MovieLens Project Submission

Johannes Resch

2025-01-17

Introduction / Overview

This report describes a movie recommendation system developed by applying machine learning and statistical methods to publicly available movie review score data provided by GroupLens (a research lab in the Department of Computer Science and Engineering at the University of Minnesota).

The goal of the recommendation system is to predict for a given user (which has provided a number of movie ratings) which rating the user would give for movies not having received a rating from this user so far.

First, to have a reasonable baseline, we implement the best performing model develeoped in the prior PH128.8 course exercise ("Regularized Movie + User Effect Model"). We train this model based on train set, then predict for test set and final holdout set. Resulting RMSE for "final holdout" set is 0.8654.

Then, trying to apply a significantly more advanced and computationally demanding model, we train a model based on matrix factorization, using the "recosystem" wrapper package (https://github.com/yixuan/recosystem). Resulting RMSE of this model on the "final holdout" set is 0.78729. This is very close to the winning algorithm of the Netflix challenge (at RMSE 0.7865), although handling of the required MF model via "recosystem" is very straightforward and thus usable even for subject matter beginners such as the author of this report.

Methods/analysis

As per the given assignment, this analysis is based on the 10M variant of the MovieLens dataset, available for download at https://grouplens.org/datasets/movielens/10m/. Reviewing the corresponding description of the dataset (https://files.grouplens.org/datasets/movielens/ml-10m-README.html), it is worth noting that only ratings from users which gave at least 20 ratings are included in the dataset.

Based on prior modules of the HarvardX PH125 course series, we consider this to help improve the quality of predictions, as trying to predict for users with only a very small number of provided ratings tends to have a significantly higher variance.

Ratings in this context are described by a "star values" described as non-continuous numeric value from 1 (worst) to 5 (best), with "half star" ratings expressed as .5 (e.g. 3 and a half star rating would be encoded as numeric value 3.5).

Initial processing of the dataset was done as prescribed in the PH125.9x introduction, using R code provided by HarvardX. This initial processing includes:

- from the downloaded dataset consisting of multiple files, creating a single R dataframe object to include records and movie data, with userID, movieID, rating and timestamp as integer values
- splitting the dataframe in two partitions: a larger one (approx 90%) for algorithm test/development and a smaller one (approx 10%) referred to as "final hold-out test set" to be used only for final RMSE evaluation of the recommendation system. This partition is explicitly not to be used for training and algorithm selection/tuning purposes as per provided project instructions.

Reviewing the dataset, we note that the parameter "genres" consists of one or more of 18 genre categories, which are concatenated using a "|" character in case a movie is associated with multiple genres.

In this report, we skip further description of exploratory data analysis, as these steps were already performed in the PH125.8 course module. To recap, we note that there are user effects, movie effects, and further correlations based on categories, movie year etc. visible in the dataset.

The following steps were used in the process of deriving a recommendation system:

- define RMSE function to have a way to assess quality of a given algorithm for the recommendation system (based on code from PH125.8 exercise)
- further partition "edx" dataset into train/test dataset (85/15 split)
- for "baseline" RMSE using "Regularized Movie + User Effect Model" from PH128.8 course module, we refactor related code taken from the course exercises, and predict for test and final holdout data sets
- for the advanced MF model, we define datastructures required for the "recosystem" package as per the package documentation (https://github.com/yixuan/recosystem). As for performance, we note that running the model optimization with 20 iterations and 12 threads takes approx. 10 minutes on a Macbook Pro M4 Max device
- for MF model training, we determine the best value for the "iteration" parameter by testing against the test set, and selecting the number of iterations that have best RMSE
- finally, we predict using the MF model for the final holdout data

Results

First, we demonstrate our "simple" model baseline based on "Regularized Movie + User Effect Model"

```
RMSE(y_hat_simple, final_set$rating)
```

```
## [1] 0.865445
```

The advanced MF model shows RMSE in a whole different league and is very close to the winning RMSE of the Netflix challenge (https://www.researchgate.net/publication/359010998_Netflix_MovieLens_Project)

```
RMSE(y_hat_mf_final, final_holdout_test$rating)
```

```
## [1] 0.7872998
```

Conclusion

We conclude that using R and available 3rd party packages such as "recosystem", even advanced matrix factorization recommendation models are easily accessible by inexperienced users, without requiring the user to fully grasp the involved complex mathematical background.