The Computational Overhead of Differentially Private Collaborative Filtering Systems

Ryan Leigh

Arizona State University, rleigh2@asu.edu

1 INTRODUCTION

Collaborative filtering is a popular technique to provide strong recommendations on many topics. An unfortunate byproduct of this technique is the real threat to the privacy and security of any user on a particular datasheet. One of the many privacy techniques, known as differential privacy, is employed in circumstances to try to alleviate the concerns around a recommendation system like collaborative filtering. Unfortunately, there is an inherent cost to differential privacy, the topic of accuracy vs privacy. This research paper will attempt to not only evaluate this theme of accuracy vs privacy but also see if there is any additional computational overhead with the addition of differential privacy on a collaborative filtering system.

Matrix Factorization is a well-known approach to these types of recommendation systems. Suggested by He, Xiangnan, et al, matrix factorization, and its limit of the inner product which simply combines the multiplication of latent features linearly, may not capture the structure of user interaction data (He, Liao, et al.). The approach used in this paper will be to investigate the use of low-rank matrix factorization against the use of a deep neural network focusing on implicit feedback but still capturing the essence of matrix factorization with its user-item interaction patterns. The neural collaborative methodology replaces the inner product from matrix factorization with a neural architecture. Furthermore, this is leveraged through a multi-layer perceptron to enable the neural collaborative filtering model to explore high levels of non-linearities. The results from this are compared to the results low-rank matrix factorization and how those results are manipulated by the same differential privacy settings on similar dataset as a control.

2 THREAT MODEL

In order to justify the usage of differential privacy, there needs to be a threat to users whose data is trained upon by such a model. One of the main threats to a recommender system is a membership inference attack. Despite the use case of this paper not containing any sensitive data, the fact remains true that if there were sensitive data given the model was victim to such an attack, it would be of significant trouble to the developers of the recommender system. The attack demonstrated by Zhang, Minxing, et al uses a user-level attack where there is no access to the data but there is access to the model. An adversary would only observe the ordered recommended items from the recommender system instead of predicting results in posterior probabilities. This will utilize a shadow model to derive training data from the target model and use to train the attack model. Instead of the proposed defense mechanism from Zhang, Minxing, et al, this paper will propose that simply the addition of noise to the training data through differential privacy will be ample to prevent these kinds of attacks.

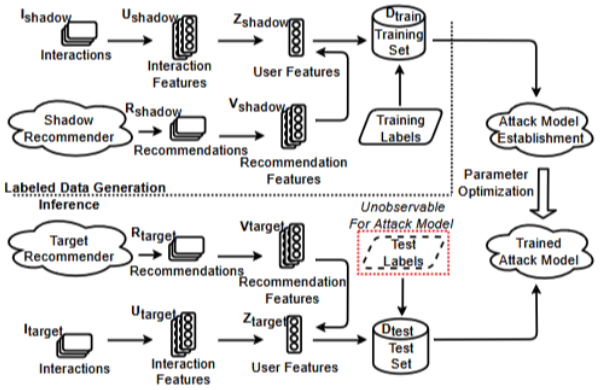

Figure 2: Attack Model Framework (Zhang, Minxing, et al)

3 EXPERIMENT

NEURAL COLLABORATIVE FILTERING

With more specificity to the experiment, the neural collaborative filtering model is to recommend movies based on user preferences. This experiment used the Movie Lens dataset, commonly tested for recommendation systems. The full dataset has a list of movies, users, and ratings by such users on such movies. All rating data is normalized to a range of 0 to 1 to facilitate training and ensure consistency in the input scale. Also utilized is negative sampling to generate implicit feedback for unobserved user-item interactions, assigning a rating of 0 to such samples. This allowed the model to learn from both positive and negative signals.

The NCF model architecture consisted of multiple components. The embedding layers were used to represent users and movies as dense vectors in a latent space. These embeddings capture meaningful relationships, with each user and movie represented by 64-dimensional vectors. The embeddings for users and movies were then combined using an element-wise multiplication operation to represent the interaction between the two. This interaction was fed into a multi-layer perceptron with three layers each followed by the ReLU activation function.

The final layer of the MLP outputs a single value representing the likelihood of interaction between a user and a movie. The output layer used the sigmoid activation function to produce a probability score between 0 and 1. The loss function binary cross-entropy loss or log loss and Adam were then employed to update gradients. During training, a mini-batch gradient descent approach with a batch size of 1024 is utilized. Each model, either with differential privacy or without, is trained to relative convergence. The dataset was split into training and testing sets, ensuring that interactions in the test set did not overlap with those in the training set.

