Model Inversion Attack on Item-to-Item Collaborative Filtering

Introduction:

Recommendation Systems are ever-pervading in an online environment. Ranging from the websites you browse to watch your favorite TV shows, to social media like Facebook, YouTube and online shopping sites like Amazon and eBay, you will always find a recommendation engine in play. The recommendations generated by these systems are not random either. They are well thought-out projections which are designed to maximize the profits of the advertiser while also incepting your subconscious with their marketing. So how do these recommendation systems go about doing something so incredible? In this report, we talk about how they leverage a user's history to make optimal recommendations and prove that if any malicious actor gets their hands on a user's recommendations, it is only a matter of time before they can generate that user's purchase history.

Scenario:

Almost every website one visits has advertisements to show to their audience. Most of the times, it is a harmless process through which the said website generates revenue. But in some cases, the act might not be just as plain. It is completely possible for a web developer to incorporate logic into their JavaScript code such that they can track and save advertisements being displayed to a particular user. Now suppose in some cases, these ads are that of an e-shopping website like Amazon. With the armed knowledge of the target user's Amazon recommendations, using our techniques, the malicious actor can try to generate the target's purchase history.

Related Work

In "Item-Based Collaborative Filtering Recommendation Algorithms", Badrul and John clearly explain the principle behind the said recommendation algorithm. When the data matrix of users and items is available at our disposal, we can learn the similarity between items by fundamentally using the following principle – items rated similarly by multiple users are similar to each other. This algorithm overcomes the scalability problem of User-Based collaborative filtering as the number of users for a website is enumerable but the number of items is relatively less.

Yehuda and Chris explain in their paper "Matrix Factorization Techniques for Recommender Systems", the intricate art of decomposing the user-item data matrix into 2 low rank matrices – User matrix and Item matrix. By defining terms like "user-bias", "item-bias", "regularization factor" and creating an objective function to minimize prediction-loss, we can use the dot product of the low-rank matrices to fill-in missing values in the original user-item matrix.

Data Exploration:

For our experiments, we use the Amazon Movies_and_TV_Shows Data which has a record of 8,765,912 unique (user, item) purchases along with their corresponding rating metric.

The Number of Unique Users: 3826085
The Number of Unique Items: 182032

Num. Rows:	8,765,567
Num. Unique:	3,826,085
Missing:	0

Frequent Items

AV6QDP8Q0ONK4	4254
A1GGOC9PVDXW7Z	2292
A328S9RN3U5M68	2175
ABO2ZI2Y5DQ9T	2136
AWG2O9C42XW5G	2046

Num. Rows:	8,765,567
Num. Unique:	182,032
Missing:	0

Frequent Items

B00YSG2ZPA	24558
B00006CXSS	24489
B000WGWQG8	23584
B00AQVMZKQ	21015
B01BHTSIOC	20889

Using these 8,765,567 unique entries of (User, Item, Rating) data points, we create an item-item collaborative filtering recommendation system which is identical to the one being used by Amazon.

Attack Description:

We use the original recommender system to get ‘n’ number of recommendations for a target user. These are the recommendations which the attacker supposedly has in their knowledge base. With the knowledge of these recommendations and a certain amount of partial history of the target user, the attacker creates an attack model using which they try to predict the user’s original purchase history.

We use 2 different types of attack models in this report –

1) Item-Item Collaborative Filtering
2) Ranking Factorization Machine

Algorithm:

1) Use the original recommendation system to make 'n' recommendations for the target user.
2) Add these recommendations along with the target's partial history (or) no history to create an attack model based on item-item collaborative filtering or ranking factorization machine.
3) Use the attack model to predict the remaining purchase history of the user.
4) Report metrics such as Precision, Recall and NDCG.

Metrics:

Precision: If 'pk' is a vector of the k elements recommended to a user and 'a' is the set of items in original purchase history for that user. The "precision at k" for this user is defined as

$$P(k)=|a \cap pk| / k$$

Recall: $R(k)=|a \cap pk| / |a|$

Normalized Discounted Cumilative Gain (NDCG):

NDCG is a measure of ranking quality of a recommender. An ideal recommender would recommend most **relevant** items to a user at the top(with highest rank). Relevance of an item is the importance it holds to the user – in terms of howlikely they are to purchase it, etc. In our attack model, as we already have the ratings of user's original purchase history, we treat the original rating of an item as its relevance.

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

If an item with high relevance is recommended last, we penalize the recommender with respect to its index in the recommendation vector.

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$$

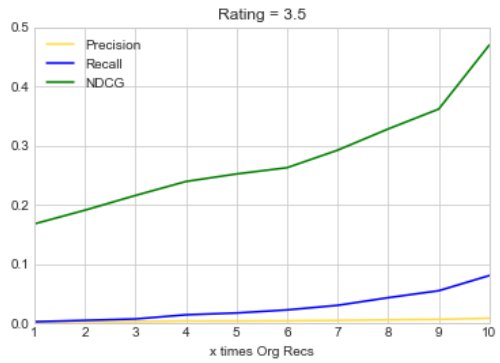
Every vector of recommendations is normalized by a Normalization Factor which is the ideal-NDCG(denominator). We push items with highest relevance at the top of the recommendations(ideal case) and calculate the DCG of this ideal-recommendation to get IDCG.

