공학석사 학위논문

STA : Sybil Types-aware Robust Recommender System

시빌 유형을 고려한 강건한 추천시스템

2015년 2월

서울대학교 대학원

전기 · 컴퓨터공학부

노 태 완

STA : Sybil Types-aware Robust Recommender System

지도교수 김 종 권

이 논문을 공학석사 학위논문으로 제출함

2015 년 2 월

서울대학교 대학원

전기 · 컴퓨터 공학부

노 태 완

노태완의 공학석사 학위논문을 인준함

2015 년 2 월

위 원 장 : 전 화 숙 (인)

부위원장 : 김 종 권 (인)

위 원 : 권 태 경 (인)

Abstract

STA : Sybil Types-aware Robust Recommender System

Taewan Noh

School of Electrical Engineering and Computer Science

The Graduate School

Seoul National University

Recently many users refer to various recommender sites when they buy things, movies, music and etc with a rapid development of internet. But there are malicious users (Sybil) to raise or lower ratings of items intentionally in these recommender sites, finally recommender system can recommend incomplete or inaccurate results to normal users. We suggest a recommender algorithm to separate ratings which users generate into normal ratings and outlier ratings and to minimize effects of malicious users. In addition, it provides stable RS about three kinds of models (Random attack, Average attack and Bandwagon attack) which are making problems

in Recommender system now. To prove performances of suggesting method, we conducted performance analysis to collect real data (crawling). As a result of performance analysis, it is proved that a performance of suggesting method is good regardless of Sybil size compared to existing algorithms.

Keywords: Recommender system, Sybil attack, Sybil Attack Models, Robust Recommender System

Student Number: 2013-20787

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Background

Lately social network service has been activated. So recommender system is popular to recommend appropriate items to users. These recommender systems are using various recommending algorithms to suggest differentiated and adequate items for each user. In addition, users directly visit many sites with recommender system to see information and ratings about items they want or to get recommendations. Related sites are Watcha (watcha.net), Naver Movie (movie.naver.com), Auction (auction.co.kr) and etc.

These sites have easier and more accessible structures without special security for users to utilize them compared to existing systems requiring complex certification. As a result, there are possibilities of Sybil attack which maliciously manipulate ratings of movies, items and etc what they want. When Sybil attack happens, related system is hard to conduct accurate recommendation due to effects of false information. In fact, it is common that movie advertising agencies or companies to operate recommender sites manipulate ratings of their or competitors' items. For

example, a manufacturer of alcoholic beverages in South Korea accused a rival company of malicious comments intentionally to the prosecution. And it was a big issue that movie companies hired people to give good ratings to their movies.

Therefore recently there are suggestions for Robust Recommender system to protect from Sybil attack [1][2][3]. But existing researches assumes that RS datasets of recommender systems are composed of normal data without effects of Sybil attack, so various kinds of Sybil attacks such as Random, Average, Bandwagon, Segment Attack and etc were generated randomly to improve the strong level of recommender systems[4].

Sybil attack is divided into Push attack which raises ratings maliciously and Nuke attack which lowers ratings malicious. Sybil who is the subject of conducting Sybil attack can conduct Push attack which gives much higher ratings than normal users to raise their items' ratings or Nuke attack which gives lower ratings to degrade competitors' ratings. But there can be users who give ratings in different ways compared others though they are not malicious users like Sybil, it is difficult to figure out normal users who don't follow the average from malicious ones.

1.2 Goal and Contribution

The purpose of this research is to build robust recommender system. We suggest the robust RS algorithm considering three kinds of attack models such as Random attack, Average attack, Bandwagon attack of present Sybil types. Algorithms we suggest show good performances while Sybil attack size increasing. In addition, Average attack and Bandwagon attack which are strong attacks to RS in existing researches show better performances than Matrix Factorization (MF)[5] method and Least Trimmed Squares Matrix Factorization (LTSMF)[3] method.

Contribution of this research is the following.

- In this research, detect Sybil users using the characteristic that they give higher ratings to items they want to manipulate than normal users. Systems restrict discovered Sybil users so that users are provided strong recommender systems.
- In this research, provide strong recommender systems about Random attack, Average attack, Bandwagon attack which are attack models of present recommender system. We conducted crawling not only Movielens data but also Naver-movie which is the most popular movie recommendation site in Korea ourselves for experiments. To evaluate performances, STA was proved that it was strong recommender system comparing with other robust RS, LTSME algorithm.

- In this research, conduct the research reflecting the reality compared to existing researches that assume the initial data are composed of normal users' ratings. In other words, assuming that existing dataset is affected by abnormal users who are malicious users or who don't follow the average, suggest a methodology to suggest best items for each user to minimize their effects.

1.3 Thesis Organization

The rest are the following. We will study Recommender system, Robust Recommender system and Sybil attack type in Chapter 2. In Chapter 3, we will explain details of Sybil Types-Aware Robust Recommender System (STA) we suggest. In Chapter 4, we will show the performance of our algorithm. And finally in Chapter 5, this research will end with conclusion.

Chapter 2 Related Work

2.1 Recommender System

Recommender system is a system to suggest appropriate items to users reviewing users' history of use. Many RS sites predict ratings of new items and recommend good items using users' history of use to recommend good items to users. There are many methods to predict users' ratings of new items. The most popular one to predict ratings of users is Collaborative Filtering (CF) [6][7].

CF model is generally divided into two categories of memory-based and model-based.

First, memory based method is divided into user-based method and item-based method. User-based method [8][9] predicts the rating using other users' ratings who have similar preferences when users' ratings of items are predicted. On the contrary, Item-based method [10][11] predicts the rating using ratings of similar items which the user gave ratings. When using User-based method and Item-based method, Vector Space Similarity (VSS) or Pearson Correlation Coefficient (PCC) methods are used to get similarity between items or users. PCC method is considering that each user have different rating styles and shows better performance than VSS.

Second, Model-based method is a method to make and use a model composed of hidden features. Examples of this method are Latent factor model [12], Bayesian hierarchical model [13], Clustering model [14] and etc. Matrix factorization method is using item-user rating matrix and latent factor to make item-latent factor matrix and user-latent factor matrix. Each matrix shows tendency of items or users using hidden features. Matrix factorization method is one of the most popular methods to predict well though data are sparse. Developments of Social network like SNS and others gives chance to study methods to predict more accurate ratings using friends of users or people in trust relations [15].

