ILLINOIS TECH

A Performance Comparison of Matrix Factorization & Neural Collaborative Filtering
-Final Project Report-

by: Oussama Hadad - A20521317

Contents

1 Introduction

As the amount of data collected by online companies increased drastically in the last decade, the use of recommendation systems by companies became more and more popular. These recommendation systems suggest to the users new content such as videos to watch, clothes to buy or new destinations to visit.
There are two main types of recommendation systems: content-based filtering recommendation systems and collaborative filtering recommendation systems. For the first type, the user gets recommendations based on their explicit feedback. The system suggests similar content to what the user ”explicitly” likes. For the second type, the system recommends items the user might like based on the preferences of similar users.
Content-based filtering captures the interests of a given user specifically without considering the preferences of similar profiles, and a major disadvantage of this method is the difficulty of suggesting a completely different type of content for the user. This limitation of content-based filtering is a strong point of collaborative filtering and what makes this method the most popular recommendation method.

Many Machine Learning algorithms can be used to build a collaborative filtering recommendation system, and the most popular ones are Matrix Factorization (MF) [1] and the different Neural Collaborative Filtering algorithms [2].
Matrix Factorization is the oldest of both Machine Learning algorithms discussed in this work. The first major paper on MF for recommendation systems was untitled ”Matrix Factorization Techniques For Recommender Systems” [1] by Yehuda Koren, Robert Bell and Chris Volinsky. This work was part of the Netflix Prize competition that these researchers won in 2009 after that MF outperformed the classic nearest-neighbor techniques for recommendation. Then, in 2017, a paper untitled ”Neural Collaborative Filtering” [2] by Xiangnan He & al, introduced Neural Networks inspired models for recommendation systems, and this paper showed through different experiments that these new models outperform the classic MF method. But in 2020, a Google Research team published ”Neural Collaborative Filtering vs. Matrix Factorization Revisited” paper [3] where the last paper results were discussed.

In this project many Machine Learning models were used and discussed in the aim of reproducing some experiments from two reference papers [2, 3] and comparing the performances of different algorithms.
Four different models for Recommendation Systems will be discussed. The first and simplest one is Matrix Factorization (MF), the second model is another version of MF called Generalized Matrix Factorization (GMF), the third is Multilayer Perceptron (MLP) and the last model is a concatenation of both GMF and MLP called Neural Matrix Factorization (NeuMF) [2].
To train and evaluate these models the leave-one-out evaluation was adopted, since this method was widely used in literature and it is the same method we find in the reference papers about Matrix Factorization and Neural Collaborative Filtering [1, 2].
Finally, to evaluate the models to metrics were introduced, Hit Ratio (HR) and Normalized Discounted Cumulative Gain. Both metrics depend on a threshold parameter K, and the impact of K on the evaluation will be discussed later in this work.

The different experiments conducted as part of this project led to the following results:
1° NeuCF models outperform MF on both HR and NDCG metrics for the different values of K.
2° As the number of negative instances in the training data increases, the performances of the NeuCf models and MF become similar. Also, for high values of negative instances,

MF has better performances than NeuMF.

2 Problem description

The most famous Machine Learning models for building collaborative filtering recommendation systems are Matrix Factorization (MF) and Neural Collaborative Filtering (NeuCF) (GMF, MLP & NeuMF).
Comparing the performances of these different models is necessary to choose one for building a recommendation system, and this performance comparison is done through three experiments:
1° Evaluation of the Top K recommended items
2° Evaluation of HR_{10} and $NDCG_{10}$ for different numbers of negative instances in the training data
3° Evaluation of HR_{10} and $NDCG_{10}$ for different numbers of predictive factors

