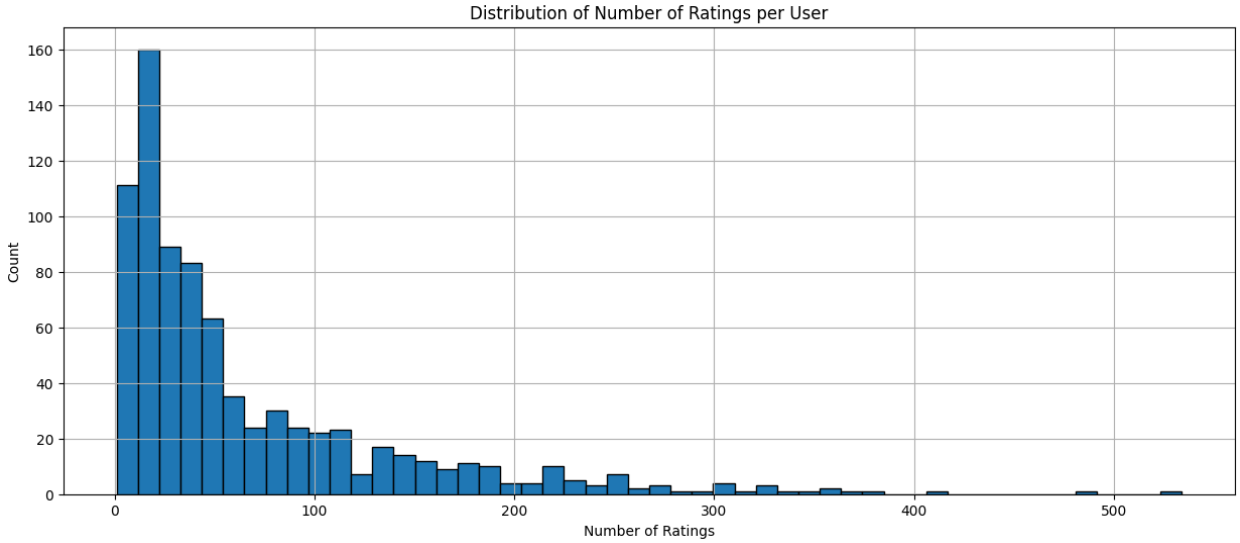
Movie recommendation systems

Introduction

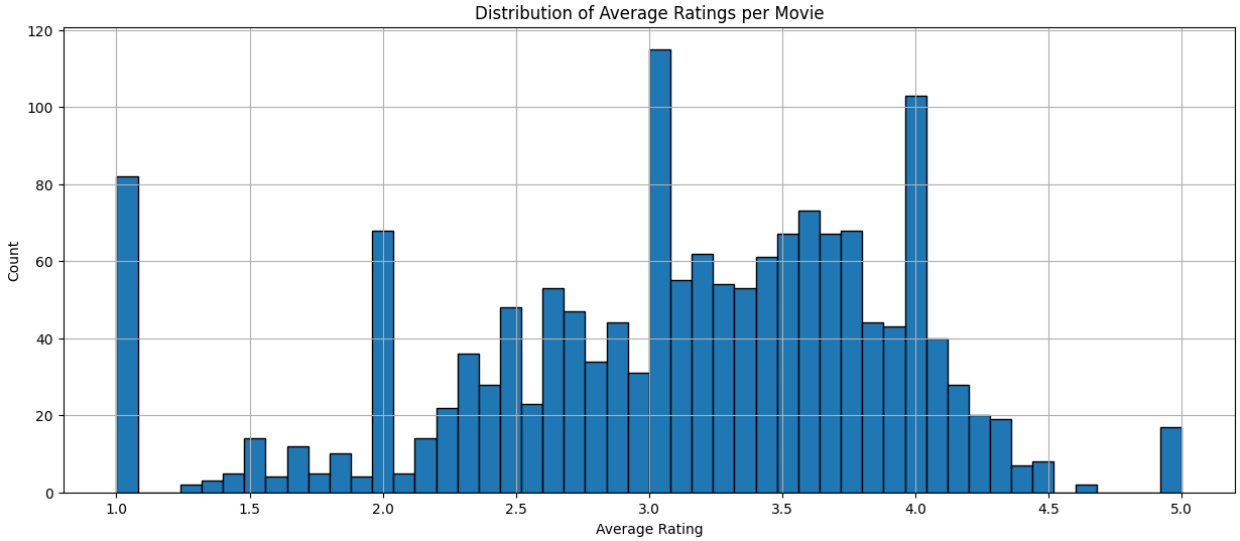
For recommendation systems a common suggestion to represent the relationship of users to itmes as a matrix. The value in intersection is the score on how how likely a user is interested in a given item.

