Project Report

on

Advancing recommender system by mitigating shilling attacks

submitted in partial fulfillment of the requirement
for the award of the Degree of

Bachelor of Engineering
in
Computer Engineering

by

Aditya Chichani
Juzer Golwala
Tejas Gundecha

under the guidance of

Prof.Kiran Gawande

Department of Computer Engineering
Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri-West, Mumbai-400058
University of Mumbai

April 2018

Certificate

This is to certify that the Project entitled "Advancing recommender system by mitigating shilling attacks" has been completed to our satisfaction by Mr.Aditya Chichani, Mr.Juzer Golwala and Mr.Tejas Gundecha under the guidance of Prof.Kiran Gawande for the award of Degree of Bachelor of Engineering in Computer Engineering from University of Mumbai.

Certified by

Prof.Kiran Gawande
Project Guide

Head of Department

Dr. Prachi Gharpure
Principal

Department of Computer Engineering
Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri(W), Mumbai-400058
University of Mumbai
April 2018

Project Approval Certificate

This is to certify that the Project entitled “Advancing recommender system by mitigating shilling attacks ” by Mr. Aditya Chichani, Mr. Juzer Golwala and Mr. Tejas Gundecha is found to be satisfactory and is approved for the award of Degree of Bachelor of Engineering in Computer Engineering from University of Mumbai.

External Examiner	**Internal Examiner**
(signature)	**(signature)**
Name:	**Name:**
Date:	**Date:**

Seal of the Institute

Statement by the Candidates

We wish to state that the work embodied in this thesis titled “Advancing recommender system by mitigating shilling attacks” forms our own contribution to the work carried out under the guidance of Prof. Kiran Gawande at the Sardar Patel Institute of Technology. We declare that this written submission represents our ideas in our own words and where others’ ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission.

Name and Signature:
1.Aditya Chichani
2.Juzer Golwala
3.Tejas Gundecha

Acknowledgments

It is really a pleasure to acknowledge the help and support that has gone to in making this thesis. We express our sincere gratitude to our guide Prof. Kiran Gawande for her invaluable guidance. We also thank her for encouraging us to work with Recommender Systems. Without her encouragement this work would not be a reality. With the freedom she provided, we really enjoyed working under her.

We thank our examiner, Prof. Reeta Koshy for her word of advice.

We thank HOD and staff of Computer Engineering Department for giving us all the facilities required to carry out this research work.

We would like to thank all our family members and well wishers for their constant encouragement for all these years, without which we could not have completed this work.

Contents

List of Figures

List of Tables

List of Abbreviations

CF	Collaborative Filtering
SVM	Support Vector Machine
SVD	Singular Value Decomposition
PCA	Principal Component Analysis
RMSE	Root Mean Squared Error

Abstract

The main aim of this project is to create a model which can be used in recommendation systems such that it gives unbiased and efficient recommendations. This model will be general and it can be incorporated with any recommender system. The approach would be research based wherein current trends and attacks in recommender systems would be studied and a research gap would be identified. The model devised would try to improvise upon the current solutions which are still lacking a foolproof solution to the recommender system challenges.

Chapter 1

Introduction

Recommender System: A recommender system or a recommendation system is a subclass of information filtering system that seeks to predict the ”rating” or ”preference” that a user would give to an item.

Collaborative Filtering(CF): Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating).

Item-Item CF: Item-item collaborative filtering, or item-based, or item-to-item, is a form of collaborative filtering for recommender systems based on the similarity between items calculated using people’s ratings of those items.

User-User CF: User-User collaborative filtering, or user-based, or user-to-user, is a form of collaborative filtering for recommender systems based on the similarity between users calculated using people’s ratings on items common amongst them.

Content-Based Filtering: Content-based filtering methods are based on a description of the item and a profile of the users preferences. In a content-based recommender system, keywords are used to describe the items and a user profile is built to indicate the type of item this user likes.

SVM: A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

PCA: PCA stands for Primary Component Analysis wherein on the basis of correlation profiles are marked as shilling profile.

Recommender systems have become a pivotal part in many business strategies.They provide a scalable way of personalising content for users in scenarios with many items and lead to increased revenues. Different techniques used for generating recommendations are Collaborative Filtering(CF) and Content-Based Filtering.However these models are prone to Shilling attacks which can lead to biased recommendations and result in heavy commercial losses. This project works on preventing and mitigating these attacks. Considering the premise that the content grows in an almost irrational way and the amount of users is a much more constant figure, recommender systems help in categorising content according to users preferences.It seeks to predict the "rating" or "preference" that a user would give to an item.The recommendations can be based on a domain- specific notion of item content (Content Based Filtering) consumed by the user or based on the preferences of users with similar taste. (Collaborative Filtering).Attackers try to take undue advantage of the underlying mechanism used to calculate these recommendations and push/nuke target items.These biasing attacks are known as Shilling attacks.

Attack Models: An attack consists of attack profiles that are introduced into the system in order to alter recommendation lists of a set of target items. Based on different assumptions about the attackers knowledge and purpose, a number of attack models have been identified. There are four popular attack models: random attack, average attack, bandwagon attack and segment attack models.

Random attack model

- Take random values for filler items
- Typical distribution of ratings is known, e.g., for the movie domain(Average 3.6, standard deviation around 1.1)
- Idea: Generate profiles with "typical" ratings so they are considered as neighbors to many other real profiles
- Give High/low ratings for target items
- Limited effect compared with more advanced models

Average attack model

- Use the individual item's rating average for the filler items
- Intuitively, there should be more neighbors
- Additional cost involved: find out the average rating of an item
- More effective than Random Attack in user-based CF
- Additional knowledge is required

Bandwagon attack model

- Add profiles that contain high ratings for "blockbusters" (in the selected items); use random values for the filler items
- Will intuitively lead to more neighbors because popular items will have many ratings and rating values are similar to many other user-profiles
- Example: Injecting a profile with high rating values for the Harry Potter series
- Low-cost attack since set of top-selling items/blockbusters can be easily determined

1.1 Motivation

Shilling attacks can lead to biased recommendations and result in heavy commercial losses. It is computationally feasible to inject shilling profiles and the structure of the user profiles in the recommender system is such that these shilling profiles go undetected. These profiles cause a lot of financial damage if no detection systems are in place. This project works on preventing and mitigating these attacks.

1.2 Objectives

- To design multiple recommendation models, perform various shilling attacks on these models and analyze their consequences.
- To evaluate and improve the accuracy of CF model using different statistical methods.
- To create a model that detects and prevents shilling attacks.
- To analyze the variation in f-measure by varying profile threshold and correlation threshold value.

1.3 Problem Statement

- The recommender system should incorporate collaborative filtering which is immune to shilling attacks.
- Any user profile trying to implement shilling attack should be detected correctly. Recommendations from genuine users should not be neglected considering them as attacking profile.
- User should get satisfactory recommendations.
- The recommendations for a user should be computed in a time-efficient manner.

1.4 Contributions

- This project has improved the accuracy of existing statistical models used in CF.
- Implemented shillings profile detection using the concept that a profile that is highly correlated to many other profiles is probably a shilling profile.
- Studied the variation of accuracy by varying the threshold values for correlation and the number of users with which a profile should be correlated to be detected as a shilling profile.
- Studied the difference between supervised and unsupervised shilling attack detection techniques and reached the conclusion that unsupervised shilling attack detection techniques are better.