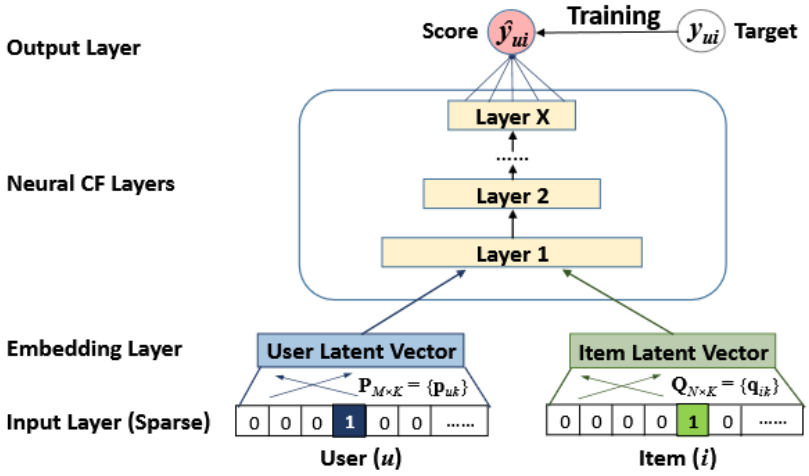

Figure 2: Neural Collaborative Filtering Architecture (He, Xiangnan, et al)

LOW-RANK MATRIX FACTORIZATION

Low-rank matrix factorization is a technique used in collaborative filtering to predict missing entries in a user-item interaction matrix. It works by decomposing the matrix into two smaller matrices: user preferences and item attributes, such that their product approximates the original matrix. The decomposition assumes that the user-item matrix is of low-rank, meaning user preferences and item characteristics can be represented in a smaller latent space.

This version of the matrix factorization is powered by singular value decomposition shown in figure 3. This will decompose a matrix and find the best lower rank shown in the figure. Effectively, a mathematical approximation. To build the recommendation system, the ratings data is reshaped into a user-item matrix, where each row represents a user, each column represents a movie, and the entries represent ratings. The lower rank ensures that individual user biases do not influence the matrix factorization disproportionately.

$$R = U\Sigma V^T$$

Figure 3: Definition of SVD (Becker)

Using only the top 50 singular values, the system reconstructs the predicted user-item matrix. These predictions estimate how much a user might rate a movie that they haven't yet rated. Once the predicted ratings are computed, the system ranks the movies based on these predictions, allowing users to receive personalized recommendations. The system then filters out movies that a user has already rated and returns the top ten as recommendations.

4 RESULTS

Low-Rank Matrix Factorization

First things first, we explore the results of the low-rank matrix factorization approach. Without any differential privacy applied to the dataset, the results were particularly good. This is the best result implemented in this paper. In figure 4, one can see the ten movies recommended to the user that was chosen. One can see that these movies have a lot of similarities being within the action, comedy genre space. Running this model took only 19 seconds.

```
Predictions:
1634  Good Will Hunting (1997)                          Drama
447   Fugitive, The (1993)                              Action|Thriller
106   Braveheart (1995)                                 Action|Drama|War
1060  Reservoir Dogs (1992)                             Crime|Thriller
2302  Shakespeare in Love (1998)                        Comedy|Romance
1318  Sling Blade (1996)                                Drama|Thriller
2502  Election (1999)                                   Comedy
34    Dead Man Walking (1995)                           Drama
1163  Star Wars: Episode V - The Empire Strikes Back (1980)   Action|Adventure|Drama|Sci-Fi|War
30    Twelve Monkeys (1995)                             Drama|Sci-Fi
```

Figure 4: Results of Low-Rank Matrix Factorization without DP

```
Predictions:
1634  Good Will Hunting (1997)                          Drama
447   Fugitive, The (1993)                              Action|Thriller
1318  Sling Blade (1996)                                Drama|Thriller
2302  Shakespeare in Love (1998)                        Comedy|Romance
1060  Reservoir Dogs (1992)                             Crime|Thriller
3041  Green Mile, The (1999)                            Drama|Thriller
2230  Life Is Beautiful (La Vita è bella) (1997)        Comedy|Drama
2807  Boys Don't Cry (1999)                             Drama
2502  Election (1999)                                   Comedy
34    Dead Man Walking (1995)                           Drama
```