2.2 Robust Recommender System

Robustness is an ability to operate system in stressful conditions. Recommender system has many stresses, but researches about RS robustness are focusing on improving performance when dataset is stressed. Especially they are when dataset is noisy or full of erroneous data, or when Sybil which intends to raise or lower ratings of items it wants attacks.

The purpose of Robust recommendation is to prevent attacks to manipulate RS by large-scale insertion of false user profiles. Attackers infect a lot of Sybils in recommender system to reach the goal they want. An attack of Sybil in recommender system to make these false profiles is suggested first in [16]. And then classification of profile injection attacks was suggested in [17]. And it was suggested that attacks on memory-based and model-based recommendation algorithm of attackers [18][19]. [18] shows five attack models of Sampling attack, Random attack, Average attack, Bandwagon attack, Segment attack and effects of these affect models on RS. Especially average attack is more accurate and effective than Random attack. Bandwagon also has similar effects with Average attack. But Bandwagon attack and Random attack don't have effects in item-based collaborative filtering. These five attack models express items which Sybil wants to manipulate as target items. Push attack is that Sybil tries to raise ratings of target item and Nuke attack is to lower them. And they use Filler item to give ratings of other items

showing they are more normal users. Robustness of model-based CF such as Factor analysis models and k-means clustering model is suggested in [20], [21], [22]. Average attack profiles are suggested in PCA-based detectors [23] using very highly correlated ones. And LTSMF methods based on least square is suggested using that attackers always give higher or lower ratings than normal users[3].

2.3 Sybil Attack Type

Sybil attack is divided into Push attack which raises ratings maliciously and Nuke attack which lowers ratings malicious. Sybil who is the subject of conducting Sybil attack can conduct Push attack which gives much higher ratings than normal users to raise their items' ratings or Nuke attack which gives lower ratings to degrade competitors' ratings. At this moment, in case of Push attack, Sybil gives the best ratings on items to manipulate and in case of Nuke attack, it gives the worst ratings to manipulate. The most basic models are Random attack model and Average attack model which were suggested by Lam and Ridel[4]. Two models include not only ratings of target item to manipulate in attack profile but also randomly selected filler items.

- **Random Attack**

As mentioned before, there are ratings of filler items and target items attack users want to manipulate in Random attack profile. Filler size is decided depending on how many filler items are selected. When giving ratings of filler items, malicious users give ratings Global items mean on all items. And they give ratings maximum or minimum ratings depending on Push attack or Nuke attack about target items.

- **Average Attack**

Average Attack, more powerful attack is introduced by Lam and Ridel[4]. The difference with Random attack is to give ratings each item mean on each item, not each item mean on all items when rating on filler items, so that attacks become more powerful. Because system judges that each item mean on each item is a user to give rating, it is more difficult to detect malicious users judging them as normal user.

- **Bandwagon Attack**

The purpose of Bandwagon attack is to attack with more power to connect target item they want to attack and the most popular items (which users give many ratings). This attack is using that popularity between users and items follows Zipf's law distribution. It is using that the most popular one in many items has small number. Assuming that these items are selected item, other users can access easily to give the maximum ratings on the most popular ones, in other words, items with many ratings. The most popular items are easy to attack strongly because it is easy to recognize without access of them. Bandwagon attack can make to seem like a general user using filler items like Random attack and Average attack. At this moment, global item mean like Random attack are used in rating filler item.

Chapter 3 System Model

3.1 Overview

Robust Recommender System which is suggested has improved a performance of recommender system except malicious users, using many Sybil attack models existing now. Generally users who want to manipulate ratings of items they want tend to give the highest ratings on items they want or the lowest items to raise each item mean of items they want. Because each item mean to manipulate is not high, Sybil users give the highest ratings by Push attack to raise each item mean. This value of rating is different from other ratings of general users and has outlier characteristics. In addition, attacking effects are increased by using filler items so that system cannot distinguish malicious users, or using selected items for more effective attack. Suggested algorithm is expressed in Figure 3.1. System is investigating ratings of each user's items and calculates a probability that users are Sybil. System calculates the number of outlier ratings, filler items and selected items among ratings of users while giving the highest ratings. Considering the calculated number of three, Sybil user probability is calculated. When Sybil user probability of each user is over Threshold T, system judges the user as Sybil. If it is lower than Threshold T, it judges the user as a normal user. System provides strong

recommender system to users excluding data users who have high possibilities to be Sybil user. STA RS protect systems from various Sybil attacks and suggests appropriate items users want based on safe system.

To measure how effective STA (Sybil Types-Aware Robust Recommender System) is, experiments was conducted in comparison of LTSMF methods and Basic Matrix Factorization methods applying many attack models to Movielens data. Now Naver-movie are supposed to have infected data by users who try to raise or lower ratings maliciously. The effect was proved in these infected data applying STA algorithm. In this research, experiments were conducted only assuming Push attack. STA algorithm is generally divided into 1. Initialization phase, 2. Detecting Sybil user probability of each user phase, 3. Prediction of rating phase.

.