1.5 Layout of the Report

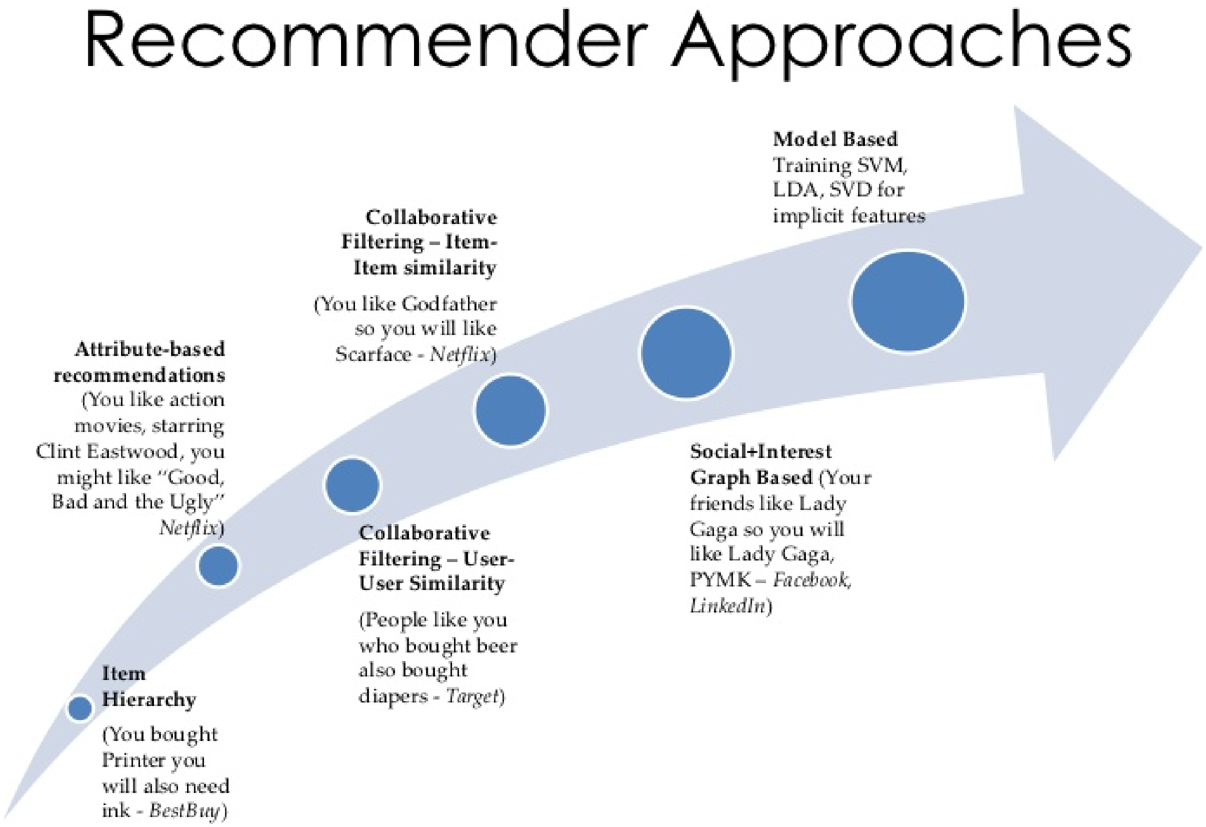
A brief chapter by chapter overview is presented here.
Chapter 2: A literature review of different types of Recommender systems and the different types of attack models.
Chapter 3: Design for attack models and detection algorithm is finalized.
Chapter 4: Different attack models are performed on the rating matrix and detection algorithm is applied. Variation in f-measure is observed by varying profile threshold and correlation threshold value.
Chapter 5: Variation of f-measure by changing parameters like profile threshold and correlation threshold value and effect of shilling profile injection on RMSE value of Target items are displayed as graphs.
Chapter 6: Future scope of current implementation is mentioned and also how the algorithm can be improved for future use.
Chapter 7: Conclusions and discussion on future course of research work.

Chapter 2

Literature Survey

Xiaoyuan Su and Taghi M. Khoshgoftaar, 'A Survey of Collaborative Filtering Techniques' Hindawi Publishing Corporation Advances in Artificial Intelligence,Volume 2009, Article ID 421425, 19 pages, August 2009.

A recommender system is an information system that is used to predict the best items or products to the users according to their past behavior and possibly using other kinds of data. By using that rating recommender system produces a rating model, then by using that rating model recommendation has been made to the users for further items according to the user preferences. To make predictions about a users interests recommender system has to learn a user models. Then, it selects items based on the similarities among the preferences of different users. The novelty of this work lies in designing a collaborative filtering recommender system model for the purpose of recommendation by using super similar and super dissimilar terminology and for average similar term using our proposed similarity metric. We take all popular similarity metrics for computing the support matrix. For a user, the support matrix provides us a decision of super similarity, average similarity or super dissimilarity among all other users. Here, we have considered the users preference that changes over time, as well as we have taken the confidence among users into account. The results that we have shown in performance evaluation section, exhibit that our offered recommender system along with this similarity model achieved better performances compared to state-of-the-art similarity metrics in terms of average mean absolute error, coverage, precision and recall. Various techniques of recommender system have been proposed which include collaborative filtering, knowledge-based, demographic-based, utility-based, content-based techniques. Collaborative filtering relies on the behavior, preferences, and opinions of a large community of other users, and hence named community-based technique. Content-based technique takes into account additional information about items and user preferences and does not depend on rating history. In demographic approach, depending on sex, age, marital status, items which are relevant to the user at that point are recommended. Hybrid approach is a combination of two or more of the above-mentioned approaches. Collaborative filtering [3] mainly consists of three following steps. (i) Building of user-item rating matrix (ii) Selecting the nearest neighbors. (iii) Generating recommendations from neighbors Collaborative filtering works on the principle that if users X and Y rate n items similarly, or have similar behaviours (e.g., buying, watching, listening), and hence will rate or act on other items similarly. In a more general sense, collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc.Content- based recommender systems make recommendations by analysing the content of textual information and finding regularities in the content. The major difference between CF and content-based recommender systems is that CF only uses the user-item ratings data to make predictions and recommendations, while content-based recommender systems rely on the features of users and items for predictions



Recommender Approaches

Item Hierarchy (You bought Printer you will also need ink - *BestBuy*)

Attribute-based recommendations (You like action movies, starring Clint Eastwood, you might like "Good, Bad and the Ugly" *Netflix*)

Collaborative Filtering – User-User Similarity (People like you who bought beer also bought diapers - *Target*)

Collaborative Filtering – Item-Item similarity (You like Godfather so you will like Scarface - *Netflix*)

Social+Interest Graph Based (Your friends like Lady Gaga so you will like Lady Gaga, PYMK – *Facebook, LinkedIn*)

Model Based Training SVM, LDA, SVD for implicit features

Figure 2.1: Recommender System Approaches

2.0.1 Major challenges :

1. Data Sparsity : The user-item matrix used for collaborative filtering becomes extremely sparse in case of large product sets and as a result performances of the recommendations of the CF systems are challenged.

2. Scalability : When the number of existing users and items grow tremendously, traditional CF algorithms suffer serious scalability problems, with computational resources going beyond practical or acceptable levels.

