MovieLens Project Report

Ngai Chun Tsui

3/7/2021

Introduction

In 2006, Netflix offered a challenge to improve their recommendation system. The movie data look like this:

```
##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18124   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :35738   Median : 1834   Median :4.000   Median :1.035e+09
##  Mean   :35870   Mean   : 4122   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53607   3rd Qu.: 3626   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
##  Length:9000055     Length:9000055
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

Our goal is to find an algorithm that accurately predicts users' ratings on movies, so that Netflix can recommend movies to users.

Key steps of our study:

1. Preparation of Data: We separate the movielens into 'edx' and 'validation'. This is described in **Create Train and Final Hold-out Test Sets** section on edX platform. We use 'edx' data set for training and testing. The 'validation' data set is only used for final calcuation of Residual Mean Squared Error (RMSE).
2. Model Training: We use regularized movie and user effect plus Principal Component Analyisis (PCA) effect model. Mathematically, we assume that the rating Y_ui is: $Y_{ui} = mu + b_i + b_u + p_{u1} * q_{1i} + p_{u2} * q_{2i} + ... + e_{ui}$

- Y_ui = The rating of user u on movie i
- mu = The average rating of all movies
- b_i = The average rating of movie i (Movie effect)
- b_u = The average rating of user u (User effect)
- p_u1 = Entry of matrix P in PCA. The user u preference for the 1st principal component.
- q_1i = Entry of matrix Q in PCA. The effect of the 1st principal component on movie i.
- e_ui = Independant identical random error centered at 0.

We separate the edx data set into train and test data set. We use the train set to train models and use the test set for testing RMSE and tuning parameters.

3. Prediction Using Validation Data Set: After finding the best parameter, we fit the models to the entire edx data set. Then we predict using validation data set. The final result of RMSE will be calculated.

Analysis

In this section, we will see how our model is trained and the rationale behind.

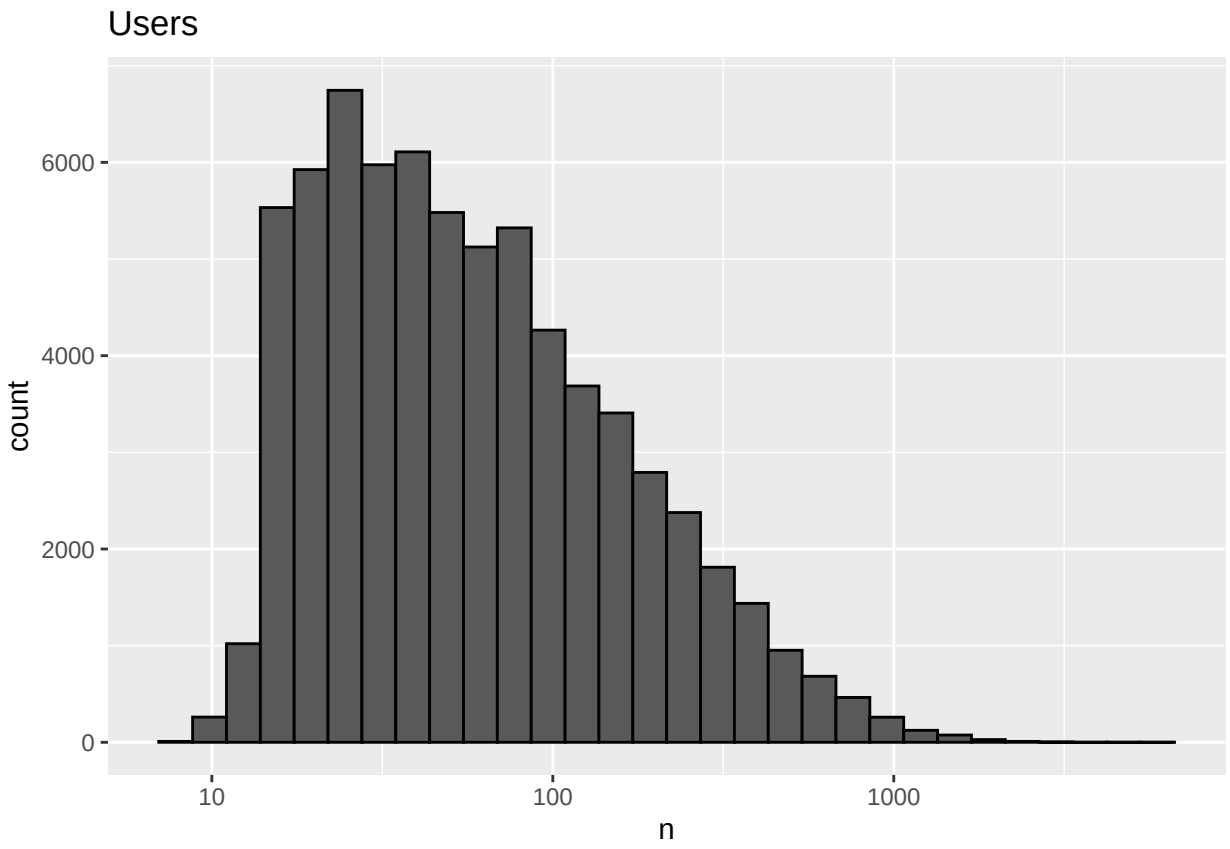
1. To begin with, we need to define a function to calculate the RMSE. The RMSE will be a measure to see if the model is effective or not.