3 Some Models for Recommendation Systems

3.1 Matrix Factorization

Matrix Factorization (MF) is the simplest of the four models used in this project. This algorithm takes 2 inputs, a user index u and an item index i (movie in the figure). Both indices are transformed into embedding vectors p_u and q_i for the user and the item respectively, and the predicted rating $\hat{y}_{ui}$ is the dot product of these vectors:

$$\hat{y}_{ui} = p_u^T q_i = \Sigma_k p_{uk} q_{ik} \tag{1}$$

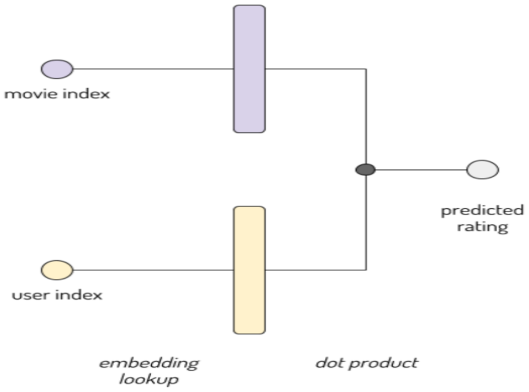


Figure 1: Matrix Factorization Model

3.2 Neural Network Models

The different Neural Network models are defined under the Neural Collaborative Filtering (NCF) framework.

3.2.1 Generalized Matrix Factorization

Generalized Matrix Factorization (GMF) has the same architecture as MF with one difference in the output layer. The last layer has a sigmoid activation function that takes

the dot product as an input.
Given the user and item embedding vectors p_u and q_i, the predicted rating $\hat{y}_{ui}$ is:

$$\hat{y}_{ui} = \sigma(p_u^T q_i) \tag{2}$$

Also, the loss function to minimize is the sum of squared distance between y and $\hat{y}$:

$$L = \Sigma_{u,i}(y_{ui} - \hat{y}_{ui})^2 \tag{3}$$

with: $\sigma(x) = \frac{1}{1+e^{-x}}$

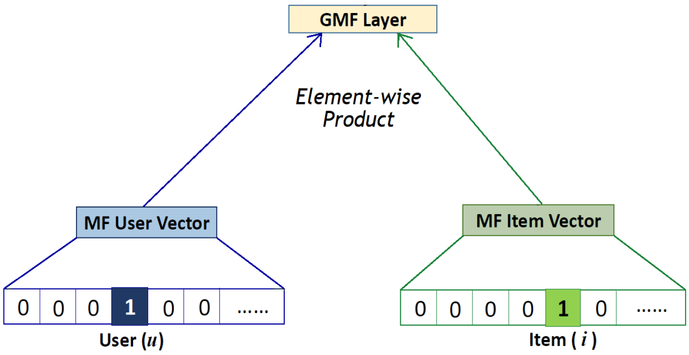


Figure 2: Generalized Matrix Factorization Model

3.2.2 Multilayer Perceptron

Multilayer Perceptron (MLP) is a more sophisticated model with many hidden layers having a ReLU activation function in the hidden layers and a sigmoid activation function in the output layer. The predicted rating $\hat{y}_{ui}$ is computed as follows:

$$\begin{aligned} z_1 = \phi_2(p_u, q_i) &= \begin{pmatrix} p_u \\ q_i \end{pmatrix} \\ \phi_2(z_1) &= a_2(W_2^T z_1 + b_2) \\ &\ldots\ldots. \\ \phi_L(z_{L-1}) &= a_L(W_L^T z_{L-1} + b_L) \\ \hat{y}_{ui} &= \sigma(h^T \phi_L(z_{L-1})) \end{aligned}$$

where a_i, b_i, W_i and h are the activation function, bias vector and the weight matrix for the i^{th} layer, and the weights of the output layer.

