**Paper:**

# Experimental Analysis of Injection Attacks of Various Recommendation Systems using Real Data

**Soichiro Hashimoto, Hajime Nobuhara**

University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8577, Japan
E-mail: s2020777@s.tsukuba.ac.jp

[Received 21/08/02; accepted 21/08/02]

**Abstract. Recommendation systems play an important role in modern applications such as Netflix, Amazon, and e-commerce applications. Although the research for improving the accuracy of the recommendation system is increasing because of this importance, the risks in the latest recommendation system have not been sufficiently studied, and the importance of the recommendation system is increasing, and the attack to the improper system of the recommendation by the posting of the false review becomes a problem. The economic impact of the attack on the recommendation system is very large, and it is said that the purchase number of the item fluctuates by 5 to 9% depending on the increase and decrease of one star for the item evaluated by 5 stages. However, in a simple recommendation system, it is difficult to distinguish between a fake user and a normal user, and an algorithm to prevent attacks is required. In this paper, we focus on "injection attacks" among attacks on recommendation systems. An overview of the injection attack is shown in Figure 1. In the injection attack, an attacker creates a profile of a fake user from the data and injects it into a data set, and the data set is learned by the recommendation system. Learning without knowing the attack affects the recommendation function, and there is a risk of recommending different items. Although various studies have been conducted to verify the characteristics of such attacks, in the process of creating false data, the effects of optimization algorithms on attack performance have not been well studied. In this paper. In this paper, we investigate the attack performance of Bi-Level Optimization.**

**Keywords:** recommendation, attack detection, security

## 1. Introduction

Recommendation systems play an important role in modern applications such as Netflix, Amazon, and e-commerce applications. Although the research for improving the accuracy of the recommendation system is increasing because of these importance, the risks in the latest recommendation system have not been sufficiently studied, and the importance of the recommendation system is increasing, and the attack to the improper system of the recommendation by the posting of the false review becomes a problem. The economic effect by the attack on the recommendation system is very big, and it is said that the purchase number of the item fluctuates by 5 to 9% by the increase and decrease of 1 star in the item evaluated by 5 stars. However, in a simple recommendation system, it is difficult to distinguish a false user from a normal user, and an algorithm to prevent attacks is required. In this paper, we focus on "injection attacks"[1] among attacks on recommendation systems. An overview of the injection attack is shown in **Figure 1**. In the injection attack, an attacker creates a profile of a fake user from the data and injects it into a data set, and the data set is learned by the recommendation system. Learning without knowing the attack affects the recommendation function, and there is a risk of recommending different items. Although various studies have been conducted to verify the characteristics of such attacks, the impact of optimization algorithms on attack performance in the process of creating false data has not been well studied. In this paper. This paper investigates the change of attack performance when several kinds of attacker models are prepared and their optimization algorithm is Bi-Level Optimization[2].

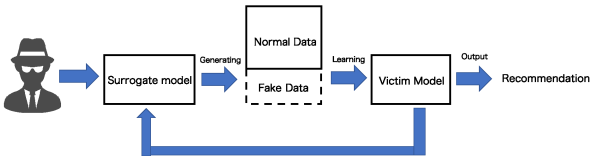


**Fig. 1.** Optimization Algorithm

## 2. Related Search

In this paper. A simplified Bi-Level Optimization algorithm is used in the learning and creation of false data. In creating a fake user, an attacker learns the behavior of the fake user to create fake data. Bi-Level Optimization algorithm is

$$\min_{\hat{X}} \mathscr{L}_{adv}(R_\theta), \quad \ldots \quad (1)$$

,

$$\theta^* = \arg\min_{\theta}(\mathscr{L}_{train}(X, R_\theta) + \mathscr{L}_{train}(\hat{X}, \hat{R}_\theta)). \quad (2)$$

.

the optimization algorithm can be expressed by two equations:. Equation (1) is called outer objective. Given a set of surrogate models and pseudo users $\mathscr{V}$, the pseudo data $\hat{X}\varepsilon\{0,1\}$ is learned by the loss function $\mathscr{L}_{adv}$. Equation (2) is called inner objective. $\theta$ is a parameter of the surrogate model, $R_\theta$ is usually a prediction of the surrogate model by the user, and $\mathscr{L}_{train}$ is a training object of the surrogate model. Thus, the inner objective is embedded in the outer objective. The inner objective first injects $\hat{X}$ of the false data and calculates a new parameter $\theta^*$ by learning the contaminated data. Finally, by using $\theta^*$ calculated by inner objective, the objective false data can be created by outer objective. The gradient is used to optimize the inner objective. the slope of the inner objective is

$$\nabla_{\hat{X}}\mathscr{L}_{adv} = \frac{\partial\mathscr{L}_{adv}}{\partial\hat{X}} + \frac{\partial\mathscr{L}_{adv}}{\partial\theta^*}\cdot\frac{\partial\theta^*}{\partial\hat{X}}. \quad (3)$$

.