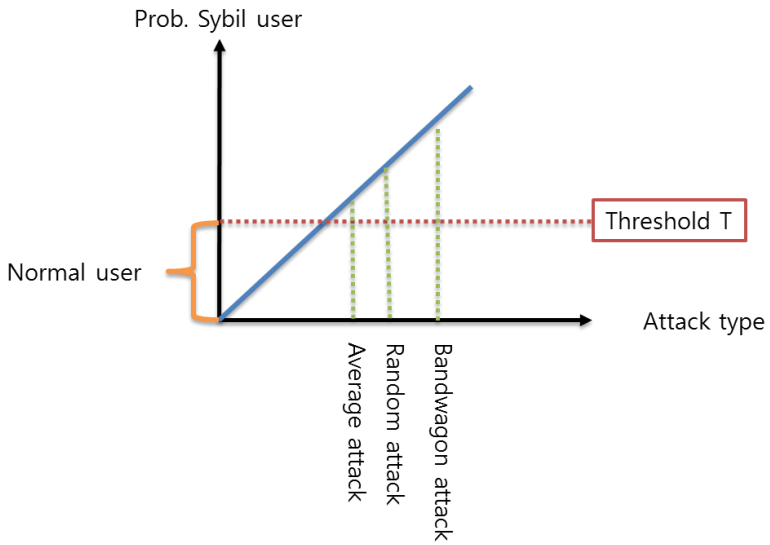

Figure 3.1 Probability of Sybil User

3.2 Notations

Figure 3.2. shows rating matrix which is used in this research. When the number of users is M and the number of items is N, a matrix can express ratings of items as rating matrix, R_{MxN}. Item i_n means item in order of N, and $n \in [1,N]$. User u_m means user in order of m, $m \in [1,M]$. R(m,n) means a rating on Item n of User m. Ratings are depending on systems and from minimum 1 to maximum 5 point or from minimum 1 to maximum 10 generally. If User m does n't give rating on Item n, it does n't have the value. And a user can give only one rating on one item. Table 1 explains notations in this research.

	Item$_1$	**Item$_2$**	…	**Item $_n$**	…	**Item $_N$**
User 1	$r_{(1,1)}$		…	$r_{(1,n)}$	…	$r_{(1,N)}$
User 2	$r_{(2,1)}$	$r_{(2,2)}$	…	$r_{(2,n)}$	…	$r_{(2,N)}$
..	..		…	…	…	…
User m		$r_{(m,2)}$	…	$r_{(m,n)}$	…	…
..	..	…	…	…	…	
User M	$r_{(M,1)}$		…	$r_{(M,n)}$	…	$r_{(M,N)}$

Figure 3.2 Rating Matrix (R_{MxN})

Notation	Description
$\mathbf{N}$	The number of items
$\mathbf{m}$	The number of users
$\mathbf{i_n}$	n-th item
$\mathbf{u_m}$	m-th user
$\mathbf{r_{m,n}}$	Rating value for item n and user m
$\mathbf{i_{(n,mean)}}$	Mean of item n
$\mathbf{i_{(n,\ STD)}}$	Standard deviation of item n
$\mathbf{i_{Global}\ (mean)}$	Mean of item N
$\mathbf{i_{Global}\ (STD)}$	Standard deviation of item N
Max Rating Value	Maximum rating value
Min Rating Value	Minimum rating value
$\mathbf{C_m^O}$	The number of Outlier item with maximum rating value which user m rated
$\mathbf{C_m^F}$	The number of Filler item which user m rated
$\mathbf{C_m^S}$	The number of Selected item which user m rated
$\mathbf{P_{sybil}(m)}$	Probability of user m being Sybil
$\mathbf{T_{rating}(m)}$	The number of ratings by user m
$\mathbf{R_{MxN}}$	Rating matrix consist of user M and item N

Table 3.1 Notations in this thesis

3.3 Initialization

Algorithm 1 : Initialization

Input: $\mathbf{R_{MxN}}$

Output: $\mathbf{i_{(1,mean)}}$, $\mathbf{i_{(2,mean)}}$, … , $\mathbf{i_{(N,mean)}}$, $\mathbf{i_{(1,STD)}}$, $\mathbf{i_{(2,STD)}}$, … , $\mathbf{i_{(N,STD)}}$
$\mathbf{i_{Global}}$ **(mean),** $\mathbf{i_{Global}}$ **(STD)**

```
1 for x=1 to N do
2 |  Calculate the average and STD of the item x from rating matrix
3 end
```

Calculate the item's Global average and Global STD
Calculate Most rated item list

Figure 3.3 The initialization phase procedure

Sybil users have different characteristics compared to other users' ratings on target item, because they give the highest ratings on target item. To distinguish these outlier characteristics, the average and standard deviation of items are necessary. And filler item can follow each item mean or global item mean, so that these values are calculated. Selected item is to give the highest rating on the most popular item, which has the most ratings. To figure out items with the most ratings, Most rated item list are calculated. At this moment, the list is made with the standard as much as the top P(Most rated item)%.

3.4 Sybil User Probability Algorithm

Algorithm 2: Calculate Sybil User Probability Algorithm

Input : $R_{MxN,}$

Output : $P_{sybil}(1), P_{sybil}(2), \dots, P_{sybil}(m)$

```
1  for m =1 to M do
2      for n = 1 to N
3          if r_{m,n} has value then
4              Calculate Outlier Rating (r_{m,n}, i_{(n,mean)}, i_{(n,STD)})
5              if r_{m,n} is Outlier rating then
6                  C_m^O = C_m^O + 1
7              end
8
9              Calculate Filler Item (r_{m,n}, i_{(n,mean)}, i_{Global (mean)})
10             if r_{m,n} is Filler item then
11                 C_m^F = C_m^F + 1
12             end
13
14             Calculate Selected Item (r_{m,n}, Most rated list)
15             if r_{m,n} is Selected item then
16                 C_m^S = C_m^S + 1
17             end
18
19         end
20     end
21     P_{sybil}(m) = (α · C_m^O + β · C_m^F + γ · C_m^S) ÷ T_rating(m)
22 end
```

```
Return P_sybil(1), P_sybil(2), … , P_sybil(m)
Calculate Outlier Rating(r_m,n, i_(n,mean), i_(n,STD))
{
    if r_m,n is Max Rating Value then
        z = (r_m,n − i_(n,mean)) / i_(n,STD)
    end
        if   z < -1.0 or z > 1.0 then
            r_m,n is outlier rating value
        end
}

Calculate Filler Item (r_m,n , i_(n,mean), i_Global (mean))
{
    if r_m,n == i_(n,mean) or r_m,n == i_Global (mean)
        r_m,n is Filler item
    end
}

 Calculate Selected Item (r_m,n, Most rated list)
{
    if r_m,n is in Most rated item list and r_m,n is Max Rating Value
        r_m,n is Selected item
    end
}
```

Figure 3.4 Calculate Sybil User Probability Algorithm

Sybil User Probability algorithm is an algorithm to calculate the probability how many users are Sybil investigating all ratings each user gives. Looking at the order,

this algorithm is checking whether every rating user gives are outlier comparing to others **(Calculate Outlier Rating)**. Second, it checks whether rating value is filler item **(Calculate Outlier Rating)**, and then investigate whether the rating value is selected item **(Calculate Selected Item)**. Last, it sums all values and divides the number of all ratings of user give so that it calculates the probability that the user is Sybil.