3.2.3 Neural Matrix Factorization

Neural Matrix Factorization (NeuMF) is a concatenation of GMF and MLP with a sigmoid activation function in the output layer. The predicted rating is:

$$\hat{y}_{ui} = \sigma(h^T \begin{pmatrix} \phi^{GMF} \\ \phi^{MLP} \end{pmatrix}) \tag{4}$$

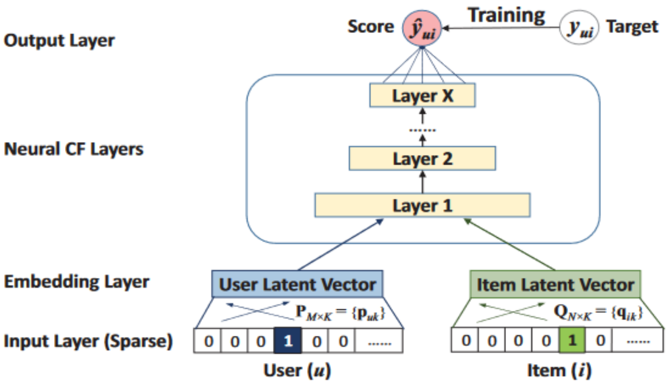


Figure 3: Multilayer Perceptron Model

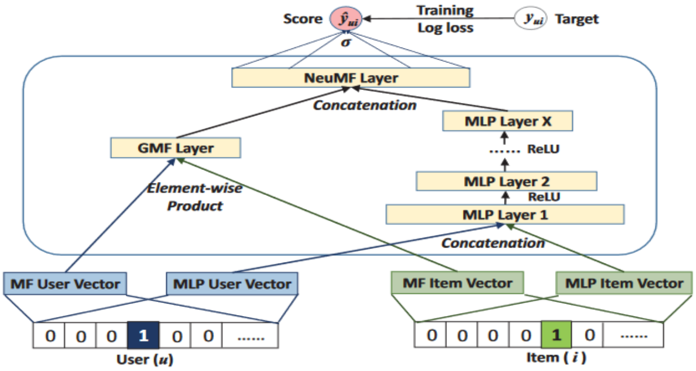
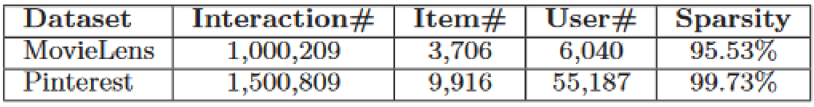


Figure 4: Neural Matrix Factorization Model

4 Data and Evaluation Metrics

4.1 Datasets

The different experiments in this project were performed on 2 datasets, the first is MovieLens 1M datset [4] and the second is a Pinterest datset [5].
These datasets statistics are:

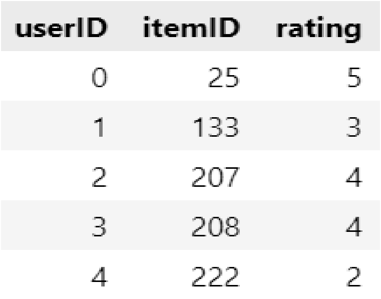


Dataset	Interaction#	Item#	User#	Sparsity
MovieLens	1,000,209	3,706	6,040	95.53%
Pinterest	1,500,809	9,916	55,187	99.73%

Figure 5

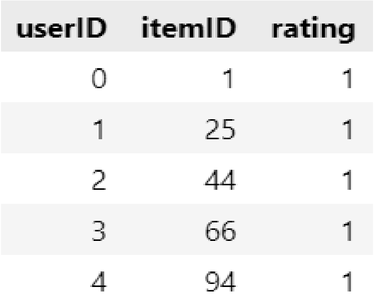
MovieLens 1M is an explicit feedback dataset (Figure 6) of four columns: userID, itemID, rating and time stamp; userID, itemID & time stamp are integers, and rating ranges from 0.5 to 5 since this dataset only contains positive interactions between users and items. Before using this data it was transformed into an implicit dataset by giving a rating of 1 to every positive interaction and 0 to negative interactions.