3. Grey Sheep : Grey sheep refers to the users whose opinions do not consistently agree or disagree with any group of people and thus do not benefit from collaborative filtering

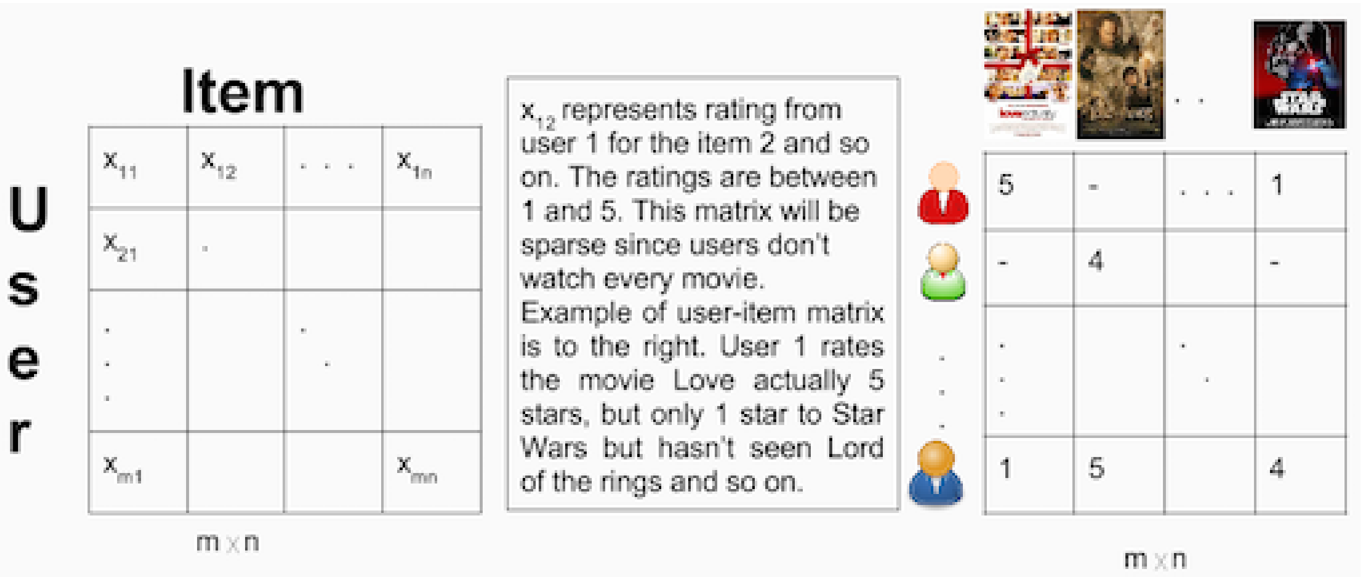
4. Shilling Attack : In cases where anyone can provide recommendations, people may give tons of positive recommendations for their own materials and negative recommendations for their competitors. It is desirable for CF systems to introduce precautions that discourage this kind of phenomenon

5. Synonymy : Synonymy refers to the tendency of a number of the same or very similar items to have different names or entries. Most recommender systems are unable to discover this latent association and thus treat these products differently

2.0.2 Memory-Based Collaborative Filtering Techniques

Memory-based CF algorithms use the entire or a sample of the user-item database to generate a prediction. Every user is part of a group of people with similar interests. By identifying the so-called neighbors of a new user (or active user), a prediction of preferences on new items for him or her can be produced. The neighborhood-based CF algorithm, a prevalent memory-based CF algorithm, uses the following steps: calculate the similarity or weight, wi,j, which reflects distance, correlation, or weight, between two users or two items, i and j; produce a prediction for the active user by taking the weighted average of all the ratings of the user or item on a certain item or user, or using a simple weighted average. When the task is to generate a top-N recommendation, we need to find k most similar users or items (nearest neighbors) after computing the similarities, then aggregate the neighbors to get the top-N most frequent items as the recommendation. Similarity Computation. Similarity computation be- tween items or users is a critical step in memory-based collaborative filtering algorithms. For item-based CF algorithms, the basic idea of the similarity computation between item i and item j is first to work on the users who have rated both of these items and then to apply a similarity computation to determine the similarity, wi,j, between the two co-rated items of the users. For a user-based CF algorithm, we first calculate the similarity, wu,v, between the users u and v who have both rated the same items. There are many different methods to compute similarity or weight between users or items.

Correlation-Based Similarity: In this case, similarity wu,v between two users u and v, or wi,j between two items i and j, is measured by computing the Pearson correlation or other correlation-based similarities. Pearson correlation measures the extent to which two variables linearly relate with each other. Some variations of item-based and user-based Pearson correlations can be found in. The Pearson correlation- based CF algorithm is a representative CF algorithm, and is widely used in the CF research community. Other correlation-based similarities include: constrained Pearson correlation, a variation of Pearson correlation that uses midpoint instead of mean rate; Spearman rank correlation, similar to Pearson correlation, except that the ratings are ranks; and Kendalls correlation, similar to the Spearman rank correlation, but instead of using ranks themselves, only the relative ranks are used to calculate the correlation. Usually the number of users in the computation of similarity is regarded as the neighborhood size of the active user, and similarity based CF is deemed as neighborhood- based CF.



x_{11}	x_{12}	. . .	x_{1n}
x_{21}	.		
⋮		⋱	
x_{m1}			x_{mn}

x_{12} represents rating from user 1 for the item 2 and so on. The ratings are between 1 and 5. This matrix will be sparse since users don't watch every movie. Example of user-item matrix is to the right. User 1 rates the movie Love actually 5 stars, but only 1 star to Star Wars but hasn't seen Lord of the rings and so on.

5	-	. . .	1
-	4		-
⋮		⋱	
1	5		4

Figure 2.2: Memory Based Collaborative Filtering

Vector Cosine-Based Similarity: The similarity between two documents can be measured by treating each document as a vector of word frequencies and computing the cosine of the angle formed by the frequency vectors. This formalism can be adopted in collaborative filtering, which uses users or items instead of documents and ratings instead of word frequencies. Formally, if R is the m n user-item matrix, then the similarity between two items, i and j, is defined as the cosine of the n dimensional vectors corresponding to the ith and jth column of matrix R. In an actual situation, different users may use different rating scales, which the vector cosine similarity cannot take into account. To address this drawback, adjusted cosine similarity is used by subtracting the corresponding user average from each co-rated pair. The Adjusted cosine similarity has the same formula as Pearson correlation. In fact, Pearson correlation performs cosine similarity with some sort of normalization of the users ratings according to his own rating behavior. Hence, we may get negative values with Pearson correlation, but not with cosine similarity, supposing we have an n-point rating scale.