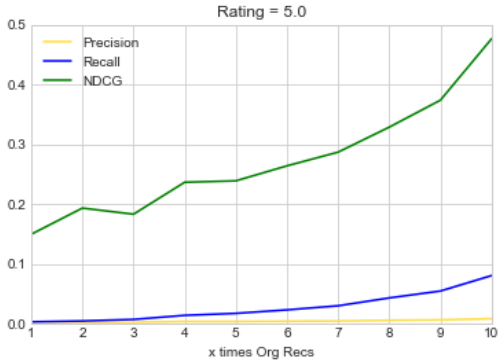
Experiments:

Exp 1:

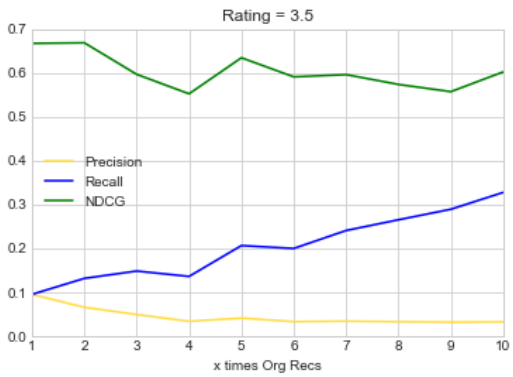
As it can be seen in the Data Exploration section, User- AV6QDP8Q0ONK4 is the most active user with 4254 items in their purchase history. To check the effectiveness of the algorithm, we remove their entire purchase history from the attacker's knowledge base and see how many recommendations of this user should they observe to regain as much original history of the user as possible.

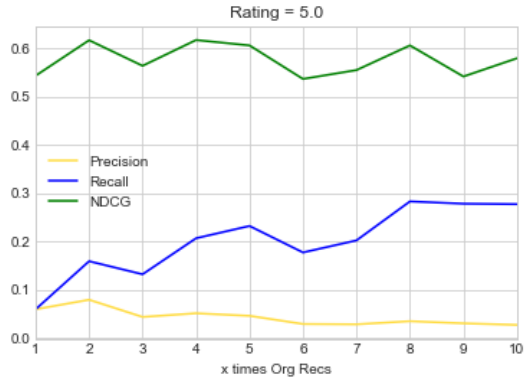
The following graphs show the performance of item-item collaborative filtering. The graph on left is when recommended items are added with user's presumed rating as 3.5 and the graph on the right has the presumed rating as 5.0.





When we put a Ranking Factorization Machine to the same task, we find the following results –





As it can be seen, a Ranking Factorization Machine can retrieve the user's purchase history with more efficiency and having as many user recommendations as possible is helpful.

Exp 2:

For this round of experiments, we chose 10 users whose purchase histories are in the range of 500-600. By altering 2 factors –

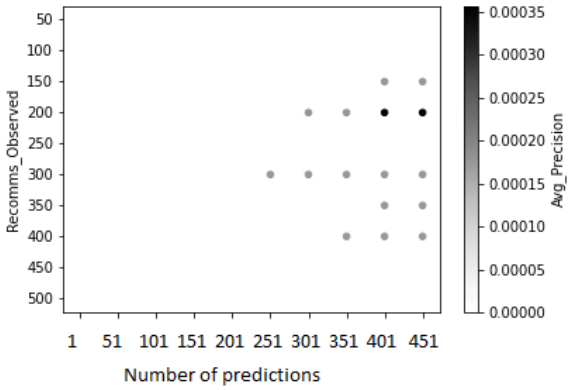
1) Number of recommendations of the users we observed and
2) Number of items we are trying to predict to be part of the users' history,

we conducted several experiments to get the following results.

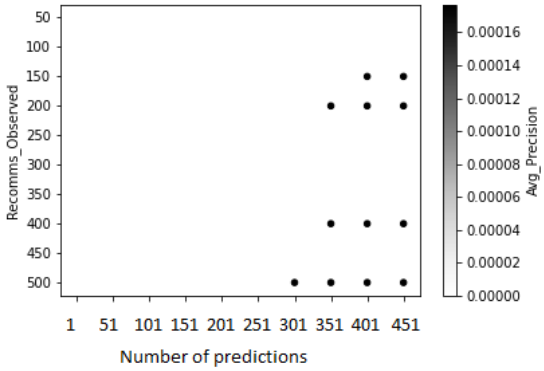
a) When no history is added to attacker's knowledge base