After previous Initialization part, system calculates the probability whether each user is Sybil, using Sybil User Probability algorithm. This probability would increase much when each rating value is related to attack models or it shows different tendency compared to normal users, investigating all ratings of each user. When investigating each rating of users, first of all, this rating is check whether it is target item in Calculate Outlier Rating function. Malicious users give the highest ratings in target item to increase the average ratings of items they want. They generally manipulate items which don't have high ratings. When they give the highest ratings different from normal users' ratings, they can have outlier characteristics. Therefore ratings are check whether they are the highest in Calculate Outlier Rating (line 26). If this rating is check to be the highest one, it is checked whether it is outlier compared to ratings of general users' items using normal distribution (line 27). If normal distribution Z is below -1, above 1.0, this rating is outlier rating and distinguished by ratings of target item. Therefore $c_m{}^o$ value should

be increased.

Because Random attack and Average attack are using filler item in Calculate Filler Item, each ratings should be check whether they are ratings on filler item. When Random attack and Bandwagon attack use filler item, filler item is randomly selected and average rating of all items is given as a rating value. Using filler item can make to seem like general users and has advantage to lower average rating of their owns. Average attack is more powerful attack than Random attack. When it calculates rating value of filler item, it calculates average rating values of each item, not every item compared to Random attack. Because average rating value of each item is rating value of filler item, they become general users more and more. Therefore in Calculate Filler Item each rating value is checked whether it has rating value ($i_{Global\,(mean)}$) of all items which were already calculated in Initialization part and rating value ($i_{(n,mean)}$) of each item (line 36). If they have same values, they are distinguished as filler item of Random attack, Average attack and Bandwagon and increase 1 of $C_m{}^F$.

Selected item are used in Bandwagon attack so that each ratings are check whether it is rating about selected item in Calculate Selected Item. The purpose of Bandwagon attack is to attack more powerfully to connect target items they want to attack and the most popular items (which users give the most ratings). It is using that the most popular one in many items has small number. Assuming that these items

are selected item, other users can access easily to give the maximum ratings on the most popular ones, in other words, items with many ratings. The most popular items are easy to attack strongly because it is easy to recognize without access of them. To check whether it is selected item, they should check that users' ratings are about popular items. And rating values should be check that they are maximum value (line 43). Popular item is supposed to be items with many ratings. At this moment, Most rated item list about items with the most ratings in Initialization part is used. The list is made with the standard as much as the top P(Most rated item)% and this research conducted experiments as 1%, 5%, 10%, 15%, 20% which has the most ratings among all items. If user' rating is selected item, $C_m{}^S$ value should be increased 1. $P_{sybil}(m)$ is a probability that User m is Sybil user. Values of $C_m{}^O$, $C_m{}^F$ and $C_m{}^S$ from every rating of users are multiplied by each α, β, γ and then divided by the number of User m's all ratings. ($\alpha + \beta + \gamma = 1$, $\alpha > \beta > \gamma$). The biggest difference between Sybil users and normal users might be $C_m{}^O$. Therefore α value is bigger than β, γ values. Filler item has low γ value because normal users who give many ratings can be distinguished as filler item. β value is smaller than α, and bigger than γ value. More number of ratings, higher probability of Sybil user, Sybil User Probability is calculated to divide total number of ratings.

3.5 Remove Sybil User from Rating Matrix

Algorithm 2: Remove Sybil User from Rating Matrix

Input : $R_{MxN,,}$ $P_{sybil}(1), P_{sybil}(2), \ldots, P_{sybil}(m)$

Output : $R_{STA\ (MxN)}$

```
1  for m =1 to M do
2      if P_sybil(m)  > Threshold T
3          remove User M in the Rating Matrix(R_MxN)
4      end
5  end
6  Return R_STA (MxN)
```

Figure 3.5 Remove Sybil User from Rating Matrix

Using Sybil User Probability of each user which was calculated in 3.4, users who has big possibility of Sybil in Rating Matrix should be removed. Investigating all users, users whose Sybil User Probability is bigger than Threshold T should be removed in Rating Matrix. $R_{STA(MxN)}$ is made to remove users with big possibilities of Sybil user.

3.6 Rating Prediction phase

For rating prediction, any rating prediction schemes among the model-based CF can be applied. We use the Matrix Factorization (MF) since it is one of the most effective method in collaborative filtering, which considers both user preferences and item characteristics could be explained by some numbers of latent factors. If m x n rating matrix R describing m users' numerical ratings on n items, low-lank matrix factorization approach can seeks to approximates the rating matrix R by a multiplication of l-rank factors,

$$R \approx U^T V \tag{3.6.1}$$

where $U \in R^{l \times m}$ and $V \in R^{l \times n}$ with l < min(m,n). Traditionally, the Singular Value Decomposition (SVD) method it used to a rating matrix R by minimizing

$$\frac{1}{2}||R - U^T V||_F^2 \tag{3.6.2}$$

where $||\cdot||_F^2$ denotes the Frobenius norm. Since rating matrix is usually extremely sparse, we can use indicator function. Therefore, we change Eq. (3.6.2) to

$$\min_{U,V} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{i,j}\,(R_{i,j} - U_i^T V_j)^2 \tag{3.6.3}$$

where $I_{i,j}$ is the indicator function that is equal to if user u_i rate item v_j and equal to

0 otherwise. In order to avoid over fitting, two regularization terms are added into Eq. (3.6.3).

$$\min_{U,V} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{i,j}\,(R_{i,j} - U_i^T V_j)^2 + \lambda_1 \frac{1}{2}||U||_F^2 + \lambda_2 \frac{1}{2}||V||_F^2 \qquad (3.6.4)$$

Then, we can find a local minimum by using Gradient based approaches.