User-Based Top-N Recommendation Algorithms

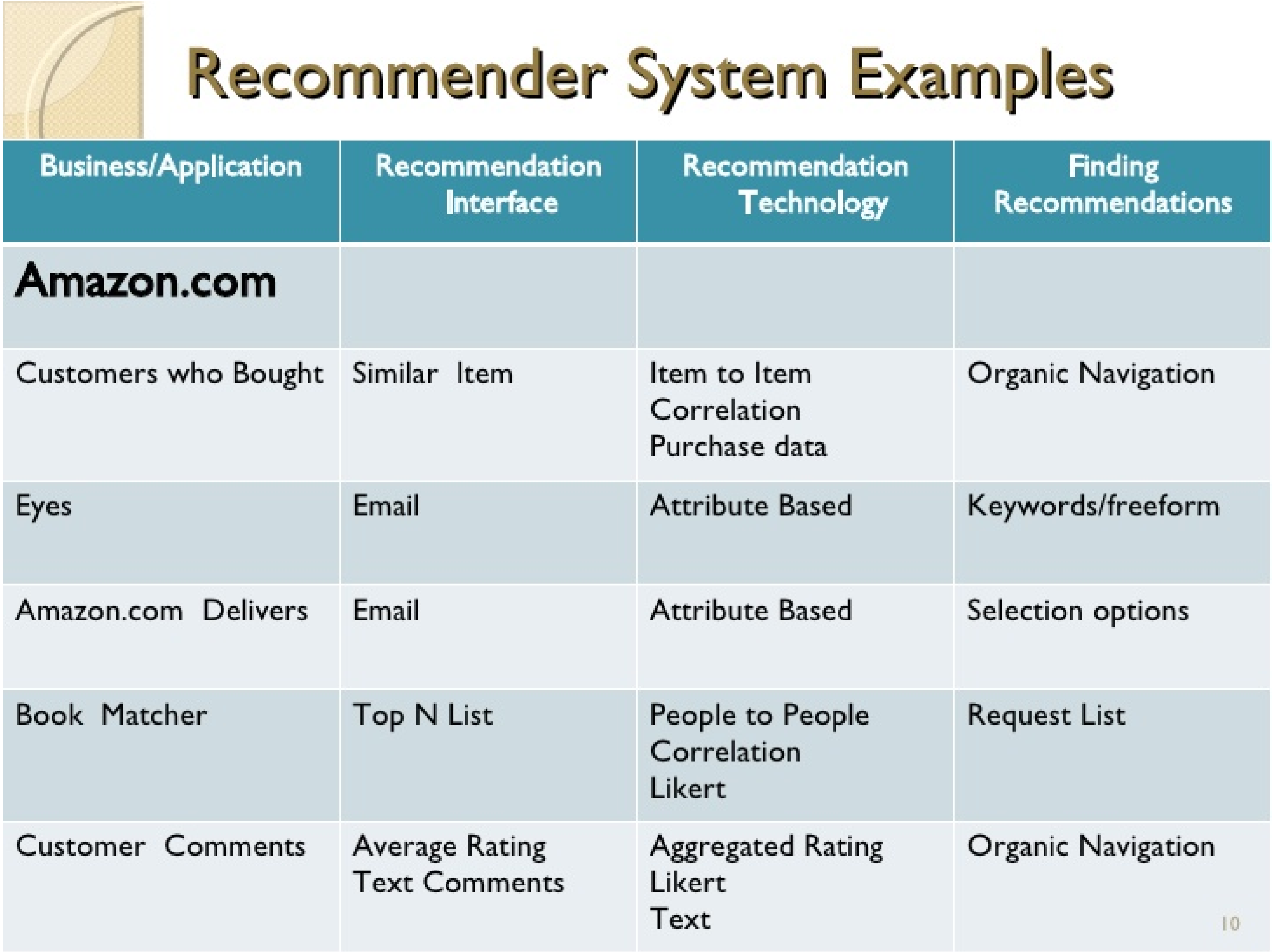
User- based top-N recommendation algorithms firstly identify the k most similar users (nearest neighbors) to the active user using the Pearson correlation or vector-space model, in which each user is treated as a vector in the m-dimensional item space and the similarities between the active user and other users are computed between the vectors. After the k most similar users have been discovered, their corresponding rows in the user-item matrix R are aggregated to identify a set of items, C, purchased by the group together with their frequency. With the set C, user-based CF techniques then recommend the top-N most frequent items in C that the active user has not purchased. User-based top- N recommendation algorithms have limitations related to scalability and real-time performance.

Item-Based Top-N Recommendation Algorithms

Item- based top-N recommendation algorithms have been developed to address the scalability problem of user-based top-N recommendation algorithms. The algorithms firstly compute the k most similar items for each item according to the similarities; then identify the set, C, as candidates of recommended items by taking the union of the k most similar items and removing each of the items in the set, U, that the user has already purchased; then calculate the similarities between each item of the set C and the set U. The resulting set of the items in C, sorted in decreasing order of the similarity, will be the recommended item-based Top-N list. One problem of this method is, when the joint distribution of a set of items is different from the distributions of the individual items in the set, the above schemes can potentially produce suboptimal recommendations. To solve this problem, Deshpande and Karypis developed higher- order item-based top-N recommendation algorithms that use all combinations of items up to a particular size when determining the item sets to be recommended to a user.

2.0.3 Hybrid Collaborative Filtering Techniques

Hybrid CF systems combine CF with other recommendation techniques (typically with content-based systems) to make predictions or recommendations. Content-based recommender systems make recommendations by analyzing the content of textual information, such as documents, URLs, news messages, web logs, item descriptions, and profiles about users tastes, preferences, and needs, and finding regularities in the content. Many elements contribute to the importance of the textual content, such as observed browsing features of the words or pages (e.g., term frequency and inverse document frequency), and similarity between items a user liked in the past. A content-based recommender then uses heuristic methods or classification algorithms to make recommendations. Content-based techniques have the start-up problem, in which they must have enough information to build a reliable classifier. Also, they are limited by the features explicitly associated with the objects they recommend (sometimes these features are hard to extract), while collaborative



Recommender System Examples

Business/Application	Recommendation Interface	Recommendation Technology	Finding Recommendations
Amazon.com			
Customers who Bought	Similar Item	Item to Item Correlation Purchase data	Organic Navigation
Eyes	Email	Attribute Based	Keywords/freeform
Amazon.com Delivers	Email	Attribute Based	Selection options
Book Matcher	Top N List	People to People Correlation Likert	Request List
Customer Comments	Average Rating Text Comments	Aggregated Rating Likert Text	Organic Navigation

10

Table 2.1: Recommender System Example

filtering can make recommendations without any descriptive data. Also, content-based techniques have the overspecialization problem, that is, they can only recommend items that score highly against a users profile or his/her rating history. Other recommender systems include demographic-based recommender systems, which use user profile information such as gender, postcode, occupation, and so forth; utility-based recommender systems and knowledge-based recommender systems, both of which require knowledge about how a particular object satisfies the user needs. We will not discuss these systems in detail in this work.Hoping to avoid limitations of either recommender system and improve recommendation performance, hybrid CF recommenders are combined by adding content-based characteristics to CF models, adding CF characteristics to content-based models, combining CF with content-based or other systems, or combining different CF algorithms.

Hybrid Recommenders Combining CF and Other Recommender Systems. A weighted hybrid recommender combines different recommendation techniques by their weights, which are computed from the results of all of the available recommendation techniques present in the system. The combination can be linear, the weights can be adjustable, and weighted majority voting or weighted average voting can be used.

For example, the P- Tango system initially gives CF and content-based recommenders equal weight, but gradually adjusts the weighting as predictions about user ratings are confirmed or disconfirmed. The strategy of the P-Tango system is similar to boosting. A switching hybrid recommender switches between recommendation techniques using some criteria, such as confidence levels for the recommendation techniques. When the CF system cannot make a recommendation with sufficient confidence, then another recommender system such as a content-based system is attempted. Switching hybrid recommenders also introduce the complexity of parameterization for the switching criteria. Other hybrid recommenders in this category include mixed hybrid recommenders, cascade hybrid recommenders, meta-level recommenders, and so forth. Many papers empirically compared the performance of hybrid recommenders with the pure CF and content-based methods and found that hybrid recommenders may make more accurate recommendations, especially for the new user and new item situations where a regular CF algorithm cannot make satisfactory recommendations. However, hybrid recommenders rely on external information that is usually not available, and they generally have increased complexity of implementation.

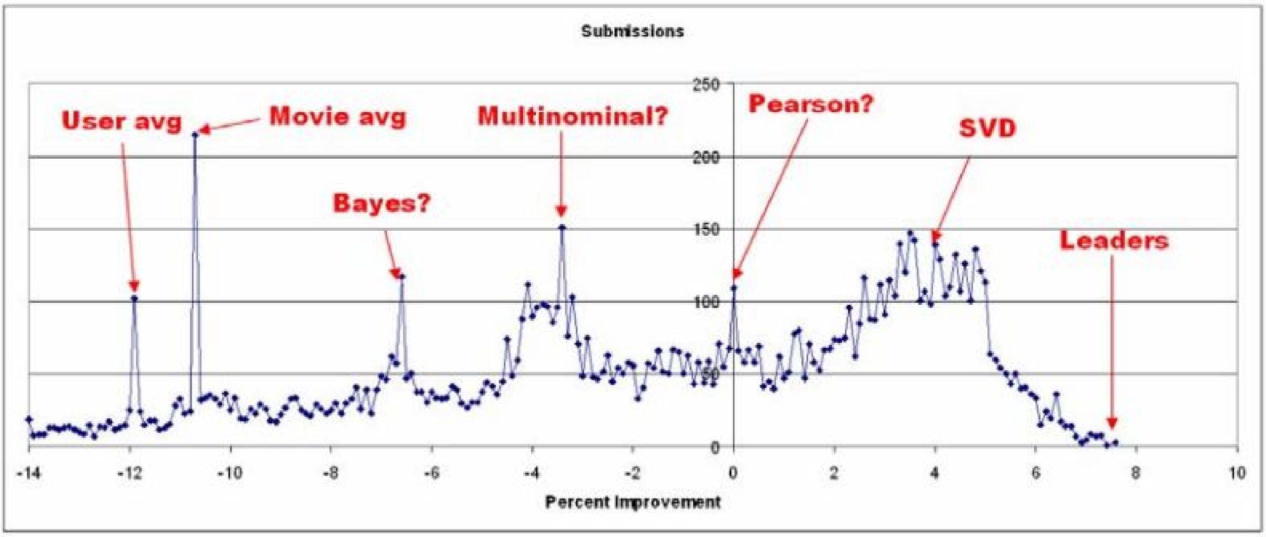


Figure 2: Detail of distribution of leading submissions indicating possible techniques