Although Bi-Level Optimization has a wide application range, it requires two minimizations, so it is not practical to apply it to attacks on recommendation systems due to its high computational complexity. In the study of Jiaxi Tang et al. [3], the unroll and inner objective gradient of the Bi-Level Optimization step number were investigated to what extent the accuracy of the optimization was affected, and the instability of ItemAE was clarified. In the research of Jonathan Shlens et al[4], an adversarial example to make noise that gives the most influence on the machine learning system is made, and the robustness of the algorithm with nonlinearity is clarified. However, since various attacks can be considered practically, it is necessary to understand each attack characteristic first. In this study, we clarify the attack characteristics that affect the attack.

## 3. Proposed Analysis Method

In this paper, the surrogate models are WRMF and ItemAE, and the victim models are WRMF, ItemAE, and ItemCF. Then, the attack performance was compared in each combination of 6 types. In experiments 4.1 and 4.2, self-made datasets were prepared, and the number of users was 900, the number of items was 300, the number of fake users was 100, and the sparsity was 88%. In Experiment 4.3, we used Gowalla, a data set of location-based services. Gowalla had 13100 users, 14000 items, 131 fake users, and 99.7% sparsity.

### 3.1. Overview

ADAM was used for the optimization of inner objective, and SGD (Stochastic Gradient Descent) was used for the optimization of outer objective. The intersection loss function used here assumes that the number of users is $U$ and the number of items is $I$, and that $r_uk$ is the predicted value for the target item $i$ of the user $r$, defined as

$$\mathscr{L}_{adv} = -\sum_{u\varepsilon U}\log\frac{\exp r_{uk}}{\sum_{i\varepsilon I}\exp r_{ui}}. \quad (4)$$

### 3.2. Dataset

#### 3.2.1. Homegrown Dataset

The data set used was the one made by itself. The dataset had $U = 900$ normal users, $V = 100$ fake users, $I = 300$ items, 88% sparsity of the data, and $X\varepsilon\{0,1\}$. In the first data set, the threshold was set to 5, and the random numbers generated by the standard normal distribution were assumed to be 1 for those higher than the threshold and 0 for those lower than the threshold. The dimension d used in the WRMF was 20, the iteration of the outer objective was 100, and the iteration of the outer objective was 100. Also, the output $\hat{X}$ was output as a threshold value 0.2, 1 if the output value was 0.2 or more, and 0 otherwise.

#### 3.2.2. Gowalla Datasets

The data set provided by Gowalla, a location information service, was used in the last experiment. Gowalla is a data set that shows whether or not a user visited an arbitrary registration point from the location information of the user, and the value is two of visiting the place (1) and not visiting (0). The number of users was 13100, the number of items was 10,000, the number of fake users was 131, and the sparse rate was 99.7.

### 3.3. Evaluation Index

To evaluate the performance of the recommendation system, one test item was set for each user, and hit ratio at 10 (HR@10) was used to calculate the ratio of the predicted item to the actual one.

## 4. Experiments

In 4.1 and 4.2, the reproduction experiment of Jiaxi Tang et al. was expanded as a preliminary experiment [3]. In the reproduction experiment, the data set was made to be 3.2 self-made data set, and the recommendation algorithm was made to be WRMF, ItemAE, and the gradient calculation formula in the minimization of equation (3) was simplified by omitting the first number loop of approximation and optimization, and the effect was investigated. In the previous study, we examined the difference in the effect for each loop omission number, but in this paper, to verify the stability of the algorithm, HR@10 and loss function $\mathscr{L}_{adv}$ for each iteration were calculated without omitting the loop. Experiments 4.3 verified the attack performance against the false profile detection function. In Experiment 4.4 we measured HR@50 in 6 combinations of surrogate models (WRMF, ItemAE) and victim models (WRMF, ItemAE, ItemBase CF).


