Improving Resistance of Matrix Factorisation Recommenders To Data Poisoning Attacks

Sulthana Shams
School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
sshams@tcd.ie

Douglas J. Leith
School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
doug.leith@tcd.ie

***Abstract*—In this work, we conduct a systematic study on data poisoning attacks to Matrix Factorisation (MF) based Recommender Systems (RS) where a determined attacker injects fake users with false user-item feedback, with an objective to promote a target item by increasing its rating. We explore the capability of a MF based approach to reduce the impact of attack on targeted item in the system. We develop and evaluate multiple techniques to update the user and item feature matrices when incorporating new ratings. We also study the effectiveness of attack under increasing filler items and choice of target item.**

Our experimental results based on two real-world datasets show that the observations from the study could be used to design a more robust MF based RS.

***Index Terms*—recommender systems, matrix factorisation, data poisoning attacks, attack resistance**

I. Introduction

The issue of robustness against malicious attack is receiving attention from the research community [1, 2, 3]. Recommender System (RS) in social media platforms such as Facebook and Twitter have been in the limelight due to the risks they constantly pose to society by influencing their user base. From the point of view of RS, not only do they recommend items by learning a user's preferences but also help users discover and develop new interests thus influencing user behavior.

In poisoning attacks, an adversary creates fake profiles with carefully crafted ratings for items and attempts to target an item with the objective of increasing or decreasing the item's rating, thus making the item more/less likely to be recommended by the system. For our work, we consider that the attacker's goal is to promote a target item, i.e. an attacker-chosen target item's rating is increased and thus is more likely to be recommended to true users. We look at a common attack strategy called 'Average Attack' on collaborative filtering systems discussed in literature [1, 4, 5]. We assume that the attacker can only inject a limited number of fake users and each fake user rates a limited number of items (including the target item and other non-target items called filler items) to evade suspicion.

In this paper, we revisit Matrix Factorisation (MF) based RS [6]. MF is widely known in RS due to its simplicity and effectiveness. The typical paradigm of MF in RS is to decompose the user-item interaction matrix $R \in \mathbb{R}^{m \times n}$ into the product of two low-dimensional latent matrices $U \in \mathbb{R}^{d \times m}$ and $V \in \mathbb{V}^{d \times n}$ such that their dot product $U^T \cdot V$ is a good approximation of R. Matrix U captures the relationship between a user and the latent features while matrix V captures the relationship between an item and the features. We call U as the user-feature matrix and V as the item-feature matrix.

Typically, when new ratings are introduced to the system, the latent feature matrices are updated to incorporate the new ratings and thus update the user-item prediction matrix. Most works in literature take random items or unpopular items with fewer ratings as target items [3, 7, 8, 9]. We consider target items with different number of ratings received by true users and look at the shift in rating of the target item after updates to U, V. We conclude that some items are easier to attack than others. Items with fewer ratings are most vulnerable to attacks presumably due to the ease with which their feature vectors in V can be changed. An item which received a large number of ratings from the true users proves harder to attack.

Based on these observations, we further explore the role of U and V as a possible defense mechanism against fake user attacks. While one common approach is regular MF where both U and V are adapted, we look at other ways to boost the recommendation robustness under data poisoning attacks by looking at different ways of updating these latent feature matrices when introduced to new ratings. For example, consider the following new ways to incorporate the newly added ratings by fake users :

i) Hold V constant and update U just for the fake users. Here attack has no effect on true users. The U for the true users remains same as before the attack. Although this offers an immunity to attacks, it has no collaborative learning involved since U and V of true users remain unchanged to any incoming ratings.

ii) Hold V constant, add attackers and update U for all users.

iii) Thirdly, perform a modified alternating least squares method where find U using (i), then update V and repeat until converged.

From our study, we observe that ii) leads to very low change in rating of target item after attack. In comparison iii) shows larger change in rating of target item. So the effect of the

This work was supported by SFI grant 16/IA/4610

attack is pronounced when V is updated.

We show that these observations could be used to make updates to predicted ratings matrix more robust.

II. Related Work

The impact of data poisoning attacks where fake users are injected in RS with carefully crafted user-item interaction has been studied extensively. Detailed survey on attack models and robustness of RS algorithms are provided in [1, 2, 4, 5].

Recently, there is a line of work [7, 8, 9, 10] focusing on modelling the attack as an optimisation problem to decide the rating scores for the fake users and model attacks specific to the type of RS. For example, [8, 9] proposes data poisoning attacks for deep learning based RS and graph-based RS respectively. [7] proposes to select a subset of true users who are influential to the recommendations, to craft ratings for the fake user's attack on regular MF based RS.