Figure 2.3: The Netflix Challenge

Zi-Jun Deng , Fei Zhang , Sandra P. S. Wang , 'Shilling Attack Detection In Collaborative Filtering Recommender System By PCA Detection And Perturbation',2016 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), pp.213-218, July 2016

2.0.4 Shilling Attack Models

An attack consists of attack profiles that are introduced into the system in order to alter recommendation lists of a set of target items. Based on different assumptions about the attacker's knowledge and purpose, a number of attack models have been identified. There are four popular attack models: random at- tack, average attack, bandwagon attack and segment attack mod- els. Ratings in an attack profile can be divided into three sets of items: a target item set IT, a selected item set IS and a filler item set IF. Fig. 1 shows the structure of an attack profile. For each attack profile, depending on the attack type, one or more items are chosen and given either the maximum and minimum rating vale. Selected set IS is the set of selected items that have special features, IS is not necessary for some attack models; IF is the set of filler items usually chosen randomly. Filler items in an attack profile are a set of items that make the profile look similar to genuine profiles. The quality of the filler items depends on the existing knowledge gathered from the recommender system. As more knowledge is obtained, an attack generated is more sophisticated. The major difference of attack models is how the ratings of filler items and the selected items are determined. The differences among attack models are the variance rating distribution in filler items and the selected items.

Profile attributes extraction

In order to distinguish attack profiles from genuine profiles, researchers divide profile attributes into general attributes and attack model specific attributes according to the attack models. General attributes try to capture the property from the perspective of descriptive statistics of the profiles, which distinguish the genuine profiles and attack profiles. Specific attributes are those that try to distinguish the genuine profiles and attack pro- files of a particular attack model. Due to the sparsity and rating high-dimensional property of rating matrix in recommender systems, it is unrealistic to apply shilling attack methods directly on the original rating matrix. Therefore, researchers will focus on profile attributes extraction techniques and dimensionality reduction techniques, detection algorithms based on supervised learning methods are then implemented on a set of attributes extracted from these profiles. The training set is created as a combination of user profiles from the MovieLens rating matrix and attack profiles generated using attack models.Each profile is labeled as either being an attack profile or as a genuine user profile. (We assume that all profiles in MovieLens rating matrix are genuine profiles.) A binary classifier is then created based on the training set using the attributes described below. Each profile is labeled as either being part of an attack or as coming from a genuine user.

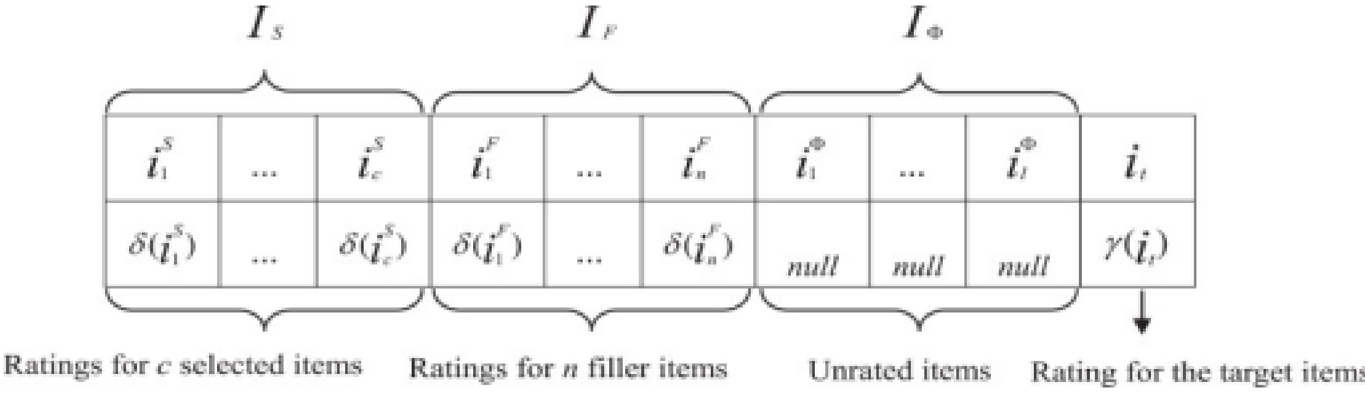


Figure 2.4: Profile Extraction

Wei Zhou , Junhao Wen , Qingyu Xiong , Min Gao, Jun Zeng, 'SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems,'*Elsevier*, 19 October 2016

Shilling attack detection based on SVM and target item analysis

In this section, we study the use of SVM based method and group characteristics in attack profiles. A two phase detecting method SVM-TIA is proposed based on these two methods. In the first phase, Borderline-SMOTE method is used to alleviate the class unbalance problem in classification; a rough detecting result is obtained in this phase; the second phase is a fine-tuning phase whereby the target items in the potential attack profiles set are analyzed. There are three parts in the process. The first part is attribute extraction. A rating matrix is composed by ratings on all items in the recommender system. In the matrix, every row of data stands for all the ratings a user rated on all items; similarly, every column of data stands for all the ratings on the item by all the users in the system. Rating data of a user is called the user profile. Different profile attributes are extracted from every profile.In this part, a rough detecting result is got using the classifier. In the third part, target item analysis method is a fine-tuning phase whereby the target items in the potential attack profiles set are analyzed. Profiles that are misjudged as attack profiles can be filter out.