**Fig. 2.** Experimental results of WRMF
(left:Loss Function middle:HR@10 of target item
right:HR@10 of test item)

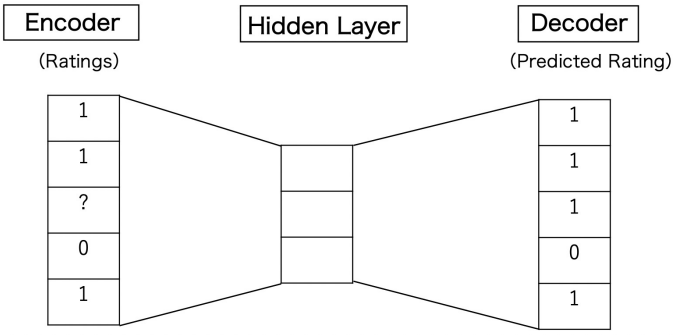


**Fig. 3.** Overview of ItemAE

### 4.1. WRMF

WRMF [5] is used to compress sparse matrices, such as in recommendation systems. Assuming that the latent variable of the user is $P \varepsilon R^{|U| \times |K|}$ and the latent variable of the item is $Q \varepsilon R^{|I| \times |K|}$, the predicted value of the normal user can be represented by $R = PQ^T$ for the normal data X. Also, when the fake data $\hat{X}$ is injected, the potential factor of the user is $F \varepsilon R^{|V| \times |K|}$ and the predicted value of the fake user is $R = FQ^T$. By using these, the training formula of the surrogate model is

$$\begin{aligned} &\mathscr{L}_{train}(X, R_\theta)\mathscr{L}_{train}(\hat{X}, \hat{R}_\theta) \\ &= \sum_{u,i} w_{ui}(x_{ui} - P_u^T Q_i)^2 \\ &+ \sum_{v,i} w_{vi}(\hat{x}_{vi} - F_v^T Q_i)^2 \\ &+ \lambda(||P||^2 + ||F||^2 + ||Q||^2). \quad \cdots \quad (5) \end{aligned}$$

.

The WRMF prediction is expressed by $R = PQ^T$, where $R$ is independent of the data $X, Y$ when the WRMF optimization method uses SGD - based algorithm. That is, $\mathscr{L}_adv$ in the first term of expression (3) cannot be differentiated by $\hat{X}$. $\frac{\partial \mathscr{L}_{adv}}{\partial \hat{X}} = 0$. Thus, equation (3) is expressed by

$$\nabla_{\hat{X}} \mathscr{L}_{adv} = \frac{\partial \mathscr{L}_{adv}}{\partial \theta^*} \cdot \frac{\partial \theta^*}{\partial \hat{X}} \cdot \cdots \quad (6)$$

.

Since the second term can be neglected, the optimization can be simplified. The results of the experiment are shown in **Figure 2**.

The left figure of **Figure 2** shows the loss function $\mathscr{L}_{adv}$ for each Iteration, and while Iteration succeeded in minimizing the loss function as a whole, the one which took the minimum value was the one whose Iteration was around 15, and the value rapidly increased when Iteration was 15 and 30. In the middle figure, HR@10 is high, but the maximum value is around 17, and the value is low between Iteration 15 and 30. Finally, in the test item, the value of HR@10 dropped sharply to 20, and then rose sharply. In this way, in ItemAE, the operation of the unstable value was done by Iteration. This is because ItemAE is a non-convex algorithm.

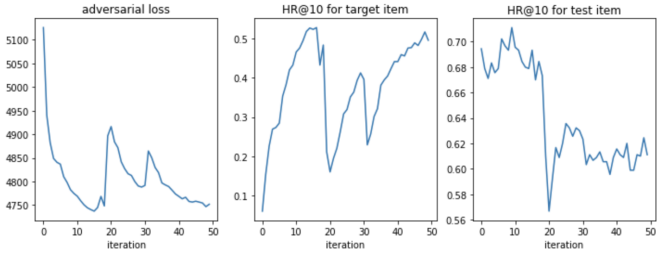


**Fig. 4.** Experimental results of ItemAE
(left:Loss Function middle:HR@10 of target item
right:HR@10 of test item)

### 4.2. ItemAE

ItemAE (Item AutoEncoder)[6] is a recommendation system of an auto-encoder.An overview is shown in **Figure 3**.

ItemAE is a system that inputs high - dimensional data to an encoder, makes it low - dimensional, and then is reconstructed by a decoder. The WRMF with SGD can ignore the first term of Eq.(3), but the second term cannot be ignored when ALS (alternating least square) is used without SGD because the loss function $\mathscr{L}_{adv}$ depends on the data X. On the other hand, ItemAE can ignore the second term. The equation for calculating the slope is

$$\nabla_{\hat{X}} \mathscr{L}_{adv} = \frac{\partial \mathscr{L}_{adv}}{\partial \hat{X}}. \quad \cdots \quad (7)$$

Since the second term can be neglected, the optimization can be simplified. The results of the experiment are shown in **Figure 3**.