Item-Item Collaborative Filtering

Item Rating – 3.5



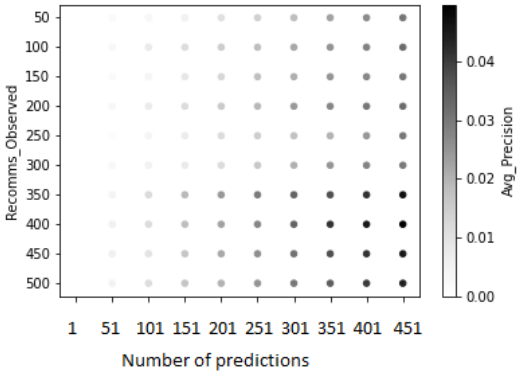
Item Rating – 5.0



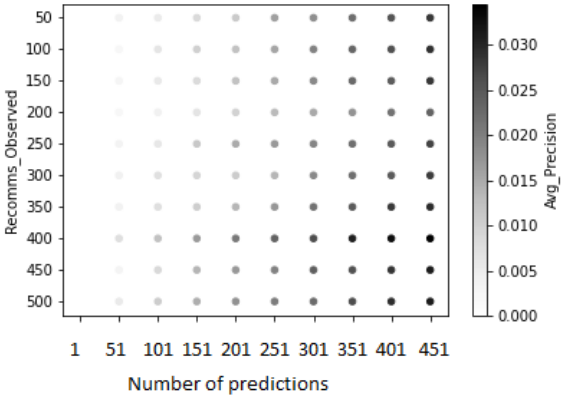
Precision is calculated for every user per (Obserbed_Recomms, Predictions) and then we plot the average_precision.

Ranking Factorization Machine:

Item Rating – 3.5



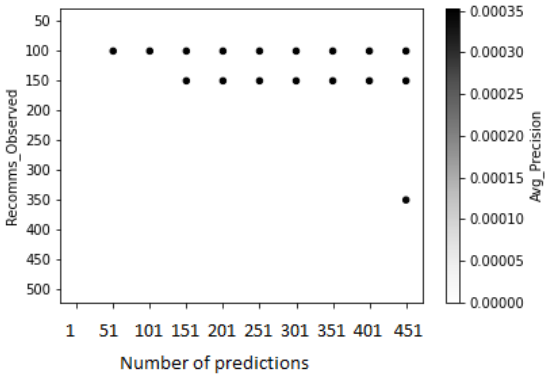
Item Rating – 5.0



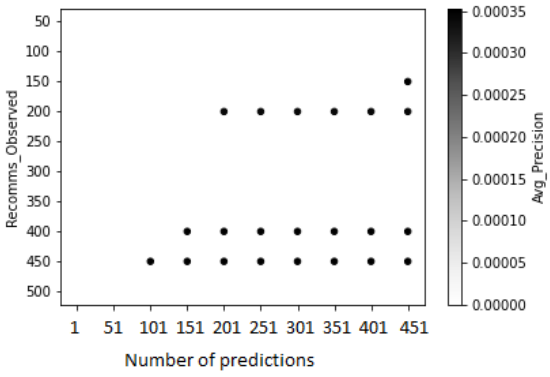
b) When 50% of the users' history is known by the attacker.

Item-Item Collaborative Filtering

Item Rating – 3.5

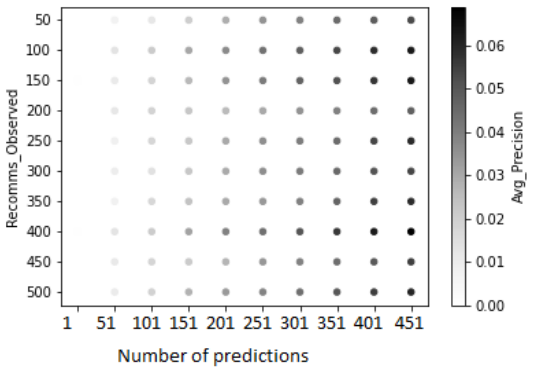


Item Rating – 5.0

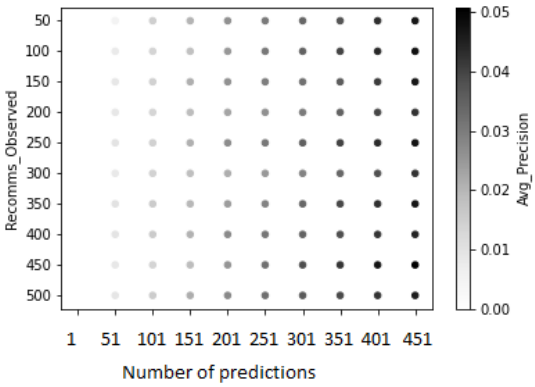


Ranking Factorization Machine

Item Rating – 3.5



Item Rating – 5.0



Conclusion:

In the above experiments we realize it is possible(albeit to a small degree) to retrieve a user's purchase history when nothing more than

their recommendations are known. We make it possible by exploiting the nature of recommendation engines which strive to learn the correlations and causations between different items in the website catalogue. We also observe that Matrix Factorization is more effective than Item-Item Collaborative Filtering to effectively retrieve a user's purchase history.

References:

- Sarwar, Badrul, and John Riedl. "Item-Based Collaborative Filtering Recommendation Algorithms." *Item-Based Collaborative Filtering Recommendation Algorithms*, 2001, files.grouplens.org/papers/www10_sarwar.pdf.
- Koren, Yehuda, and Chris Volisnky. "Matrix Factorization Techniques for Recommender Systems." *Matrix Factorization Techniques for Recommender Systems - IEEE Journals & Magazine*, 2009, ieeexplore.ieee.org/document/5197422.