```
RMSE <- function(true_ratings, predicted_ratings) {
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

2. We separate the 'edx' data set into 'train' and 'test' data set.
3. We take a look at the histogram of ratings grouped by users and movies.

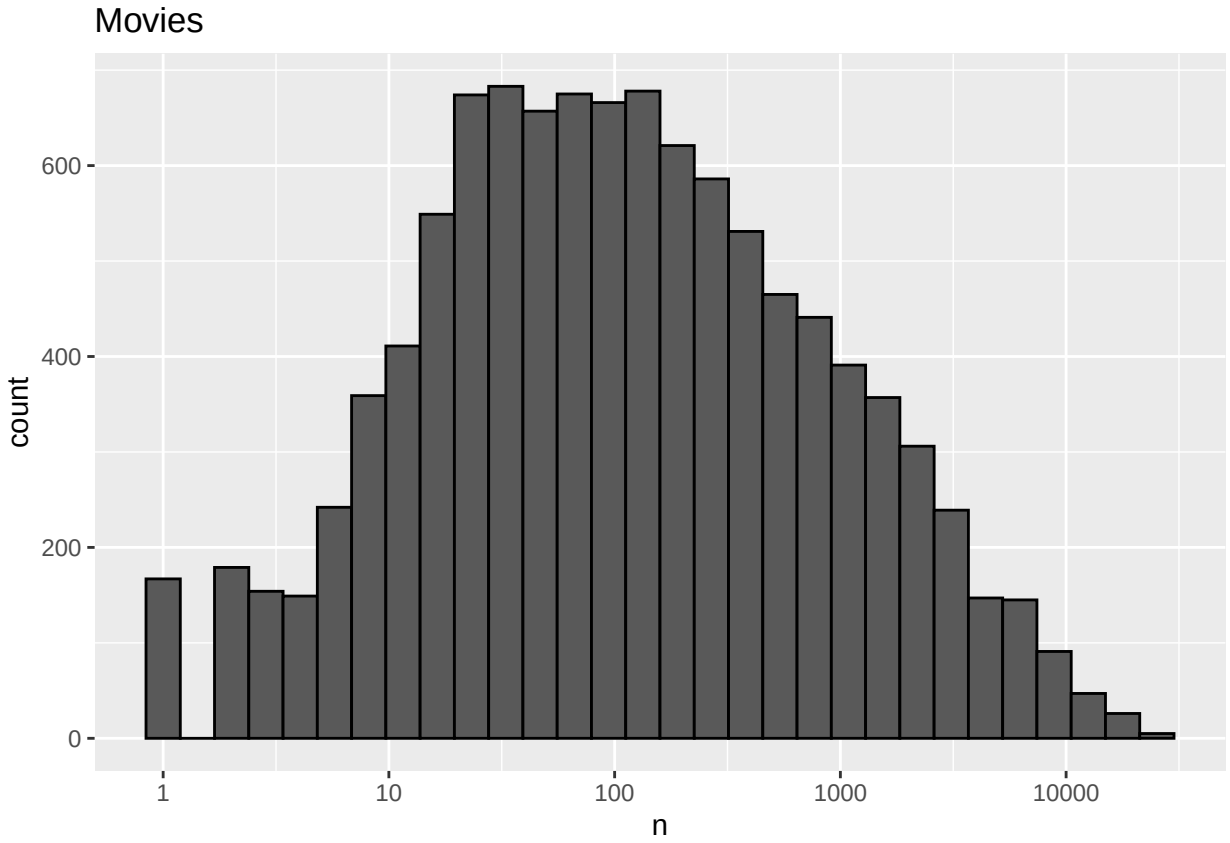
Histogram of the no. of ratings per user