Chapter 4 Evaluation and Analysis

4.1 Datasets

name	#users	#items	#ratings	rating scale
Movielens	1394	2590	100,000	{1,2,…,5}
Naver-movie	9,913	677	360,467	{1,2,…,10}

Table 4.1 Dataset characteristics

Data which we used in experiments is like Table 4.1. Above all, Movielens site is a site to recommend not only information of movies to users but also appropriate movies for each user. Naver movie site is favorite movie recommender site in Korea. Data which we conducted crawling are 1394 users of Movielens data and 2,590 movies they gave ratings. Users can give ratings from 1 to 5 point. Naver-movie data are ratings of movies released from 2011 to 2012. They are data of 9,913 users about 677 movies with more than 10 ratings. Each user gave at least 5 ratings. Naver

movies can give ratings from 1 to 10 point.

4.2 Metrix

To compare our approach and other previous approach, we use Mean Absolute Error (MAE) to measure the predict accuracy of these methods.

The metrics MAE is defined as,

$$\mathrm{MAE} = \frac{1}{T}\sum_{i,j} |R_{i,j} - R'_{i,j}| \ , \qquad (4.2.1)$$

where R_{ij} denotes the rating user i gave to item j, R'_{ij} denotes the rating user i gave to item j as predicted by a method, and T denotes the number of tested ratings.

4.3 Experimental Setup

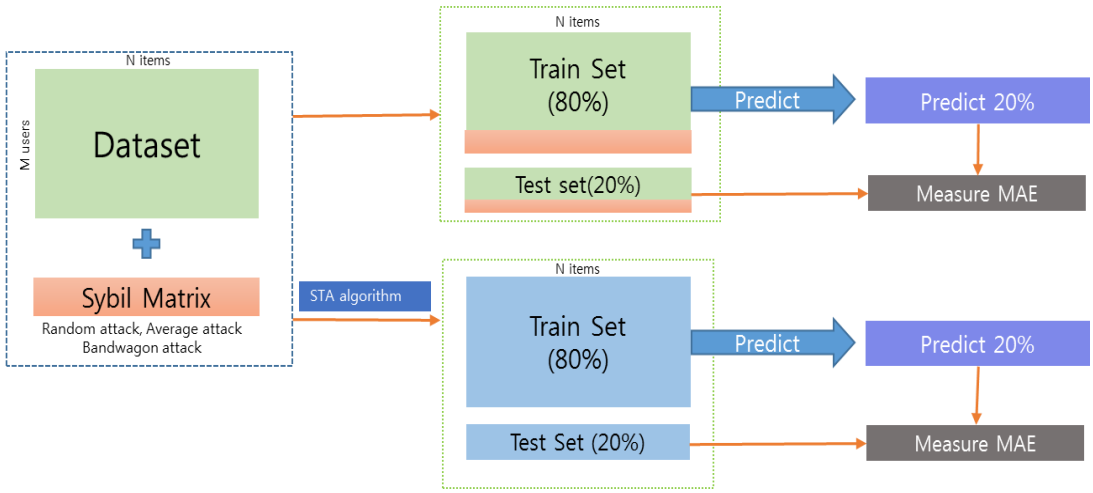

Figure 4.1 Experimental scenarios with three different attack types

We first composed attacked dataset in scrawling Movielens dataset applying three kinds of attack models (Random attack, Average attack, Bandwagon attack). Attack size was composed of 1%, 5%, 10%, 30%, and 50% of all normal users. This dataset is composed of 80% of Train set and 20% of Test set. 20% of hidden ratings were predicted using 80% of Train set, and accuracy was analyzed by Mean Absolute Error (MAE) comparing these values with 20% of Test set. Each experiment was conducted 10 times and the average was shown in graph. And in this research,

conduct the research reflecting the reality compared to existing researches that assume the initial data are composed of normal users' ratings. In other words, STA performance was measured which is assuming and suggesting that existing dataset are affected by abnormal users who are malicious users or who don't follow the average. Naver-movie site many Koreans use had a problem that ratings of movies are too low or high because movie companies or advertising agencies manipulate artificially. Therefore experiments were conducted using dataset of Naver-movie which were affected by malicious users already.

Comparing algorithm in this research, we measured LTSMF method against MF method. First, MF method predicts ratings simply applying MF method in dataset. Second, though our algorithm also uses MF method, STA algorithm applies MF method after finding Sybil users and removing them in dataset. Last, LTSMF method is similar with MF method, but the difference is a method that it can be far away from singularities, without using the biggest residuals' squares.

4.4 Experimental Results and Analysis

4.4.1 Impact of Most rated items

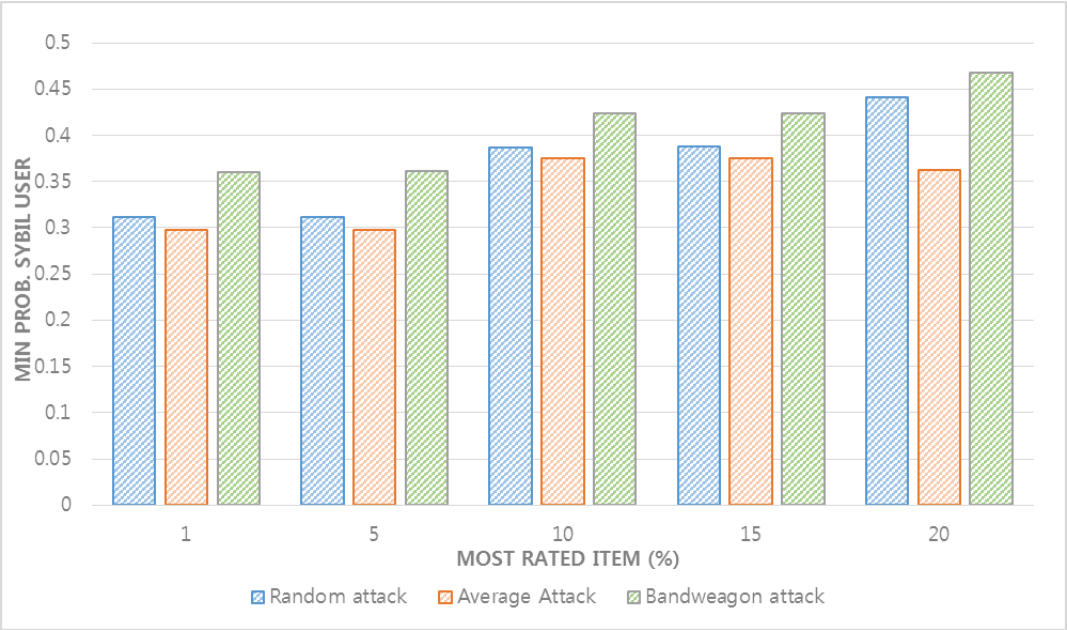

Figure 4.2 Impact of Most rated items to Minimum probability of Sybil user

The experiment used Movielens data and was conducted when Sybil attack size is 1%. That is, 1394 normal users and 14 Sybil users, 1% of normal ones are added in