The second dataset that was used is from Pinterest (Figure 7) and it is an implicit feedback dataset with the following statistics: In order to train and evaluate the models the



userID	itemID	rating
0	25	5
1	133	3
2	207	4
3	208	4
4	222	2

Figure 6: The First Lines from MovieLens 1M Dataset



userID	itemID	rating
0	1	1
1	25	1
2	44	1
3	66	1
4	94	1

Figure 7: The First Lines from Pinterest Dataset

leave-one-out evaluation was adopted since this method has been used in many similar works [4, 5, 6]. Therefore, the last interaction from every user is saved in the testing set and the remaining interactions go to the training set. This training set contains only positive interactions, thus negative interactions should be added. The method that is used consists of adding a certain number of negative interactions for each positive interaction. After updating the training set the evaluation set is also updated by adding 100 random negative interactions for every user. Only 100 negatives are evaluated since computing the predictions for all negative interactions is very time consuming.

4.2 Evaluation metrics

After training the models 2 evaluation metrics were used. The first is Hit Ratio (HR$_K$):

$$HR_K(x) = \begin{cases} 1 & \text{if x is ranked in the top K;} \\ 0 & \text{else.} \end{cases}$$

and the second is Normalized Discounted Cumulative Gain (NDCG$_K$):

$$NDCG_K(x) = \begin{cases} \frac{1}{1+log(rank_x)} & \text{if x is ranked in the top K;} \\ 0 & \text{else.} \end{cases}$$

where: K is a threshold parameter.

5 Experiments and Results

5.1 Evaluation of the Top K recommended items

In this section the models were trained on a training set with 4 negative interactions per positive interaction, and then evaluated on the testing set defined in the previous section.

Figure 8 shows the performance of the different models on the MovieLens 1M dataset and their performances evaluated with the Hit Ratio and NDCG metrics.
The Neural Network models have better performances than Matrix Factorization and this difference increases with K.

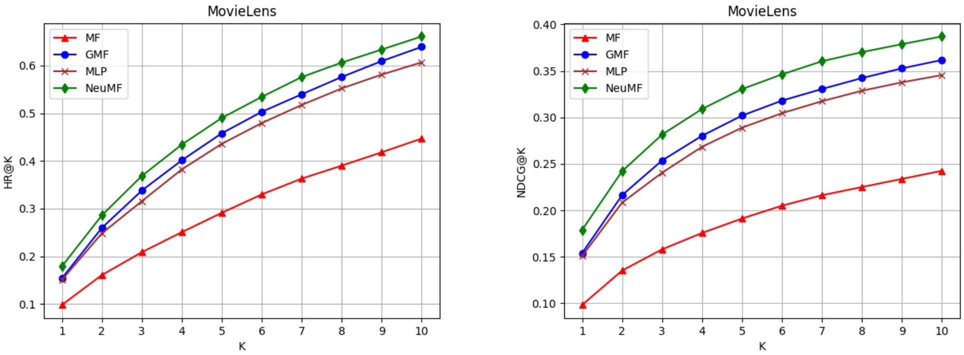


Figure 8: HR_K & NDCG_K for MovieLens

Figure 9 gives the same performances for Pinterest dataset, and it leads to similar conclusions. The NN models outperform Matrix Factorization although the performance difference is smaller for this dataset. To conclude on this section, Matrix Factorization is outperformed by NCF on both datasets. Hence for recommending a large number of items the NCF models are better than Matrix Factorization.