Figure 5: Results of Low-Rank Matrix Factorization with DP (epsilon=3.0)

```
Predictions:
1634  Good Will Hunting (1997)                          Drama
447   Fugitive, The (1993)                              Action|Thriller
1060  Reservoir Dogs (1992)                             Crime|Thriller
106   Braveheart (1995)                                 Action|Drama|War
2302  Shakespeare in Love (1998)                        Comedy|Romance
2230  Life Is Beautiful (La Vita è bella) (1997)        Comedy|Drama
1318  Sling Blade (1996)                                Drama|Thriller
34    Dead Man Walking (1995)                           Drama
2502  Election (1999)                                   Comedy
253   Star Wars: Episode IV - A New Hope (1977)         Action|Adventure|Fantasy|Sci-Fi
```

Figure 6: Results of Low-Rank Matrix Factorization with DP (epsilon=5.0)

Neural Collaborative Filtering (NCF)

The next step is to observe the results of the NCF models. First off, we see the results of the NCF without any differential privacy applied. This ran very well, though I needed to run it on a smaller dataset to get results in a reasonable amount of time. This first test was completed in 4.71 minutes. A quick time only running about 20 epochs before reaching a reasonable convergence given the computational power I have in my possession.

```
Recommended Movies:
13    Nixon (1995)                                        Drama
37    It Takes Two (1995)                                 Children|Comedy
47    Pocahontas (1995)                                   Animation|Children|Drama|Musical|Romance
90    Mary Reilly (1996)                                  Drama|Horror|Thriller
301   Ready to Wear (Pret-A-Porter) (1994)                Comedy
317   Strawberry and Chocolate (Fresa y chocolate) (1993) Drama
401   Federal Hill (1994)                                 Drama
514   RoboCop 3 (1993)                                    Action|Crime|Drama|Sci-Fi|Thriller
1211  Manhattan (1979)                                    Comedy|Drama|Romance
1525  Wild America (1997)                                 Adventure|Children
```

Figure 7: NCF without DP

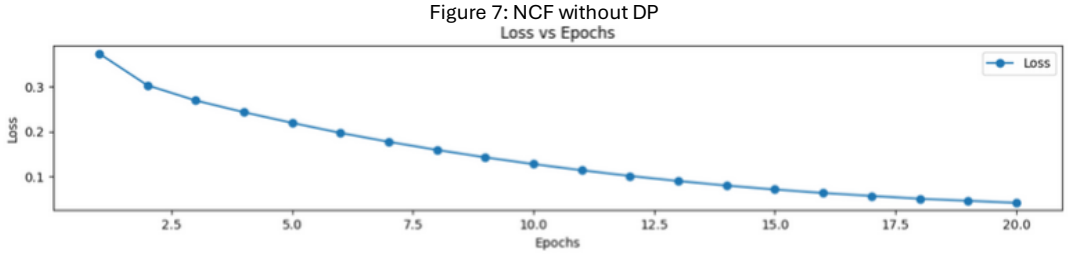

Figure 7: NCF epoch loss without DP

The next test was with differential privacy enabled with an epsilon of 3.0. This did not perform well at all, and it took 17.33 minutes. The next test took the same amount of time with a higher epsilon yielding slightly better values.

```
Recommended Movies:
18    Ace Ventura: When Nature Calls (1995)               Comedy
37    It Takes Two (1995)                                 Children|Comedy
123   Flirting With Disaster (1996)                       Comedy
350   Flintstones, The (1994)                             Children|Comedy|Fantasy
457   Good Man in Africa, A (1994)                        Action|Adventure
476   Kalifornia (1993)                                   Drama|Thriller
658   World of Apu, The (Apur Sansar) (1959)              Drama
681   Butterfly Kiss (1995)                               Drama|Thriller
809   Convent, The (O Convento) (1995)                    Drama
1604  Hugo Pool (1997)                                    Romance
```

Figure 8: NCF Results with DP (epsilon = 3.0)