The left figure of Figure.3 shows the loss function $\mathscr{L}_{adv}$ for each Iteration, and while Iteration succeeded in minimizing the loss function as a whole, the one which took the minimum value was the one whose Iteration was around 15, and the value rapidly increased when Iteration was 15 and 30. In the middle figure, HR@10 is high, but the maximum value is around 17, and the value is low between Iteration 15 and 30. Finally, in the test item, the value of HR@10 dropped sharply to 20, and then rose sharply. In this way, in ItemAE, the operation of the un-

| Algorithm | WRMF | ItemAE |
|---|---|---|
| Recall | 0.946 | 0.924 |

**Table 1.** Attack Performance on Homegrown Data

| Surrogate \ Victim | WRMF | ItemAE | ItemCF |
|---|---|---|---|
| WRMF | 0.139 | 0.119 | 0.238 |
| ItemAE | 0.114 | 0.118 | 0.115 |

**Table 2.** Attack Performance on Real Data

stable value was done by Iteration. This is probably because ItemAE is a non-convex algorithm[7].

### 4.3. Comparison of attack performance (Homegrown Dataset)

WRMF and ItemAE were used as attack generation algorithms, and the probability of detecting generated false profiles was analyzed. By this, it is possible to examine each attack's performance. The attack detection algorithm uses a neural network. The neural network used is a hidden layer of 2 layers, the first layer is Rectified Linear Unit of 100 units, and the second layer is a sigmoid unit. The input is an array of $|U| * |I|$ ratings, and the output is $X\varepsilon\{0,1\}$.The training data were 75%, the test data were 25%, and the number of epochs was 300, and the results were the average value of 5 times. The recall was used as an evaluation index. Recall shows the percentage of correct predictions that are correct. When Recall has a high value, false profiles are easily detected and attack performance is low. The results are shown in **Table 1**.

In this way, WRMF has a higher value than ItemAE, and the detection rate is higher. Therefore, in the detection system using a neural network, ItemAE can produce a fake profile with higher attack performance.

### 4.4. Comparison of attack performance (Real Dataset)

The surrogate models were WRMF and ItemAE, and the victim models were WRMF, ItemAE, and ItemBase CF. HR@50 was measured for each combination of 6 models. The data set used 3.2 geolocation service Gowalla data sets. ItemBaseCF [8] is an item-based cooperative filtering and a memory-based recommendation system. It is known as the most common recommendation system. A comparison of attack performance is shown in **Table 2**.

surrogate model was WRMF and the victim model was ItemCF. This is probably because ItemBase CF is a memory-based algorithm. Also, in general, collaborative filtering is vulnerable to data sparsity and the data set used in this study has a high sparsity of 99.7%, so the vulnerabilities may be more coordinated. The next higher value is considered to be due to the "white box setting"[9] in which both the surrogate model and the victim model are WRMF. In the white box set, the attack was made on the assumption that the attacker already knew the victim's recommendation model in advance, so the attack with higher attack performance was possible. Finally, there was no large fluctuation in the value of ItemAE even in the case where the above is desired. The non-linearity of ItemAE is shown to be flexible to attack because the value does not change even in the white box set. The same values for ItemBaseCF also indicate that the nonlinearity in the generation of fake data in neural networks is difficult to function against memory-based algorithms[10].

## 5. Conclusion

In this paper. Focusing on the injection attack which is an attack to the recommended system, the effect of optimization algorithm Bi-Level Optimization used in the injection attack on each recommended system was analyzed by calculating HR@50 for 2 kinds of proxy models and 3 kinds of victim models in all combinations of 6 types. The results show that memory-based cooperative filtering is highly vulnerable to attacks and ItemAE is more robust to attacks. In the future, the algorithm which discriminates the false review will be constructed based on this research result.

**References:**


[1] Burke Robin, Mobasher Bamshad, Williams Chad, and Bhaumik Runa. Classification features for attack detection in collaborative recommender systems. 2006.

[2] Colson Benoît, Marcotte Patrice, and Savard. Gilles. An overview of bilevel optimization. annals of operations research. 2006.

[3] Tang Jiaxi, Wen Hongyi, and Ke Wangr. Revisiting adversarially learned injection attacks against recommender systems. 2020.

[4] Goodfellowand Ian J., Shlens Jonathon, and Szegedy Christian. Explaining and harnessing adversarial examples. 2015.

[5] Gu Quanquan, Zhou Jie, and Ding Chris. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. 2010.

[6] ALhadlaw Omar and Kunna Arjun. Cs230: Deep learning based collaborative filtering. 2018.

[7] Smys S, Chen Joy, and Shakya Subarna. Survey on neural network architectures with deep learning. 2020.

[8] Su Xiaoyuan and Khoshgoftaar Taghi, M. A survey of collaborative filtering techniques. 2019.

[9] Prinz Katharina, Flexer Arthur, and Widmer Gerhard. On end-to-end white-box adversarial attacks in music information retrieval. 2021.

[10] O'MahonyMichael, Hurley Neil, J, Silvestre Guenole, C.M. Promoting recommendations:an attack on collaborative filtering. 2002.