender systems under attack. In *Proceedings of the 25th Florida Artificial Intelligence Research Society Conference*, FLAIRS-25. AAAI, May 2012.
In [53], we show how the underlying implementation choices for item-based and user-based collaborative filtering recommender systems can affect the accuracy and robustness of recommender systems. We also show how accuracy and robustness can change over a system's lifetime by analyzing a set of temporal snapshots from system usage over time. Results provide insight into some of the trade-offs between robustness and accuracy that operators may need to consider in development and evaluation.

[52] C. E. Seminario and D. C. Wilson. Case study evaluation of mahout as a recommender platform. In *CEUR Proceedings series for the Workshop on Recommendation Utility Evaluation: Beyond RMSE (RUE 2012), held in conjunction with the 6th ACM Conference on Recommender Systems (RecSys '12)*, September 2012.
In [52], we describe the evaluation of changes made to the Mahout-based[78] development/test platform using accuracy and coverage metrics, and a novel metric that combines accuracy and coverage in order to address the trade-off between those two metrics. We demonstrated that the "best" mean absolute error (MAE) may not always be the lowest MAE, especially when coverage is also considered and that a combined metric can be useful addressing the accuracy vs. coverage trade-off. Details about this research are provided in Chapter 4.

[60] 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, October 2013.

[54] 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 Conference*, FLAIRS-27. AAAI, May 2014.
In [60] and [54], we describe our initial work in the evaluation of power user selection techniques and the power user attack with a simulated power user attack on "new" items. We used an ablation approach using accuracy and coverage to evaluate power user selection before the attack; after the attack, we used accuracy and

[78] apache.mahout.org

robustness metrics for evaluation. Results show that power users selected using the SNA in-degree centrality technique have significant impacts on recommender system accuracy and robustness, especially on user-based and SVD-based collaborative recommenders. *This paper was nominated for Best Student Paper Award.* Details about this research are provided in Chapter 5.

[51] C. E. Seminario. Accuracy and robustness impacts of power user attacks on collaborative recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*, RecSys '13. ACM, October 2013.
In [51], I summarize my research proposal on power user attacks and provide additional research results on the evaluation of power user selection techniques and power user characteristics across multiple collaborative filtering algorithms. Details about this research are also provided in Chapter 5.

[61] 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*, UMAP '14, July 2014.
In [61], we describe the power user model and how it is used to generate synthetic power users that can be used for attack purposes. Methods for generating synthetic power users are evaluated before and after the power user attack. Results indicate that synthetic power users can be used to mount effective attacks against user-based and SVD-based recommender systems. Details about this research are provided in Chapter 6.

[55] C. E. Seminario and D. C. Wilson. Attacking item-based recommender systems with power items. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys '14. ACM, October 2014.
In [55], we describe a power item model that uses influential (power) items to attack collaborative recommenders. A novel power item attack model is also introduced. Methods for generating synthetic power users using the power item model are evaluated after the power item attack. Results show that power user attack profiles generated with power items can be effective against user-based, item-based, and SVD-based recommenders. Details about this research are provided in Chapter 7.

[62] D. C. Wilson and C. E. Seminario. Mitigating power user attacks on a user-based collaborative recommender system. In *(To appear) Proceedings of the 28th Florida Artificial Intelligence Research Society Conference*, FLAIRS-28. AAAI, May 2015.
In [62], we posit various mitigation strategies against power user attacks based on either removal or influence-reduction of power users. These mitigation strategies are evaluated against power user attacks on user-based recommender systems. Results

indicate that influence-reduction is a better strategy than power user removal. Details about this research are provided in Chapter 8.

The following Computer Science Education paper, although not related to attacks on recommender systems, was published during the time this dissertation was being developed.

[28] Celine Latulipe, N. Bruce Long, and Carlos E. Seminario. Structuring Flipped Classes with Lightweight Teams and Gamification *In Proceedings of ACM SigCSE Technical Symposium on Computer Science Education*, SigCSE 2015, ACM, March 2015.
In [28], we present a new approach to help make computer science classes both more social and more effective: "lightweight teams". Lightweight teams are class teams in which the team members have little or no direct impact on each other's final grades, yet where there is a significant component of peer teaching, peer learning and long-term socialization built into the curriculum. We explain how lightweight teams have been used in a CS1 class at our institution, and how this approach, combined with a flipped class approach and gamification, has led to high levels of student engagement, despite the difficulty of the material and the frustration that is common to those first learning to program. *This paper was selected as the winner of the SigCSE 2015 Best Paper Award, presented in March 2015.*

APPENDIX B: STATISTICS FOR RECOMMENDER SYSTEM DATASETS USED IN THIS DISSERTATION

These are the statistics for the datasets used in this study. The MovieLens[79] datasets are ML100K, ML1M, and ML10M. The Yahoo! Music[80] datasets are Y9M, Y1M, and Y365K.

Table 6: Statistics for MovieLens and Yahoo! Music datasets

	ML100K	**ML1M12**	**ML10M**	**Y9M**	**Y1M**	**Y365K**
# Users	943	6,034	69,878	23,179	2,717	15,400
# Items	1,664	3,678	10,676	136,736	126,038	1,000
# Ratings	99,693	904,757	10,000,034	8,846,899	1,002,415	365,704
Avg Global Rating	3.530	3.590	3.512	3.159	3.225	2.734
StDev Global Rating	1.126	1.120	1.060	1.596	1.589	1.565
Avg # Ratings/User	105.719	149.943	143.107	381.677	368.942	23.747
StDev # Ratings/User	100.567	174.143	216.709	825.134	714.678	20.224
Max # Ratings/User	736	2,029	7,358	33,920	8,541	648
Min # Ratings/User	19	2	20	20	20	10
Avg User Rating	3.588	3.708	3.614	3.455	3.476	2.822
StDev User Rating	0.445	0.431	0.428	0.886	0.886	0.965
Avg User Entropy	1.870	1.845	1.774	1.618	1.602	1.554
StDev User Entropy	0.260	0.269	0.272	0.556	0.554	0.536
Avg # Ratings/Item	59.912	245.992	936.684	64.701	7.953	365.704
StDev # Ratings/Item	80.655	356.893	2487.314	149.346	17.860	543.756
Max # Ratings/Item	583	3,291	34,864	4,106	467	5,587
Min # Ratings/Item	1	1	1	2	1	66
Avg Item Rating	3.077	3.236	3.192	2.970	3.048	2.492
StDev Item Rating	0.780	0.687	0.567	0.701	1.120	0.478
Avg Item Entropy	1.626	1.796	1.762	1.935	1.104	2.045
StDev Item Entropy	0.619	0.460	0.347	0.363	0.749	0.252
Sparsity	0.936	0.959	0.987	0.997	0.997	0.976

[79] www.grouplens.org

[80] http://webscope.sandbox.yahoo.com/catalog.php?datatype=r