Data analysis

So we are given a dataset where we have a sparse matrix (~1500 movies, but most users scored less than 100 of them). And we want to build a model that would fill it with non-zero values and then suggest high score unwatched films to users.



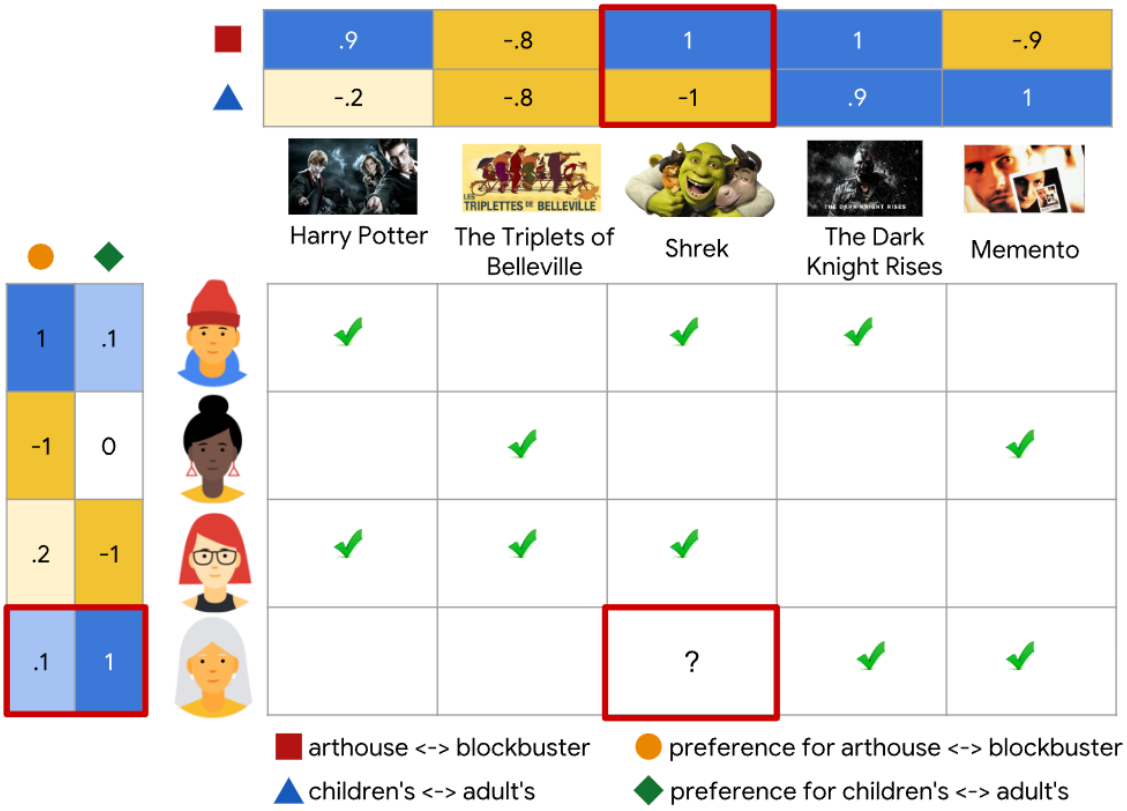
It is particularly interesting that there are a lot of “hated” films with average score close to 1. And some of them are scored completely 5. It’s better idea to take this outliers into account and suggest to each new user 5-rated films and never suggest 1-star.

