Q5 Report

The intention of this question is to build a recommendation system using features other than user ID, movie ID and rating given by a user or group of users as in user-user and item-item collaborative filtering.

Available features outside of rating, userid and movieid are Age,Genres and Gender.

In my algorithm I have build an neural network consisting of 3 parallel embedded layers and one input for age. To use the embedded layer I first encoded Genre and Gender to numerical encodings. Then I fed Genre, Gender and UserID into their respective Embedding layers.

An embedding layer embeds each unique category into a trainable dense vector that represents the unique category. This is specially useful when we have a large number of categorical attributes. We have 299 Genres of movies hence Embedding is very useful instead of one-hot encoding the entire 299 categories making the dataset very sparse and neural network very large. Each embedding also has a output dimension where whatever the input size of the attribute, the output dimension is fixed sized vector.

As per the data description file, age belongs to 7 categories and are in fact discrete ordinal values. Hence I turned this from a numerical attribute back into ordinal and applied an embedding layer on it too.

All embedded layers are reshaped and concatenated and fed into a Dense output layer of 32 neurons and finally collected in one dense unit which does linear regression. Optimizer Adam is used with loss function being mean squared error.

Advantage of my approach is that it allows to use a more generalized approach for each user without having to depend on memory. This is a model based approach instead of a typical memory based approach like collaborative filtering. Furthermore it allows to account for the changes in user preference at any given time, as my approach userid,age and genre at any play an important role in specifying the rating than depending on the users past ratings on movies, which can change over time as users can have phases of liking certain genres of movies.

<u>Task 2</u>

Typical memory based collaborative systems depend heavily on similarity measures between users-uses and items-items, while model based techniques like the one I have built above try to use machine learning algorithms to train a vector of items for a specific user. Now for the same user, they can easily predict the rating by that user whenever a new item/movie is added.

In my model above, there is no movie id used as an input. However movie id would be an input in a collaborative filtering algorithm. This allows my model to be more generalized to a user, and item attributes than to the specific item identity itself.

In my jupyter notebook, I have attempted to build a item-item collaborative filtering system from scratch.

The algorithm is as follows

1) Create a pivot table with userID as index and movieID as column, with every cell giving the rating. This is a very sparse matrix as many users don't rate many movies.
2) Then I make a correlation matrix on the movies gives the correlation between each movie. To avoid affects by spurious ratings which suggest no trend, I have used min_periods = 50 to build the correlation matrix
3) Then for each user:
4) Obtain the movies to predict ratings for
5) Obtain the user ratings given by the user in concern for other movies
6) For each movieA in the movies the user has rated:
7) Find the correlations with the other movies ("movielist") for that movie
8) Multiply the user rating for MovieA with the correlations of all movies in "movielist"
9) Make a list of all the movies that have been rated based on correlation and user rating of MovieA("ratinglist")
10) For each movie to predict the rating of:
11) Find the rating from the "ratinglist" which was computed above

Due to time constraints I could not obtain the results from this code but the code is available on the Jupyter notebook for review.

https://keras.io/api/layers/core_layers/embedding/

https://stackoverflow.com/questions/69143694/concatenating-parallel-layers-in-tensorflow