In [10], instead of attacking the top-N recommendation lists, their goal was to study the change in the rating predictions after attack, for all missing entries of the rating matrix.

Most works in literature [7, 8, 9] use HR@N or 'Hit -Ratio' as the metric to study the effectiveness of attack where Hit-Ratio of a target item is the fraction of normal users whose top-N recommendation lists contain the target item.

We feel that the top-N recommendation list per user is too fragile a metric for observing attack effectiveness since the relevance of that list is user dependent. The standard prediction shift metric [1, 4, 11] used in literature also seem crude. It does not account for the initial rating of the target item before attack. i.e a target item with low initial rating would show larger deviation than an item with rating closer to mean value before attack. Keeping in mind all of the above, we introduce a new metric that gives the change in rating relative to the maximum deviation possible after attack. i.e. It depicts the ratio of the maximum deviation that the attack has achieved.

Although the impact of filler items in attack effectiveness in terms of Hit Ratio is studied in [8, 9], no relationship between the hit ratio and the number of filler items was concluded. The relationship was shown to be heavily dependent on the data-sets. Interestingly, our study on the same using the relative change in mean metric yielded a different result. Increasing filler items also increased the relative change in rating of the target item under attack.

While there are many studies exploring defensive techniques against data poisoning attacks [12, 13, 14], to the best of our knowledge, there is no existing studies that look at the factors affecting the defence capability of MF based RS.

III. Attack Model

For the type of attacks that we focus on, there is a *target item* that the attacker is interested in promoting and a set of *filler items* that is used to make the fake users seem real and ensure that some correlation is established with other true users.

A. Target Item

The target item is given the maximum rating to promote it in the system. We consider three types of target items based on the number of ratings received from the true users. Specifically, in our experiments, we sample an item uniformly at random from those items which have received 1, 10, 100 ratings and treat it as the target item.

B. Attack Knowledge

We assume that the adversary knows the mean rating and standard deviation for every item in the system. This is a reasonable assumption since such aggregate information about user preferences may be found online from databases which publicly displays the average user ratings of items. (e.g. movie databases, amazon product databases etc)

The filler items are chosen randomly from the list of items. Intuitively it will be much more difficult to detect such a fake user profile since the set of rated items change from profile to profile. The ratings for the filler items are sampled from the Gaussian distribution using the mean rating and standard deviation of every item available with the adversary.

IV. Experiments

A. Datasets

We evaluate the effectiveness of attack on the MovieLens dataset (943 users rating 1682 movies, contains 100000 ratings from 1-5) which is widely used in literature for evaluating recommender systems under attack and Goodreads 10K dataset (53,424 people rating 10,000 books, 5.9M ratings from 1-5).

We take a dense subset of the Goodreads dataset, obtained by selecting the top 1000 users which have provided the most ratings. This provides us with 1000 users and 8557 items rated by these top 1000 users.

B. Evaluation Setup

Unless mentioned otherwise, the attack size is fixed to 1% of the total true user population. We also look at how the number of filler items and the number of ratings of targeted item by true users impact attack effectiveness. We sample 50 instances of target items under each set-up and will average their experimental results.

C. Performance Metrics

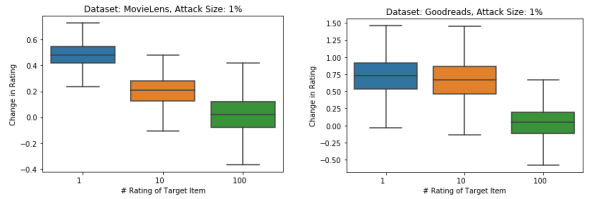
We use change in rating of target item relative to the maximum deviation possible as our evaluation metric.

$$\text{Change in Rating}_u = \frac{\mu_f(u,i) - \mu_o(u,i)}{|5 - \mu_o(u,i)|}$$

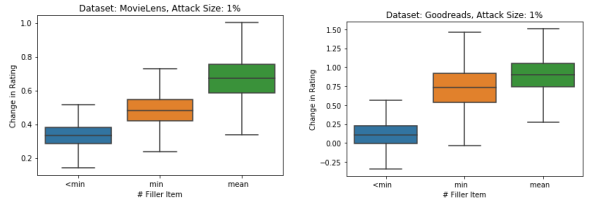
where $\mu_f(u,i)$ is the predicted rating of target item i of user u after attack, $\mu_o(u,i)$ is rating of the same target item i of user u before attack and 5 is the maximum rating that can be given to target item.