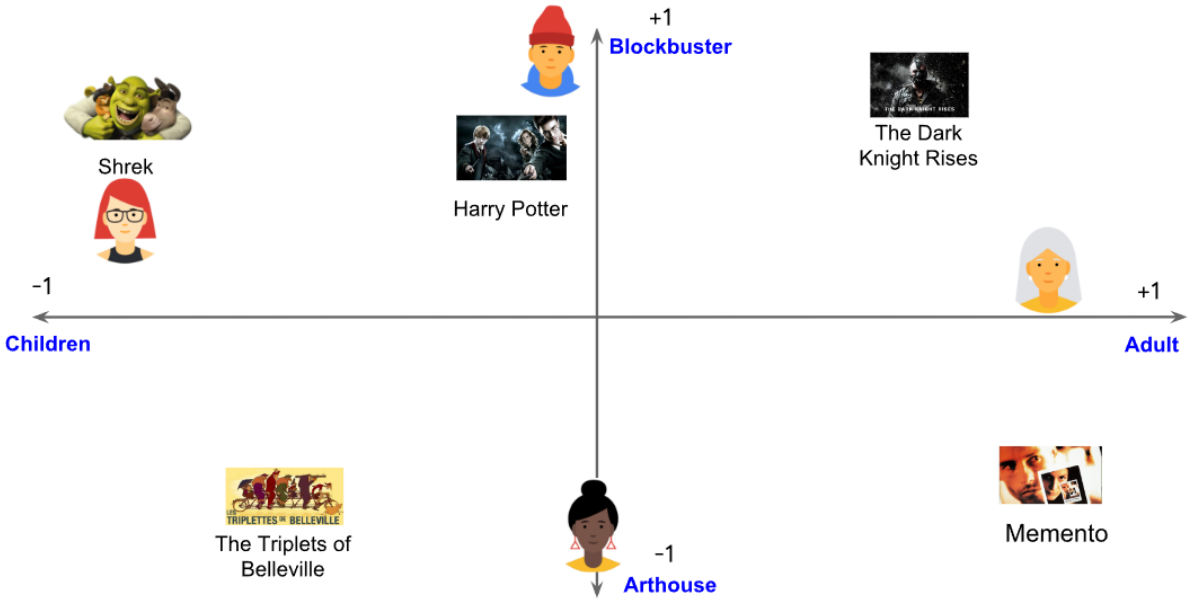


Model implementation

For this assignment I simply run the lab code from the IU PMLDL course and analysed the result. Moreover I added some functionality to use the model.

The basic idea: there is a bipartite graph of users and items. We “smooth” data of users by their connections to each other and aggregate it into some embedding. We are trying to learn a good embeding representation of movies and user such that after multiplication we get the relevance matrix - where all possible scores are stored. Based on the given matrix in turn we make predictions (suggestions) for users.