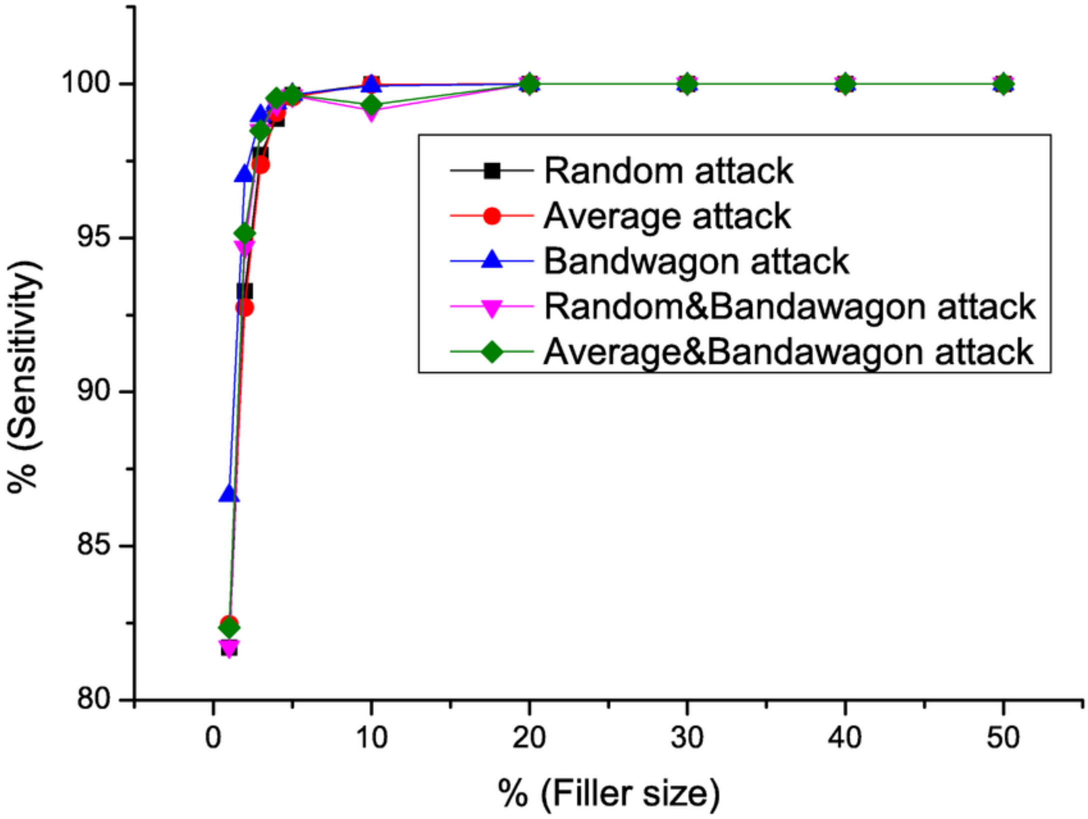


Figure 2.5: Types of Shilling Attacks

Manjeet Kaur and Shalini Batra, 'A Novel Trust Mechanism for Collaborative Recommendation Systems,' ***Computing and Network Sustainability: Proceedings of IRSCNS 2016 published by Springer Singapore*****, pp. 343-351, Oct 2016.**

Novel Trust Mechanism

Collaborative filtering works by building a user item database of preferences of items by users. An active user is matched against the database to produce neighbors. In the proposed approach, clustering algorithm is applied in the initial stage. Clustering the user item rating database divides the entire set into K clusters, to one of which the new customer will belong. Now recommendations are found on much smaller part of entire user database, i.e., cluster leading to increased processing speed, and hence performance is enhanced. Trust being time-dependent, information can be analyzed using the changed trust level of crime investigators on their secret informers. Few police officers are well known for having relied on informers more than their own team. The trust on informers is dynamic in nature, and it varies as follows: Informers that come up with more useful information in terms of the material it possess toward solving the case are updated with higher trust, and informers with less or no clues are less relied. Hence, recommendations are produced by first constructing the clusters to find closest neighbors, and then trust is calculated that is updated each time recommendations are looked for by the active user. In our proposed approach, speed of classic collaborative filtering method is improved using clustering and attack such as shilling attack is eliminated by introducing trust mechanism.

Clustering of Users

Initially, K-mean clustering algorithm has been used to organize the dataset into meaningful groups, assuming that each object belongs to only one cluster and overlapping of cluster does not happen. These clusters should have high intra-cluster similarity and low inter-cluster similarity. Input: Dataset user-item rating matrix and number of clusters k. Output: Set of cluster centers C. 1. Set the number of clusters. Say k. 2. Pick any centroids of k clusters. 3. Compute Euclidean distance of each user in the dataset from each of centroids. 4. Allocate each user to the cluster it is similar to on the basis of Euclidean distance calculated above. 5. Compute centroids for these clusters by calculating average of column value of users in each cluster. 6. If cluster membership is changed go to Step 3, else stop.
Placing Active User Active user will reside in one of the clusters based on its closeness to the centroids of clusters. Euclidean distance of active user from centroids is calculated, and user is placed in the cluster with the least Euclidean distance. Euclidean distance between two users is calculated as: D= Mod(X1X2)+Mod(Y1Y2) It shows that active user will get its place in the cluster that has users having preferences similar to that of the active user. Recommending Top T Items to Active User. This database is sorted, and top T items of the database are recommended to active user. These are actually items with relatively high score. Updating Trust in Trust Database; trust can be contradictory; sometimes trust has different meanings in various subjects. It depends on mind, experience, and circumstances. Trust is not symmetric in nature, and its value is unidirectional as well. Secret informer with useful information: If informer has some useful information and trustworthy past records, he can be trusted in future too. Considering this scenario in the recommender system, if active user not only visits the recommended items but rate them as well, trust on neighbors with such items is updated as: Secret informer with no useful information: If informer does not have any useful matter in the information that could provide lead to solve the case but has trustworthy past, trust on them may change a little bit but may not get reduced drastically. Similarly, a situation may be where the active user may only visit the recommended item but do no rate them. Secret informer with no information: Sometimes informer does not get any information for the present case and further no trustworthy past is associated with this him, thus it can be said that the trust is more influenced. Considering the same scenario in recommender system, an active user may neither have rated the recommended item nor have visited the item. Here, even though no item is actually recommended to active user by this neighbor, count is considered as 1. By above formulae, trust on neighbors who have more potential of recommendation is increased and for those who do not provide good recommendations is decreased.

Zhihai Yang and Zhongmin Cai 'Detecting abnormal profiles in collaborative filtering recommender systems', *J. Intell. Inf. Syst.*, Yang C17, India, pp. 499–518,2017.

Shilling Attack detection

Shilling attacks, a.k.a. profile injection attacks, attempt to influence the system's behavior by injecting a set of fake profiles into the database of normal user profiles. Even the most robust of the recommender systems studied have not been unaffected by shilling attacks, and no collaborative system could be. In recent years, extensive studies have been proposed for detecting and reducing the effects of shilling attacks. These studies mainly focus on the three subareas: the shilling attack generative models, the shilling attack detection features, and the shilling attack detection algorithms. Many kinds of techniques have been applied to the detector design, such as statistical methods, classification models, matrix factorization models, etc. Most classification based detectors represent every user profile in a feature space and then use supervised or semi-supervised learning for training the classification model. Statistical and matrix factorization based detectors are usually unsupervised, and they essentially try to compute a rank score of each user for measuring its suspicious degree to be an attacker. Supervised classification and unsupervised matrix factorization are two mainstream techniques used in shilling attack detectors. They have been adopted as baseline tools as a new detector is developed. However, none of work has conduct a complete yet in-depth analysis on which type of shilling attacks these detectors are/are not be qualified, and more importantly why they can/cannot identify a special type of attackers. Furthermore, we would like to summarize the essential characteristics of attackers that might determine the success or failure of different kinds of detectors.

Zi-Jun Deng , Fei Zhang , Sandra P. S. Wang , 'Shilling Attack Detection In Collaborative Filtering Recommender System By PCA Detection And Perturbation',2016 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), pp.213-218, July 2016

Shilling Profile Generation

The profile of a shilling attacker, i.e., shilling profile, is in essence a rating record on various items, and it is usually composed of ratings on three-typed items: target items, filler items, and non-voted items, as shown in Fig. 1. Rating target items is determined by the attack intent, i.e., rating the highest score in a push attack or the lowest score in a nuke attack. Filler items can make a shilling profile look normal, yet exert profound impacts on other users. In certain attack types, such as bandwagon, a subset of filler items, a.k.a. selected items, may be pre-selected for a precise impact. The shilling attacks can be categorized in several types according to the methods for selecting filler items and assigning ratings to them. There are three ways to select filler items: random selection, selection over popular items, and the combination of random and popular. The random-filler model (RFM) and average-filler model (AFM) represent two different ways to assign ratings to items in IF. Therefore, the shilling attack models are finally summarized as six types.