data. Filler size is about 26 items, 1% of it and variables which can lead the best result are set as α=0.05, β=0.35, γ=0.6 values.

In STA algorithm, to judge whether ratings of users are selected items, popularity of each item are checked in the top P%. At this moment, experiments are conducted changing p% into 1%, 5%, 10%, 15%, 20%. For example, calculating selected item in STA algorithm, the item is calculated as selected item when it is in the top 1% and giving the highest rating. Then y axis is the lowest value of Sybil User Probability among 14 Sybil users. It can be distinguished as Sybil user when it is generally over 0.3% in graph. Overall, in case of Random attack, because the average of all items is given and Average attack is giving the average of each item, Average attack seems to be more powerful. Therefore Average attack is more difficult to distinguish than Random attack. This algorithm is considering Sybil users' attacks on selected items, and it can detect Bandwagon attack which is a powerful attack. When detecting Sybil users, higher Most rated item (P %), higher Sybil User Probability.

But in Figure 4.3, other users also have higher possibilities to be distinguished as Sybil, giving lower values increases probabilities to detect. Figure 4.3 shows that what % of Sybil User probability is among all users. Generally they have high Sybil user Probability within 5% of Sybil users. Like the above graph, Average attack is more difficult to detect than Random attack and Bandwagon attack. And higher Most rated item (P%), worse performance. The reason is that higher Most rated item (P%), more items that normal users give ratings also can be distinguished as selected item so that their performances are poor.