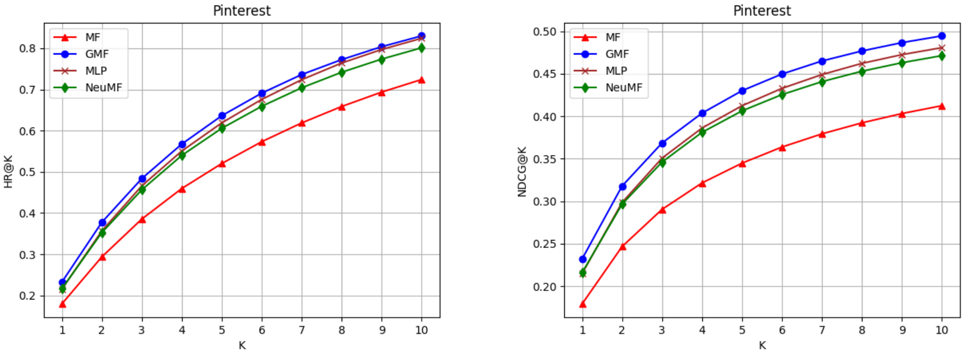


Figure 9: HR_K & NDCG_K for Pinterest

These models were trained on the same datasets with 4 negative interactions per positive interaction. In the next section the impact of this number will be discussed.

5.2 Evaluation of HR_{10} and NDCG_{10} for different numbers of negative instances in the training data

In this section the models were trained on different training sets each with a number of negative samples ranging from 1 to 10, and the performances were evaluated using Hit Ratio and NDCG for K= 10; i.e. evaluated on the top 10 recommendations.
Figure 10 shows the performance of the different models on the MovieLens 1M dataset and their performances evaluated with the Hit Ratio and NDCG metrics.
The Neural Network models outperform Matrix Factorization for small numbers of negative samples, but MF makes better performances as the number of negative instances increases.

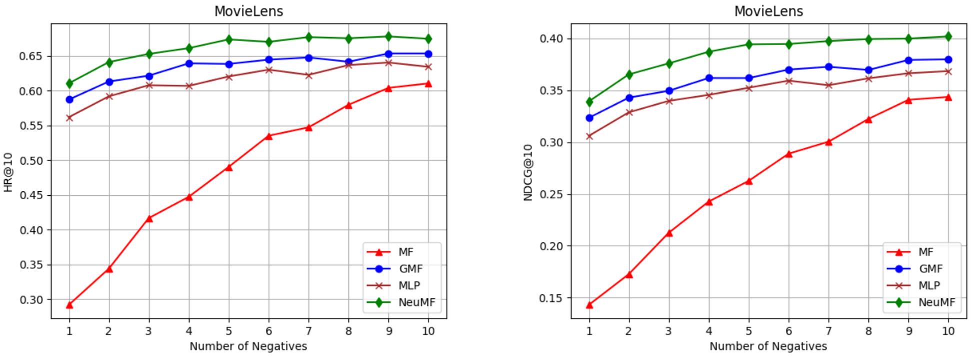
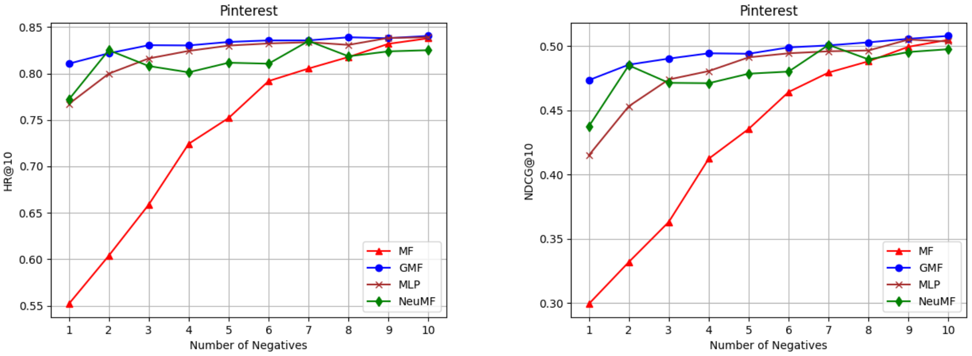


Figure 10: HR_{10} & $NDCG_{10}$ for MovieLens and different numbers of negative instances

Figure 11 shows a similar trend to the previous figure for small numbers of negative instances, but as this number increases MF becomes close to the other models and for a high number of negative instances MF is better than NeuMF and has similar performances to GMF and MLP. To conclude on this section, Matrix Factorization has good performances for high numbers of negative instances. For Pinterest dataset, which is bigger than MovieLens 1M, MF outperformed NeuMF and had similar performances to MLP and GMF.



