Movie Recommendation Algorithm

Nicolas Kirsch

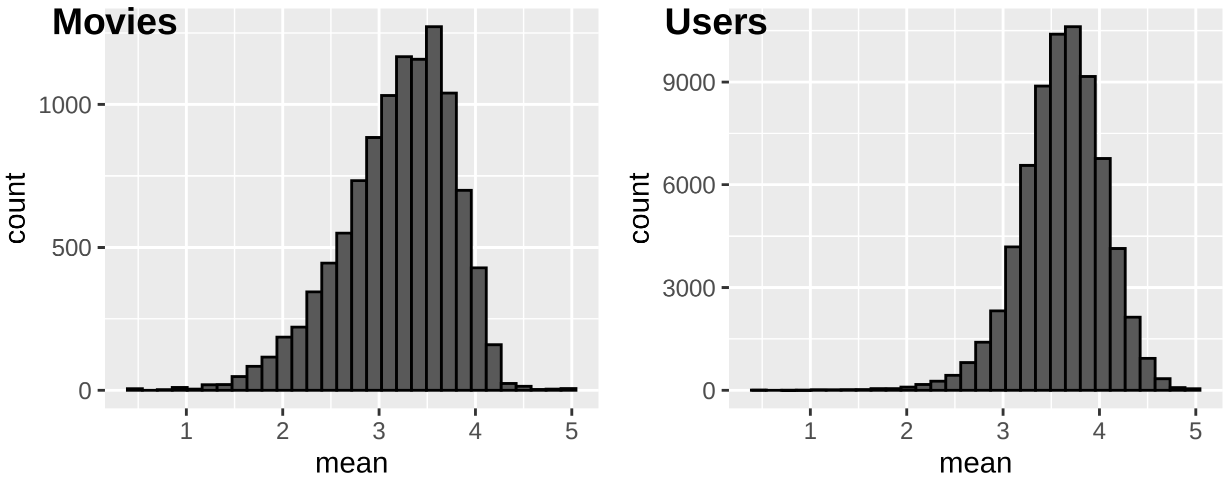
16/09/2019

Introduction

The popularity of recommendation systems has grown with the development of applications and websites offering personalized products to their users. Netflix and Amazon use such systems. They use the ratings given by users to different products to predict how they would rate other products and hence whether they would be interested by them. Similarly, our objective was to create an algorithm predicting the rating a specific user would give to a specific movie. To do so, we used a subset from the "movielens" dataset in the "dslabs" package. This subset was divided in two. The edx data frame represented 90% of the data and was used to train our models. The validation data frame represented the other 10% and was used to test our models. There were 9000055 rows in edx and 999999 in validation. Each row of the dataset corresponded to the rating one user gave to one movie. A user or a movie could therefore appear more than once in the dataset. The dataset also had six columns: UserID, MovieID, Rating, Timestamp, Title and Genres. We aimed to study the user and movie effect on the ratings and whether this effect could be tuned to give the best prediction possible.

User and Movie effect

First, we used exploratory data analysis to gain insight on the dataset. We wanted to evaluate the variability of ratings between users and between movies. To do so, we computed the mean rating for users who rated more than a hundred movies. We did the same for movies rated more than a hundred times.



We can see that there is a strong variability across movies and users. Therefore, users and movies have an impact on ratings. They bias the rating and therefore were useful for our prediction algorithm. Our algorithm was:

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

With Yu,I the real rating, mu the mean rating across the edx dataset, bi the movie bias, bu the user bias and epsilon u,I the independent errors

We tested this algorithm on the validation dataset. The RMSE value between the validation ratings and the predicted ratings was **0.6853**

Regularization

This value was good, but ameliorating it was possible. We decided to look at the movies with the best predictions from our model. Those movies were not well known and only had a very small number of ratings.

```
## Joining, by = "movieId"
```

```
## # A tibble: 5 x 3
##   title                                  b_i     n
##   <chr>                                <dbl> <int>
## 1 Hellhounds on My Trail (1999)         1.49     1
## 2 Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994)   1.49     2
## 3 Shadows of Forgotten Ancestors (1964)  1.49     1
## 4 Fighting Elegy (Kenka erejii) (1966)   1.49     1
## 5 Sun Alley (Sonnenallee) (1999)         1.49     1
```

We found that this was also true for the movies with the worst predictions.

```
## Joining, by = "movieId"
```

```
## # A tibble: 5 x 3
##   title                                      b_i     n
##   <chr>                                    <dbl> <int>
## 1 Besotted (2001)                          -3.01     2
## 2 Hi-Line, The (1999)                      -3.01     1
## 3 Accused (Anklaget) (2005)                -3.01     1
## 4 Confessions of a Superhero (2007)        -3.01     1
## 5 War of the Worlds 2: The Next Wave (2008) -3.01     2
```

We therefore had to modify our algorithm to make it penalize movies with extreme ratings coming from a very small number of users. We wanted to do the same for users giving extreme ratings to a very small number of movies. To do so, we added regularization to our algorithm. Thanks to regularization, we were able to hinder the impact of these kind of movies by making their bias smaller if the number of ratings received was also small. Similarly, the user bias was smaller if the number of ratings given was small. We had:

$$\boldsymbol{b_i(\lambda) = \frac{1}{\lambda + n_i}\sum_{u=1}^{n_i}(Y_{u,i} - \mu)}$$

$$\boldsymbol{b_u(\lambda) = \frac{1}{\lambda + n_i}\sum_{u=1}^{n_i}(Y_{u,i} - b_i - \mu)}$$

For these two equations, lambda was a tuning parameter. We thus performed cross validation by dividing the edx dataset in two (train_set and test_set) to find the optimal value for lambda.

```
test_index <- createDataPartition(y = edx$rating, times = 1, p = 0.2, list = FALSE)
train_set <- edx[-test_index,]
test_set <- edx[test_index,]
```

We calculated the regularised user and movie biases for each lambda using the train_set and created a model using them. We then calculated the value of the RMSE between the ratings of the test_set and the predicted ratings from our model. To make sure there would be no NAs in our cross validation, we semi joined test_set and train_set so that users and movies would be common to the two subsets. We tested values for lambda from 0 to 10 with an increment of 0.25.

We kept the lambda minimizing the RMSE.

```
lambda <- lambdas[which.min(rmses)]
```

With this lambda, we regularised the user and movie biases for the whole edx dataset. We then created a new, regularised predictive model. We tested this algorithm on the validation dataset and got an RMSE of **0.8648**.

Results

```
## # A tibble: 2 x 2
##   method                  RMSE
##   <chr>                  <dbl>
## 1 Movie and user effect 0.865
## 2 Regularized model     0.865
```

Comparing the RMSEs obtained from the two models we created, we were able to see that regularizing the biases has a significant impact on the RMSE.

Conclusion

In conclusion, we attempted to predict the rating a specific user would give to a specific movie. To do so, our model considered the bias each user and movie had on the rating. This gave us a value of 0.8653 for the RMSE. We then tuned the two biases using regularization and got an RMSE of 0.8648.

Though this RMSE is satisfying, it could have been even smaller. This machine learning algorithm only considers the impact specific users and movies have on the rating. It is not using groups of similar movies or groups of similar users for its predictions. For example, the influence of a movie's genre or of the period during which the movie was produced has no impact on the prediction even though it has an impact on its rating.