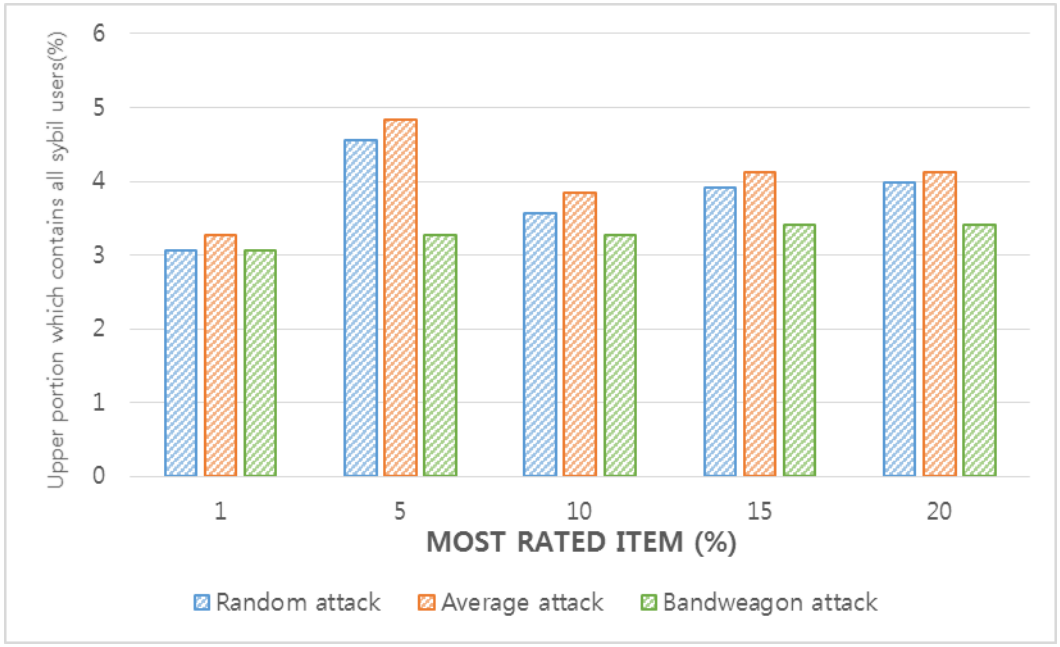

Figure 4.3 Impact of Most rated items

4.4.2 Performance on PS

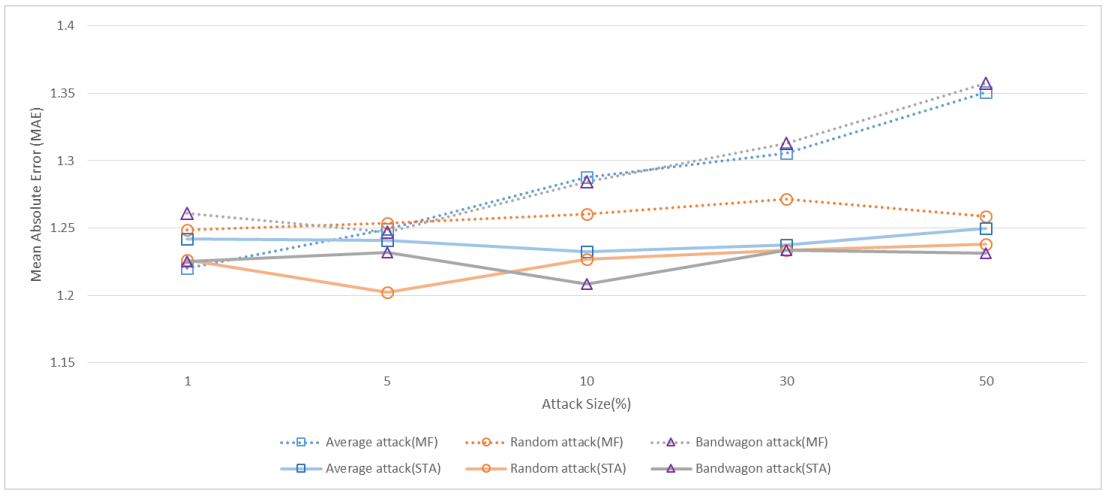

Figure 4.4 MAE Comparison with MF and STA

Figure 4.4 is comparing Basic Matrix Factorization (MF) and STA algorithms about three kinds of attacks in Movielens data. Changing Attack size into 1%, 5%, 10%, 30%, 50% of all honest users, Mean Absolute Error (MAE) were measured. First, general performance of STA is better than MF method. STA also uses MF method to predict ratings, but has better performances because dataset removes users with big possibility of Sybil. But bigger Attack size, worse performances in MF method. Especially Average attack and Bandwagon attack show worse performances than Random attack. Average attack and Bandwagon attack shows similar result, but Bandwagon attack is more difficult to detect using selected item. However in STA, considering not only filer item and target item of three kinds of attack models but also selected item of Bandwagon, it can be shown that Bandwagon attack was detected strongly. In addition, though attack size increased up to 50% of honest users, there was no difference compared to when attack size was 1%. This shows a big difference with results of MF methods. And there are good results about three kinds of attack models evenly.

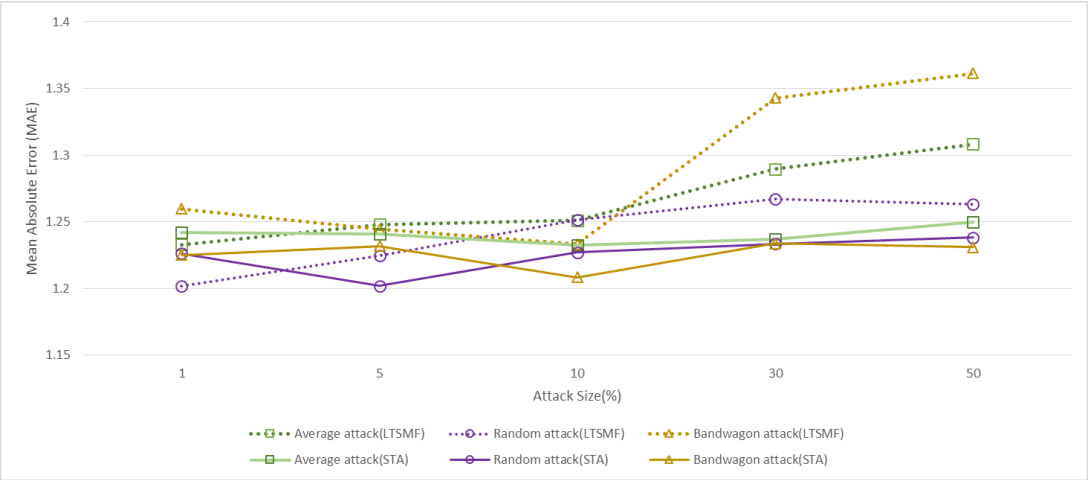

Figure 4.5 MAE Comparison with LTSMF and STA

Figure 4.5 is a graph to compare STA method and LTSMF method while changing attack size. First STA algorithm shows generally good performances regardless of

attack size. When attack size increases, it shows better predicting results than LTSMF algorithm. But LTFSMF algorithm gets worse rapidly when attack size becomes bigger. In LTSMF method, Bandwagon attack has the worst performance, in that order of Average attack and Random attack is strong. But STA algorithm shows the best results of Bandwagon attack. And three kinds of attack models show good performances without big differences with Random attack and Average attack.

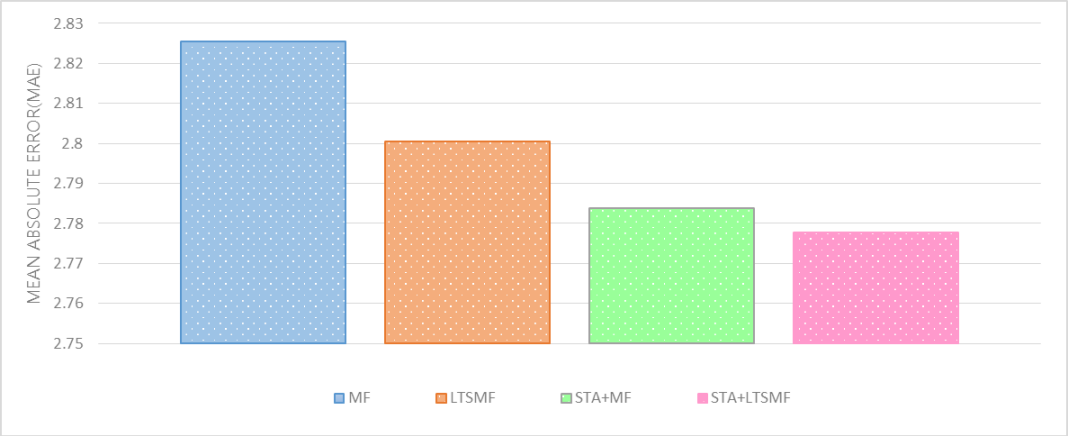

Figure 4.6 MAE Comparison with MF, LTSMF and STA using Naver-movie data

Figure 4.6 is a graph to compare LTSMF, MF and STA methods about Naver-movie.

Existing researches assumes that previous data are composed of normal users' ratings and suggested alternatives to make random Sybil users. But it is true that existing recommender sites are already affected by malicious users. Naver-movie data is supposed to be affected by malicious users too. We compared infected dataset with MF method, LTSMF, STA method. Naver-movie data are composed of movies which have at least 10 ratings and users who have at least 10 ratings. Simply data using MF method shows the highest values. Second, LTSMF and STA show the best performances. STA method is using MF method to predict ratings. This time, result of using MF method instead of LTSMF method shows the best performance. With these results, STA algorithm is operating well in dataset which is supposed to be infected by Sybil users already.