V. Results Analysis

Let us first consider the usual MF where we adapt both U and V and look at how number of attacker filler items and ratings of target item by true users impact the attack.

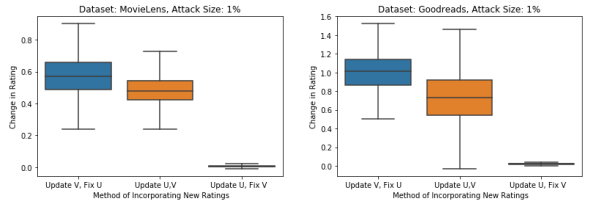


(a) (b)



(c) (d)

Fig. 1: Box-plot comparing the distribution of change in rating over true users for number of ratings of target item and different number of filler items respectively for MovieLens and Goodreads dataset when updating both U and V



(a) (b)

Fig. 2: Boxplot comparing the distribution of change in rating over true users for different methods of adapting new ratings using U, V for MovieLens and Goodreads dataset.

1) Impact of the number of ratings of target item : Figure 1 (a),(b) have box-plots showing distribution of change in rating when targeting items with different number of ratings. For this study, we fix the number of filler items to the minimum number of items rated by true users in the respective data-set.

It can be seen from both Figure 1(a) and Figure 1(b) that some items are easier to attack than others. From table I, an item with just 1 rating shows the most mean shift in rating after attack compared to an item with 100 ratings when both U, V are updated. In fact, target item with 100 ratings proves much harder to attack with a mean change in rating close to 0 for both data-sets.

Targeting items with higher number of ratings seems to make it difficult for the attackers to change the feature vector. In comparison, an item with fewer ratings is fragile and an attack to such an item would be very difficult to defend. It becomes extremely easy for the attackers to change the feature vector of such an item.

2) Impact of the number of filler items: Figure 1 (c),(d) show the impact of the number of filler items on our attacks for target item.

	# target ratings		# filler items	
Dataset/Update Method	1	100	<min	=mean
ML/ Update U,V	0.48	-0.04	0.33	0.67
ML/ Fix V, Update U	0.005	0.04	0.005	0.005
GR/ Update U,V	0.73	-0.005	0.10	0.89
GR/ Fix V, Update U	0.02	0.1	0.02	0.02

TABLE I: Mean change in rating for MovieLens and Goodreads datasets for number of ratings of target item =1,100 and number of filler items=minimum, mean number of items rated by true users. Legend: ML=MovieLens dataset, GR=Goodreads dataset

For this study we fix the item with 1 rating as the target item for both MovieLens and Goodreads data-set since it is the easiest to attack.

For both data-set, we observe the distribution of change in rating against number of filler items as 1) less than the minimum number of items rated by true users in the data-set 2) equal to minimum number of items rated by true users in the data-set 3) equal to the mean number of items rated by true users in the data-set.

It can be seen from both Figures 1 (c) and (d) and table I that increasing filler items from less than minimum to mean increases the mean change in rating of target item when both U, V are updated. It seems that changes to V are more pronounced when filler items increase. Perhaps because such a fake user would be similar to more true users and thus contributes more to change in V.

Although an attacker would achieve increased change in rating of target item when using more filler items, rating items more than the mean number of items rated by true users may be flagged as a suspicious behaviour.

For the rest of the experiment, we fix the number of filler items to the minimum number of items rated by true users in the data-set and choose a target item with one rating to better capture the effects of attack for the next part of the experiment.

A. Incorporating New Ratings

Figure 2 (a),(b) compares the distribution of change in rating over true users in the data-set when applying different ways to adapt U and V vectors to incorporate new ratings into the system. As discussed previously, for all the scenarios below we fix the number of filler items to the minimum number of items rated by true users in the data-set and choose a target item with one rating to show our results.

1) Update V and fix U: In this set-up, we first update U only for attackers by holding V constant. We obtain an updated U for fake users but with values for true users same as before. Then proceed to update V.

Here, change in rating observed after attack is slightly higher for both the data-sets in comparison to regular MF. The effect of number of ratings of target item and number of filler items are similar to Figures 1 (a),(b) and so are not reported separately. Just as in regular MF, a fragile item with one rating can be easily attacked while a well-reviewed item proves harder to attack.

2) Update U and fix V: We hold V constant, add attackers and update U for all users. From Figure 2, attack has only a small effect on ratings for true users in this scenario. Updated U is very close to its original version before attack assuming that the regularisation penalty is not too high. This means the change in rating on target item after attack is very low.