B. Supervised Learning Given a number of selected features, every user profile in both training and test data is represented as a feature vector. Since the training examples often fall in two classes, i.e., normal and shilling, the shilling attack detection problem is transformed to a binary classification problem. Rather, all kinds of algorithms can then be used for training the classifier. In this paper, we consider two kinds of classification models, including the C4.5 decision tree and Nave Bayesian (NB). C4.5 was ever used for building the decision tree to identify various types of shilling attackers in. NB was used for shilling attack detection in due to its good interpretability. An important assumption within NB is that the continuous values of each feature yield the Gaussian distribution, which lays the foundation of parameter estimation. Since C4.5 and NB model are preliminaries in data mining, we here omit the details on the model training.

Unsupervised Shilling Detection Algorithms

Generally, unsupervised detectors run on user-item rating matrix directly, rather than the feature space. In this section, we briefly introduce two well-known unsupervised detectors: PCA-based algorithm and MDS-based algorithm. These two detectors are in common with using the matrix factorization technique, but on user-user covariance (i.e., similarity) matrix and dissimilarity matrix respectively. A. PCA-based Algorithm The PCA-based algorithm is also PCA Select Users, PCA for short in this paper. It is the first and the most famous unsupervised detector in shilling attack detection. Based on user-item rating matrix, PCA computes the user-user covariance matrix. Mathematically, if let ru and rv be rating vectors of user u and v, the covariance denoted by Cov u,v.

Youquan Wang, Lu Zhang , Haicheng Tao , Zhiang Wu, Jie Cao,'A Comparative Study of Shilling Attack Detectors for Recommender Systems,'IEEE 2015 12th International Conference on Service Systems and Service Management (ICSSSM),pp. 1-6, June 2015.

Supervised Shilling Detection Algorithms

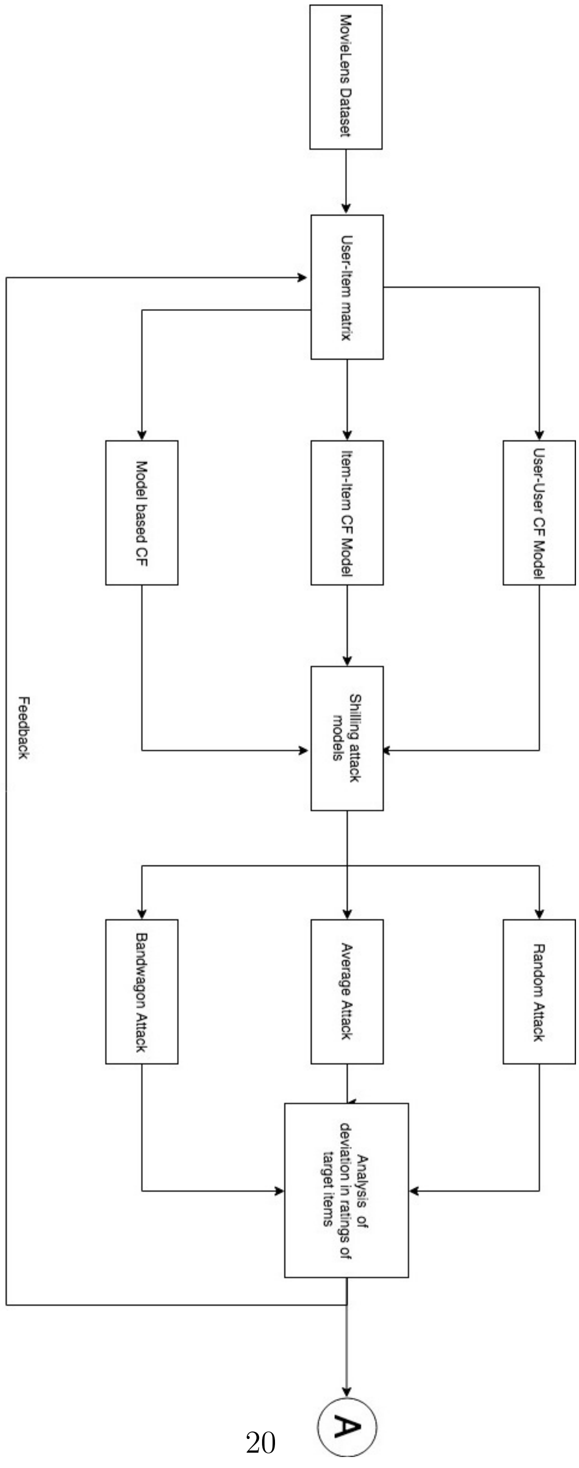
The most common way models shilling attack detection as a classification problem, and utilizes supervised or semi-supervised learning technique for training models. This type of detectors is feature-based, that is, it first defines a set of metrics and thus transforms each profile (i.e., rating record) into a vector in the feature space. Various classification models, such as C4.5, SVM, Nave Bayesian, etc, can then be used. In this section, we introduce feature selection and thus supervised shilling detectors in a wrapped view. A. Feature Selection In the literature, a number of features are defined for distinguishing shilling attackers against normal users. However, some of them are not disjointed. Altogether 10 features are collected from the literature, including Entropy, DegSim, LengthVar, RDMA, WDMA, WDA, MeanVar, FMTD, GFMV, and TMF. Due to the limited space, we omit the definitions of these features here. In fact, among 10 candidate features, some of them are not disjointed. For example, WDMA and WDA are the same as RDMA except on the normalization method. Meanwhile, a feature is usually designed purposefully for one attack model. For instance, FMTD is designed for bandwagon attacks and MeanVar is intended for average attacks. Therefore, it is necessary to adopt the feature selection to screen several discriminative features. Feature selection is also a supervised learning process, which is not in conflict with the supervised detectors. That is, based on labeled instances for training, feature selection tries to find a subset of most discriminative features on the training data. Since the our target is to compare shilling attack detectors rather than feature selection algorithms, we here use a simple heuristic MC-Relief to select the right metrics for the subsequent supervised training. For clarity, we give a brief introduction to the MC-Relief heuristic. Generally, MC-Relief aims to estimate the weight of features to distinguish the user profiles that are near to each other. Given m features l (1 l m) is the weight of a feature. For Estimating l, MC-Relief does a random sampling on training data. Each time, assuming the user u is picked out, us nearest neighbors in c different classes are searched, and then the weight of each feature will be updated once. Intuitively, the larger difference of a feature on two instances in the same class, the less discriminative of this feature, and vice versa for a pair of instances from different classes.

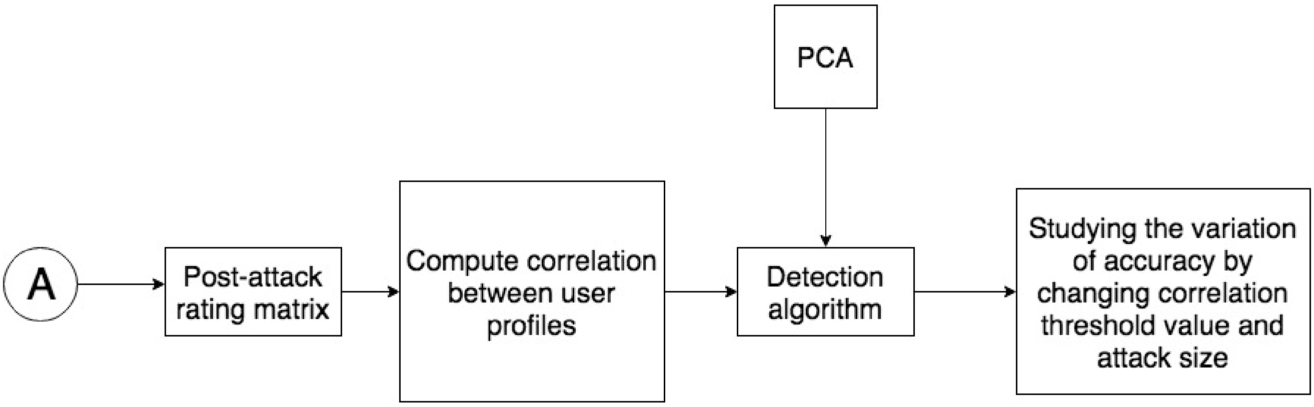
2.1 Research Gap Identified

- In SVM-TIA, genuine profiles may also be identified as attacker profiles
- Accurate trust modelling in spam user detection model has not become viable yet
- To efficiently utilise a large dataset through distributed computing for more accurate results