Chapter 5 Conclusion

We suggest Robust Recommender System algorithm which is called STA in this research. We suggest robust RS algorithm considering three kinds of attack models of Random attack, Average attack, Bandwagon attack about recommender systems of present Sybil. In this research, Sybil users are detected using the characteristic that they give higher ratings than normal users to manipulate ratings on items. System restricts discovered Sybil users and provides strong recommender system to users. We conducted crawling not only Movielens data but also Naver-movie which

is the most popular movie recommendation site in Korea ourselves for experiments. To evaluate performances, STA was proved that it was strong recommender system comparing with other robust RS, LTSME algorithm. And it provides that it shows good performances though attack size increases. In this research, the research was conducted reflecting the reality compared to existing researches that assume the initial data are composed of normal users' ratings. In other words, assuming that existing dataset is affected by abnormal users who are malicious users or who don't follow the average, suggest a methodology to suggest best items for each user to minimize their effects.

Bibliography

[1] H. Yu, C. Shi, M. Kaminsky, P. Gibbsons and F. Xiao, "Dsybil: Optimal Sybil-Resistance for Recommendation Systems", S&P '09

[2] N. Hurley, "Robustness of Recommender Systems", RecSys '11

[3] Z. Cheng and N. Hurley, "Robust Collaborative Recommendation by Least Trimmed Squares Matrix Factorization", ICTAI '10

[4] B. Mobasher, R. Bruke, R. Bhaumik, and C. Williams, Toward Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness, TOIT '07

[5] Y. Koren, R. bell, and C. Volinsky. "Matrix factorization techniques for

recommender systems", Computer, 43(8):30-37, 2009

[6] R. pan, Y. Zhou, B.Cao, N. N. Liu, R. M. Lukose, M. Scholz, and Q. Yang, "One=class collaborative Filtering, In IEEE International Conference on Data Mining (ICDM 2008), Pages 502-511.

[7] X. Su and T. M. Khoshgoftar. A survey of collaborative filtering techniques, Advances in Artificial Intelligence, 2009

[8] H. Ma, H. Yang, M. R. Lyu, and I. King. Social recommendation using probabilistic

matrix factorization, CIKM 2008, pages 931-940

[9] Rong Jin, Joyce Y. Chai, Luo Si, "An Automatic Weighting Scheme for Collaborative Filtering", In Proc. of SIGIR '04, JULY 25-29, 2004.

[10] Greg Linden, Brent Smith, and Jeremy York, "Amazon.com Recommendations Item-to-Item Collaborative Filtering", IEEE Internet computing, 7(1):76-80, 2003

[11] Mukund Deshpande and George Karypis, "Item-Based Top-N Recommendation Algorithms', ACM Transactions on Information Systems, 22(1):143-177, 2002.

[12] John Canny, "Collaborative Filtering with Privacy via Factor Analysis", In Proc. of SIGIR '02, pages 238-245

[13] Yi Zhang and Jonathan Koren, "Efficient Bayesian Hierarchical User Modeling for Recommendation Systems", In Proc. of SIGIR '07, pages 47-54

[14] Arnd Kohrs and Bernard Merialdo, "Clustering for Collaborative Filtering Applications", In Proceedings of CIMCA, 1999

[15] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu and Irwin King, "Recommender Systems with Social Regularization", In Proc. WDSM '11

[16] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre, "Promoting recommendations: An attack on collaborative filtering," in DEXA, ser. Lecture Notes in Computer Science, A. Hameurlain, R. Cicchetti, and R. Traunmuller, Eds., vol. 2453. Springer, 2002, pp. 494–503

[17] R. Burke, B. Mobasher, and R. Bhaumik, "Limited knowledge shilling attacks in

collaborative filtering systems," in Proceedings of the 3rd IJCAI Workshop in Intelligent Techniques for Personalization. IJCAI, 2005

[18] B. Mobasher, R. Burke, and J. Sandvig, "Model-based collaborative filtering as a

defense against profile injection attacks," in Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference. AAAI, July 2006.

[19] R. D. B. Jeff J. Sandvig, Bamshad Mobasher, "A survey of collaborative recommendation and the robustness of model-based algorithms," IEEE Data Engineering Bulletin, vol.31, no. 2, pp. 3–13, June 2008

[20] Z. Cheng and N. Hurley, "Robustness analysis of model-based collaborative filtering systems," in Proceedings of the 20th incarnation of the annual conference

on Artificial Intelligence and Cognitive Science. LNCS, August 2009.

[21] Z. Cheng and N. Hurley, "Effective diverse and obfuscated attacks on model-based recommender systems," in Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009. ACM, October 2009, pp. 141–148.

[22] Z. Cheng and N. Hurley, "Trading robustness for privacy in decentralized recommender systems," in Proceedings of The Twenty-First Conference on Innovative Applications of Artificial Intelligence. AAAI, July 2009

[23] B. Mehta, "Unsupervised shilling detection for collaborative filtering," in Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence. AAAI, July 2007, pp.1402–1407

요 약

최근 인터넷의 급 성장과 함께 사용자들은 물건이나 영화, 음악 등을 구매 할 때 여러 가지 추천 사이트를 참고한다. 하지만 이러한 추천 사이트에는 악의적으로 아이템의 평점을 높이거나 낮추려는 악의적인 사용자(Sybil)들이 존재하며, 결과적으로 추천시스템은 불완전하거나 부정확한 결과를 일반 사용자들에게 추천할 수 있다. 본 논문에서는 사용자들이 생성하는 평점들을 일반적인 평점과 일반적이지 않은(Outlier) 평점으로 구분

하고, 악의적 사용자의 영향력을 최소화 하는 추천 알고리즘을 제안한다. 또한 현재 Recommend System에서의 문제가 되고 있는 3가지 attack 모델(Random attack, Average attack and Bandwagon attack)에 대해서도 안정화된 RS를 제공한다. 제안하는 기법의 성능을 입증하기 위해 실제 데이터를 직접 수집 (crawling)하여 성능분석을 진행하였다. 성능분석결과 제안하는 기법의 성능이 기존 알고리즘과는 다르게 Sybil size에 상관 없이 좋은 성능을 보이는 것을 확인하였다.

주요어 : 추천 시스템, 시빌 공격, 시빌 공격 모델, 강건한 추천 시스템

학번 : 2013-20787