MovieLens Recommendation System

Loren Grooms

01/07/2020

Introduction

MovieLens.org is a website created by GroupLens Research containing over 25 million ratings for thouands of movies by a multitude of online users. Using a subset of this data, this project aims to utilize these ratings in order to form a movie recommendation algorithm that will help users find similar movies to ones they enjoy. In creating this algorithm, we will calculate the average movie rating overall, the average rating per movie, average rating per user, and then factor in a penalty for small sample size. Combining these calculations we will form a list of predicted movie ratings, which we will test by comparing to a validating set of real ratings.

Method

This analysis begins by installing any required packages that the user is missing, then downloading a 10 million-entry subset of MovieLens's online database. Delimiters are then removed from the data and it is organized into separate tables by rating and movie, which are then converted into data frames and merged. This constitutes the primary data to be analyzed. Ten percent of this data is then set aside as our final validation set, and the remaining data is split again to form test (20%) and training (80%) sets. To form our predictive model, we consider the individual ratings themselves $y_{u,i}$, the overall average rating μ, per-movie average ratings b_i, and per-user average ratings b_u in our training set. Using these observations and calculations, we can predict ratings using the following equation:

$$Y_{u,i} = \mu + b_i + b_u$$

To improve our prediction, we will regularize our data to account for small sample sizes with the following equation:

$$\frac{1}{N}\sum_{u,i}(y_{u,i} - \mu - b_i - b_u)^2 + \lambda(\sum_i b_i^2 + \sum_u b_u^2)$$

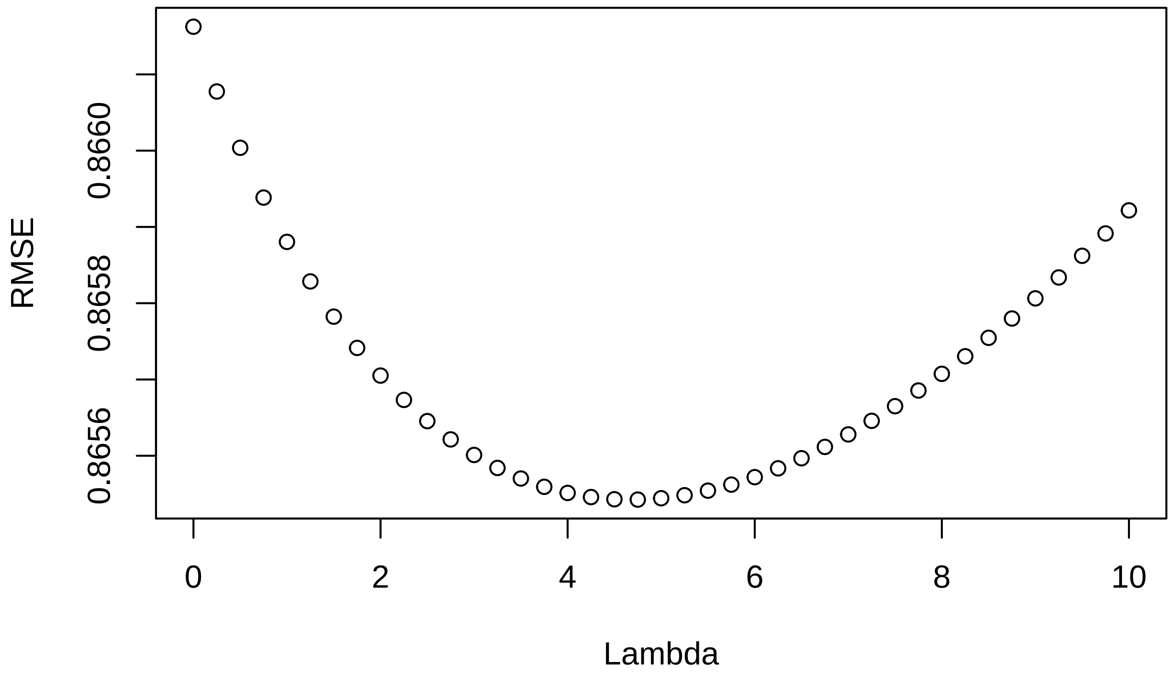
λ is a tuning parameter, so we use cross-validation to select the λ which results in the lowest RMSE. We do this by testing a sequence of possible parameters in the regularization equation, using the regularized data in the prediction equation, then finding the RMSE with each prediction compared to the test set. We use the following equation to calculate RMSE:

$$\sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$$

Once the ideal λ is identified, we can apply the equation once more to the overall set and calculate the RMSE using the validation set.

Results

The results of the cross-validation can be seen in the following graph:



We can locate the λ resulting in the lowest RMSE using the following code:

```
tuning_params[which.min(test_errors)]
```

```
## [1] 4.75
```

After running the final regularization and prediction equations with the ideal λ, we can calculate our RMSE using the validation set:

```
RMSE(predicted_ratings, validation$rating)
```

```
## [1] 0.8648201
```

Conclusion

Using MovieLens's extensive review database, we have identified and accounted for several interfering factors and created a fairly effective movie rating prediction algorithm. Further iterations should include more corrections for additional effects as they are discovered and studied.