Conclusion: Recommender Model

Katherine Whitehead

November 2024

1 Conclusion

As discussed in the introduction, recommender systems are inherently difficult to measure performance as when working perfectly they should suggest a movie that the user would enjoy. It is therefore very difficult to test this with the data available to us, as it is unlikely that users will have watched the most ideal movie for them based on their preferences, and it is even more unlikely that this rating would fall in the test data set. Therefore, having looked at other projects which aimed to achieve similar outcomes, we decided that the MSE would be a good way of gauging how well a model is performing. The models predicted a rating that a user would give for each movie, and then the mean squared error was calculated for those predicted ratings that had the relative actual rating in the test data.

In conclusion, both models were more successful than our base model in terms of value for MSE. We can see the results in the table below:

Model	Base Model	Neural Collaborative Filter	Matrix Factorisation
MSE	1.07	0.92	0.48

According to the MSE result, the matrix factorisation model performed the best by a considerable amount. This implies that this model could best predict what a user would rate a movie based on their previous watching for this data.

A main difference between between our two models is that the matrix factorisation model only considered the most important variables, which were the rating, the User ID and the Item ID. On the other hand the neural collaborative filter included all variables available in the data set.

It's important to consider how reliable our data is. It may be the case, as is often true with streaming accounts, that multiple people are watching movies under the same User ID. Their different preferences and ratings for different movies are likely to skew how our models interpret their likes and dislikes. This adds an element of potential error and reinforces the need to have much more data in order to try and avoid cases like these influencing the model too much.

One of the key questions that we were trying to answer during this task was how well the models would scale up to larger volumes of data. Looking first at the collaborative filter, due to its neural network architecture, it's able to adapt well for large amounts of data, and would work effectively using parallel computations.

One of the issues that effected the performance of the neural collaborative filter model was the sparsity of the data. There was very little data to training and test our model on, which meant that the model did not pick up niche patterns in the data. This is likely to have been the cause of the low coverage score (11.47%) as well, because collaborative filters are heavily influenced by popularity. Hence, with more data available we can expect that the neural model will not only cope well, but actually improve its performance once it has had exposure to a more diverse range of preferences in the user data.

Now moving on to look at the matrix factorisation, although very effective and its simplicity has benefits in terms of computation cost, if the model is adapted to handle massive data there may be issues that appear. Firstly, as the number of user and items grow, the size of the factorized matrices increases. This can create issues for memory storage and computation time. In terms of efficiency and cost, we may find that there are not enough resources available to be able to achieve this up-scaling of the model. Another disadvantage is that matrix factorisation can only find linear relationships between the data and is unable to capture more complex patterns that a neural network would be able to.

Overall the neural collaborative filter method would be much more effective when applied to massive data. It is designed to cope with lots of complex and sparse data whilst retaining a relatively low computational cost, which a matrix factorisation model may struggle to do with limited resources. Another advantage is that in a real-world setting, we would expect new users to join the network, adding these extra features to a neural network is a lot easier to implement and train on than in a matrix factorisation model.