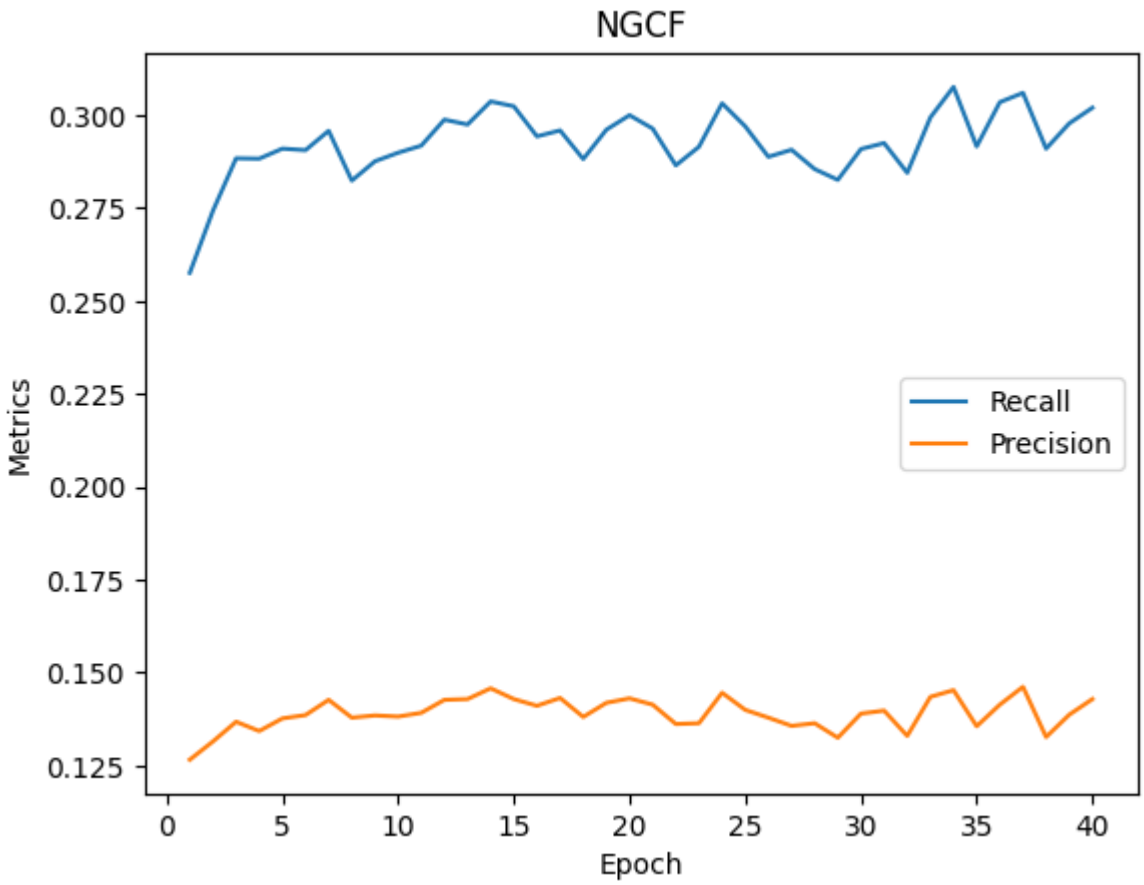


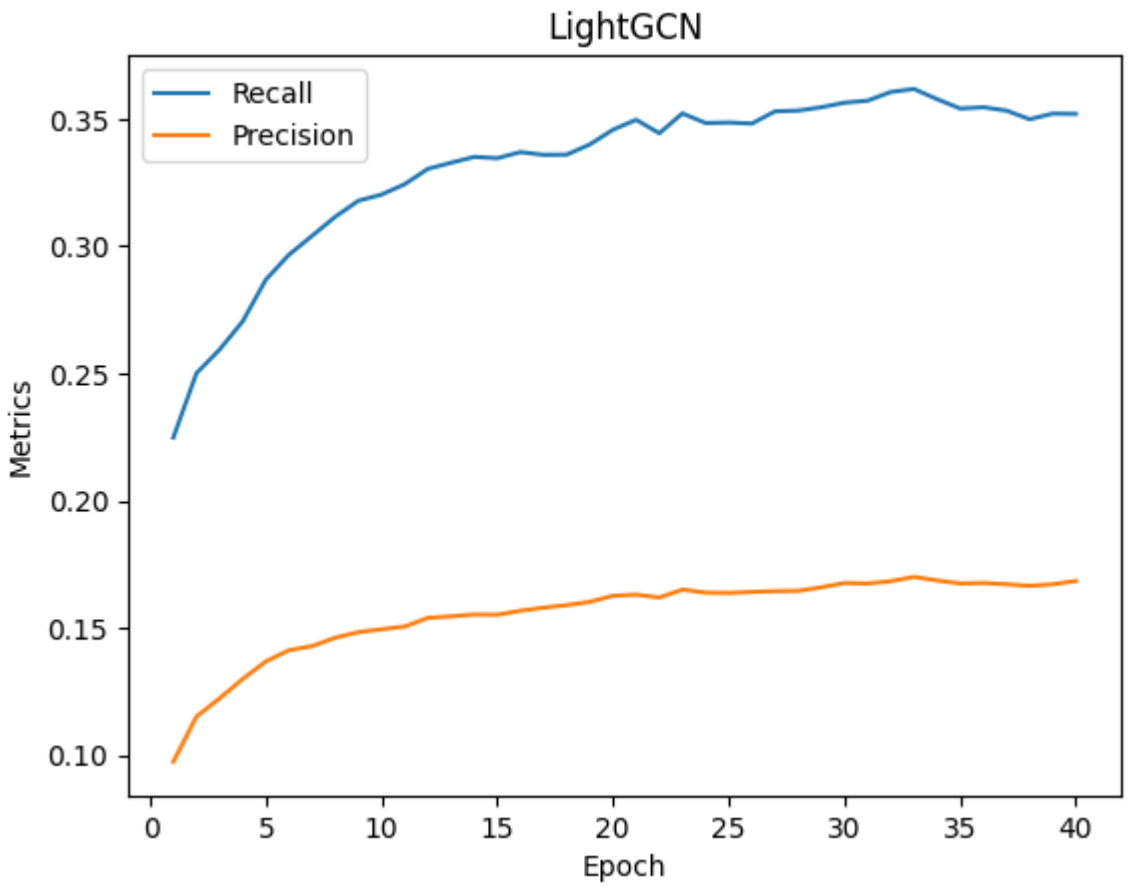


Training and measuring

Since initially we have just a sparse matrix and there is no future info of which movies users will like we have to take out a few movies from the training set so we use them for testing predictions.
After training the network to do embedding that are close to our given matrix we test it. Simply calculate the expected relevance matrix from the whole data embedding, mask out train data. Choose suggestions from it and compare with test pairs.

During the training process I tried to change hyperparameters such as number of layers, K number of top scores, epochs. However it did not really make a change in the results. After 20-30 epochs models converge to their global minimum and only fluctuate by a small percent in recall and precision.


NGCF
Metrics
Epoch
Recall
Precision
0.300
0.275
0.250
0.225
0.200
0.175
0.150
0.125
0
5
10
15
20
25
30
35
40



Results

As you can see from the graphs Light GCN has a slightly better metrics, therefore It's more preferable for usage as the recommendation system.

Advantages of this approach s that we do not really care about our embedding - there is no actual human meaning behind this, so it s easy to change and implement. Moreover, it's fine that the matrix is sparse, we still can get the info by using just convolutions, what would be harder if would try to implement it in a forward pass technique.

Disadvantages are that for a fresh item or user it's hard to give any info of them, because of their few connection to the database. Moreover, this is a collaborative method, we did not use any info about the user in the trainig process. So the model cannot give result by simple giving age, gender, occupation, zip code.

One of the suggestion to address this problem - for a new user or item(movie in our case) look through the list of known users and find the most similar by the provided features, then predict a new film for the “doubler” user and give it to the newbie. Moreover we can just simple take the highest scored film of the “counterpart” user (since the fresh one did not what any at all). This technique is implemented in the end of GNN notebook.

SVD

The solution for this technique was taken from github. This method is kind of the same as the GNN - create a latent space and map user with items to it. However it uses linear algebra techniques, so the result is always deterministic. They also provided a function for search the films in the database, so you can try to get a first film ,then based on its latent space find the closest film by cosine similarity and suggest it. This method is pretty similar to the approach I used in GCN, however the main focus for new user not on their data, but on the film they are watching. This can be beneficial for guest-mode user, since we only know the film they are currently watching.

All other references used in this research you can find in the reference file