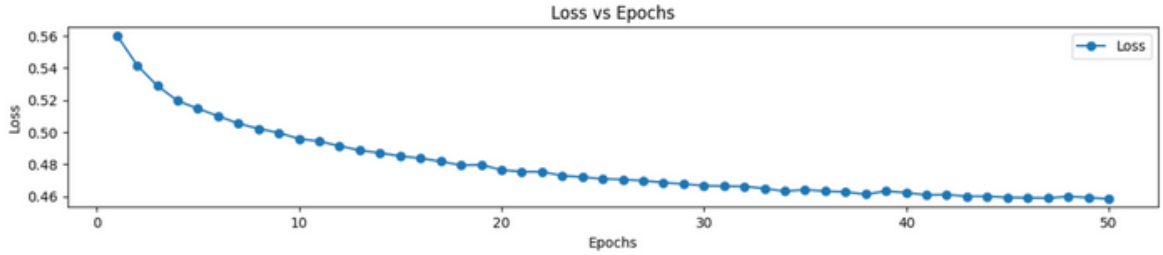

Figure 9: NCF epoch loss with DP (epsilon = 3.0)

```
Recommended Movies:
37    It Takes Two (1995)                                 Children|Comedy
91    Vampire in Brooklyn (1995)                          Comedy|Horror|Romance
133   Down Periscope (1996)                               Comedy
600   Fargo (1996)                                        Comedy|Crime|Drama|Thriller
1111  Everything Relative (1996)                          Drama
1427  Rhyme & Reason (1997)                               Documentary
1465  Children of the Revolution (1996)                   Comedy
2331  Karate Kid, Part III, The (1989)                    Action|Adventure|Children|Drama
4923  What Time Is It There? (Ni neibian jidian) (2001)   Drama
5329  Horse's Mouth, The (1958)                           Comedy
```

Figure 10: NCF Results with DP (epsilon = 5.0)

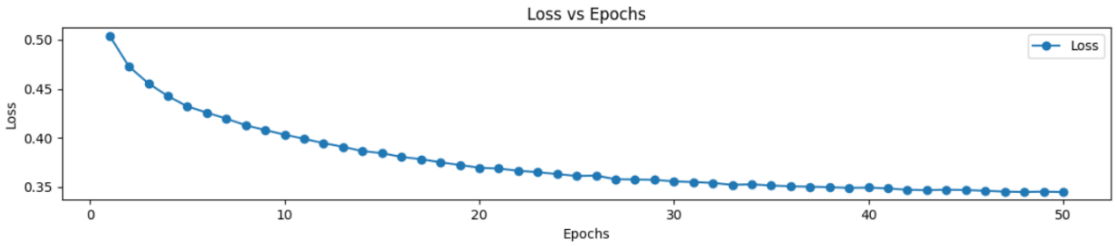

Figure 11: NCF epoch loss with DP (epsilon = 5.0)

5 CONCLUSIONS

The results given show the initial hypothesis was correct. The implementation of differential privacy cost the neural network a significant delta in accuracy and increased the time by about 3.67 times. Though, when using a non-neural recommendation system, the implementation of the same varying degrees of differential privacy did not accost the results as much accuracy. Going even further, this program ran within seconds, completely obliterating the NCF in terms of training time. It is important to remember that it is difficult to evaluate the closeness of a movie numerically as these movies are subjective in their quality. Though, as many researchers have proven before, there can be a recommendation system that is successful in recommending movies. I would further advocate that the neural network did need some preprocessing or pretraining to assist with training. Though, I concede that this is speculation and not necessarily grounded in truth. Furthermore, the ability to utilize more data and train longer would lead to better results rather than relying on smaller datasets that my existing hardware can control. My belief is that there should be some improvements to the logic or implementation of the program before it is reasonable to conclude that my existing hardware is an issue.

Recommendation systems are one of the harder ML problems out there due to an array of factors. I put in my best effort but my lack of experience with this sort of thing really shows. The conclusions are evident, there is a *significant* increase in computational overhead. If someone were training a recommendation system, differential privacy is not the best of methods. From Zhang, Minxing, et al, a popularity randomization defense would work to solve the issue of a membership inference attack. According to my results, the only benefit of differential privacy is that it makes the recommendations unusable.

REFERENCES

Becker, Nick. "Matrix Factorization for Movie Recommendations in Python." *Nick Becker*, 10 Nov. 2016, beckernick.github.io/matrix-factorization-recommender/.

He, Xiangnan, et al. "Neural Collaborative Filtering." *arXiv.Org*, 26 Aug. 2017, doi.org/10.48550/arXiv.1708.05031.

Zhang, Minxing, et al. "Membership Inference Attacks against Recommender Systems." *arXiv.Org*, 16 Sept. 2021, arxiv.org/abs/2109.08045.

APPENDICES

Github Repo: https://github.com/IWumbo203/CollaborativeFilteringResearch

YouTube Link: https://youtu.be/4sEyPtnntOE