(a)

(b)

Figure 11: HR_{10} & $NDCG_{10}$ for Pinterest and different numbers of negative instances

5.3 Evaluation of HR_{10} and $NDCG_{10}$ for different numbers of predictive factors

The predictive factors determine the size of the users and items embeddings. This embedding size is the double of the factor. The models performance should be increase as the number of factors increases, but the simulations in Figure 12 failed to reproduce this trend. Hence we failed to do this experiment.

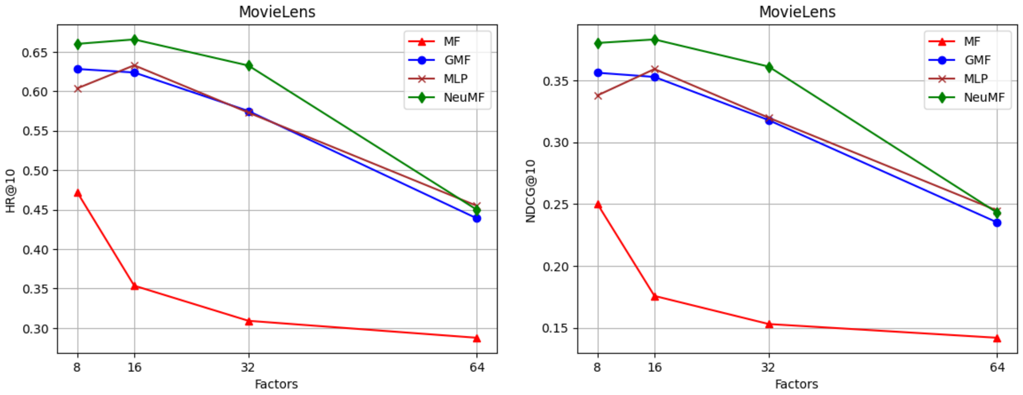


Figure 12

6 Conclusions and future work

This work introduced different models for building collaborative filtering recommendation systems. The first and simplest of the four models is matrix factorization, and the other three, GMF, MLP & NeuMF, are based on neural networks. The performances of these models were compared on some popular user-item datasets such as MovieLens 1M, and the conclusions made from the comparison of these models are:
1) Neural Network models are better than Matrix Factorization when the training is performed with a low number of negative instances.
2) Increasing the number of negative instances enhances the performance of Matrix Factorization, and for a high number of negative instances this method has better performances than some NN models.
The second evaluation shows how Matrix Factorization and NN models have similar performances although the NN models are very complex and need more computation than MF. This point leads to one conclusion, simple models can be as useful as highly complex models.
In future, the impact of the training and testing data form, i.e. whether it is implicit or explicit feedback, on the models performance can be studied. In addition to the efficiency of these models for building online recommendation systems.

7 References

[1] Yehuda Koren, Robert Bell and Chris Volinsky, *MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS*
[2] Xiangnan He & al, *Neural Collaborative Filtering*
[3] Steffen Rendle & al, *Neural Collaborative Filtering vs. Matrix Factorization Revisited*
[2] Y. Koren. *Factorization meets the neighborhood: A multifaceted collaborative filtering model. In KDD, pages 426–434, 2008.*
[3] Xiangnan He & al, *Neural Collaborative Filtering*
[4] F. Maxwell Harper and Joseph A. Konstan. 2015. *The MovieLens Datasets: History and Context. ACM Trans. Interact. Intell. Syst. 5, 4, Article Article 19 (Dec. 2015), 19 pages.*
[5] X. Geng, H. Zhang, J. Bian, and T. Chua. 2015. *Learning Image and User Features for Recommendation in Social Networks. In 2015 IEEE International Conference on Computer Vision (ICCV). 4274–4282.*