The effect of number of filler items and ratings of target items is reported in table I. The increasing number of filler items seems to have no effect on this set-up. Also, the mean change in rating for any choice of target item are found to be negligible compared to regular MF. We believe that the small increase in change in rating as we move from target item with 1 rating to 100 ratings for this set-up comes from the regularisation penalty applied.

We conclude that the effect of the attack on the target item is captured by V matrix. As long as V is kept constant, updated U after attack is very similar to the one before attack unless regularisation parameter is too high. This ensures that the predicted ratings matrix after attack is very close to the predicted matrix before attack. So effect of attack on true users is found negligible.

Thus using method 2) for incremental updates to U when new ratings are added, then periodically using regular MF to update U and V would help in monitoring attacks. If a big difference between their results is observed, then that might flag a warning for items that change rating a lot.

B. Conclusions

In this paper, we revisited the MF approach to RS and studied the effect of attack under different update methods of latent matrices when incorporating new ratings. We also studied the effectiveness of attack under increasing filler items and choice of target item.

We can use these observations to make updates to RS more robust. Items that are more vulnerable to attacks can perhaps be defended from fake users by using dummy ratings which would make it harder for injected fake users to change their feature vector. Also updates to latent feature matrices need not be performed frequently together. Instead, regular MF methodology could be used periodically with Approach 2 for incremental updates to the ratings matrix. Thus any large shift in rating of items could be monitored periodically and necessary actions taken.

Our approaches are simple, yet effective and can be easily used in existing systems.

References

[1] S. K. Lam and J. Riedl, "Shilling recommender systems for fun and profit," in *Proceedings of the 13th International Conference on World Wide Web*, ser. WWW '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 393–402. [Online]. Available: https://doi-org.elib.tcd.ie/10.1145/988672.988726

[2] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems," *ACM Transactions on Internet Technology*, vol. 7, pp. 23–es, 10 2007.

[3] C. Wu, D. Lian, Y. Ge, Z. Zhu, E. Chen, and S. Yuan, *Fight Fire with Fire: Towards Robust Recommender Systems via Adversarial Poisoning Training*. New York, NY, USA: Association for Computing Machinery, 2021, p. 1074–1083. [Online]. Available: https://doi-org.elib.tcd.ie/10.1145/3404835.3462914

[4] K. Patel, AmitThakkar, C. Shah, and K. Makvana, "A state of art survey on shilling attack in collaborative filtering based recommendation system," 11 2015.

[5] S. Mingdan and Q. Li, "Shilling attacks against collaborative recommender systems: a review," *Artificial Intelligence Review*, vol. 53, 01 2020.

[6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[7] M. Fang, N. Z. Gong, and J. Liu, *Influence Function Based Data Poisoning Attacks to Top-N Recommender Systems*. New York, NY, USA: Association for Computing Machinery, 2020, p. 3019–3025. [Online]. Available: https://doi-org.elib.tcd.ie/10.1145/3366423.3380072

[8] H. Huang, J. Mu, N. Z. Gong, Q. Li, B. Liu, and M. Xu, "Data poisoning attacks to deep learning based recommender systems," *Proceedings 2021 Network and Distributed System Security Symposium*, 2021. [Online]. Available: http://dx.doi.org/10.14722/ndss.2021.24525

[9] M. Fang, G. Yang, N. Z. Gong, and J. Liu, "Poisoning attacks to graph-based recommender systems," *Proceedings of the 34th Annual Computer Security Applications Conference*, Dec 2018. [Online]. Available: http://dx.doi.org/10.1145/3274694.3274706

[10] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 1893–1901.

[11] B. Mobasher, R. Burke, R. Bhaumik, and J. Sandvig, "Attacks and remedies in collaborative recommendation," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 56–63, 2007.

[12] C. Williams, B. Mobasher, and R. Burke, "Defending recommender systems: Detection of profile injection attacks," *Service Oriented Computing and Applications*, vol. 1, pp. 157–170, 10 2007.

[13] P.-A. Chirita, W. Nejdl, and C. Zamfir, "Preventing shilling attacks in online recommender systems," in *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, ser. WIDM '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 67–74. [Online]. Available: https://doi-org.elib.tcd.ie/10.1145/1097047.1097061

[14] R. Bhaumik, C. Williams, B. Mobasher, and R. Burke, "Securing collaborative filtering against malicious attacks through anomaly detection," 01 2006.