Chapter 3

Design




A
Post-attack rating matrix
Compute correlation between user profiles
PCA
Detection algorithm
Studying the variation of accuracy by changing correlation threshold value and attack size

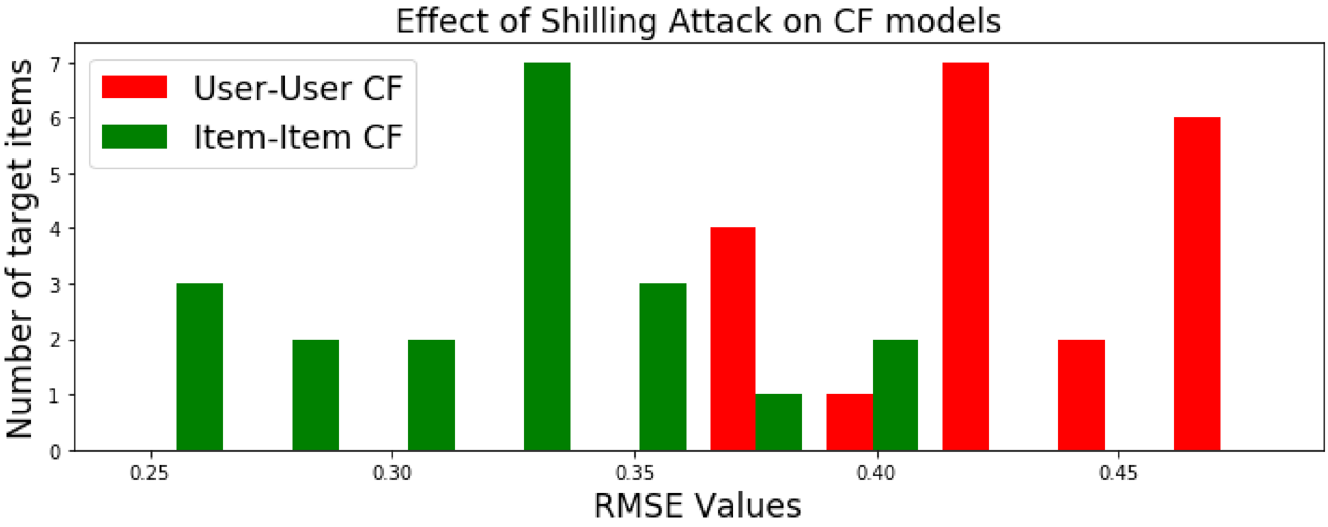
Chapter 4

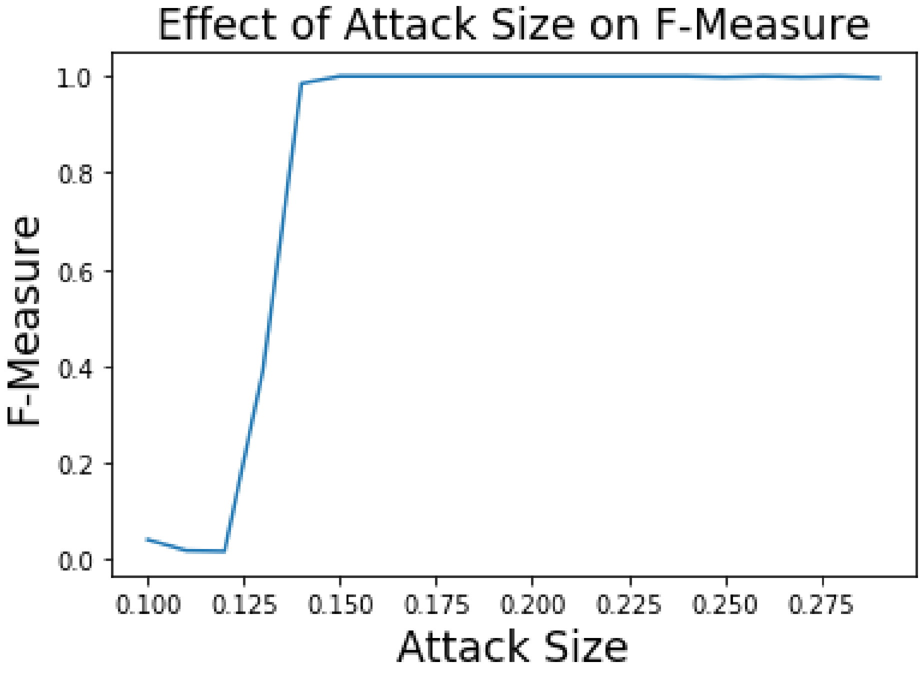
Implementation

- Perform various shilling attacks. Different models used are Average attack, Random attack and Bandwagon attack.
- Compute correlation between the users.
- Apply PCA Detection, a profile is detected as shilling profile if it has correlation equal to or higher than a threshold and it has this correlation value with a number of users in the system(attack size).
- Vary the correlation threshold value and the attack size and compute accuracy. Observe change in accuracy depending on the change in parameters.

Chapter 5

Results and Discussion





Chapter 6

Future Scope

- Formulate proper heuristics to find the number of shilling profiles injected in the system such that profile threshold can be set accurately.
- The shilling detection algorithm can be implemented for any recommender system since it is using an unsupervised shilling attack detection and prevention technique.

Chapter 7

Conclusions

- Accuracy of shilling profile detection depends on certain parameters.
- On increasing the filler size, the accuracy increases.
- By varying the correlation threshold value and profile threshold, the accuracy also varies.
- On increasing the size of the target item set, the accuracy also increases.

Bibliography

[1] Zhihai Yang and Zhongmin Cai 'Detecting abnormal profiles in collaborative filtering recommender systems', *J. Intell. Inf. Syst.*, Yang C17, India, pp. 499–518,2017.

[2] Gunes, Ihsan and Kaleli, Cihan and Bilge, Alper and Polat, Huseyin, 'Shilling Attacks Against Recommender Systems: A Comprehensive Survey,' *Artif. Intell. Rev.*, vol. 42, pp. 767–799, Dec 2014.

[3] Xiaoyuan Su and Taghi M. Khoshgoftaar, 'A Survey of Collaborative Filtering Techniques' *Hindawi Publishing Corporation Advances in Artificial Intelligence* , Volume 2009, Article ID 421425, 19 pages, August 2009.

[4] Manjeet Kaur and Shalini Batra, 'A Novel Trust Mechanism for Collaborative Recommendation Systems,' *Computing and Network Sustainability: Proceedings of IRSCNS 2016 published by Springer Singapore*, pp. 343-351, Oct 2016.

[5] Wei Zhou , Junhao Wen , Qingyu Xiong , Min Gao, Jun Zeng, 'SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems,'*Elsevier*, 19 October 2016

[6] Zi-Jun Deng , Fei Zhang , Sandra P. S. Wang , 'Shilling Attack Detection In Collaborative Filtering Recommender System By PCA Detection And Perturbation', *2016 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pp.213-218, July 2016

[7] Youquan Wang, Lu Zhang , Haicheng Tao , Zhiang Wu, Jie Cao,'A Comparative Study of Shilling Attack Detectors for Recommender Systems,' *IEEE 2015 12th International Conference on Service Systems and Service Management (IC-SSSM)*, pp. 1-6, June 2015.