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



Histogram of the no. of ratings per movie

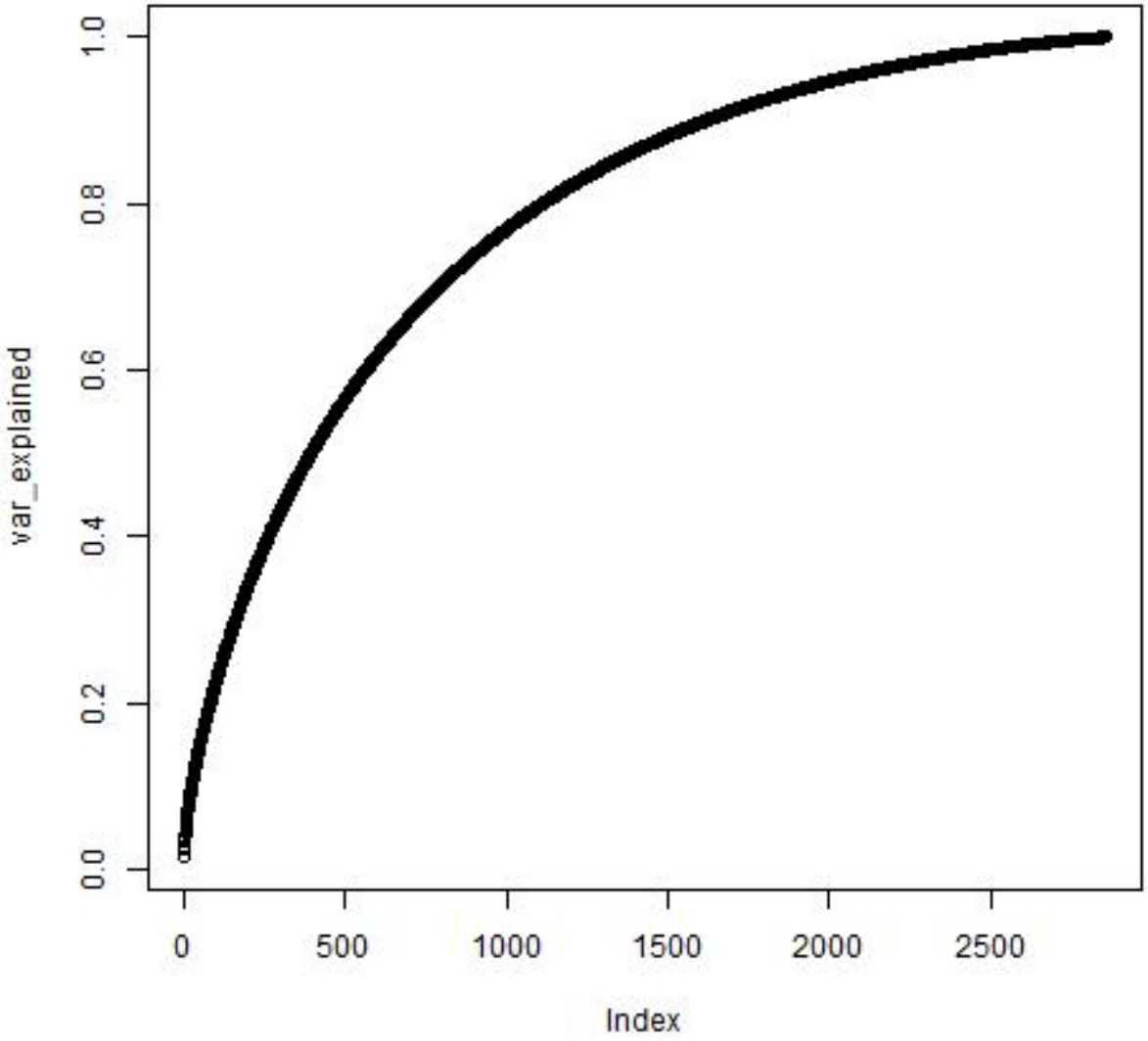
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



We see that there are users who give very few ratings and movies which receive very few ratings. If we use all the data to do PCA, it will consume a lot of computing power and sometimes reach over the memory limit. Hence, we will filter and remove users with less than 100 ratings and movies with less than 400 ratings. The remaining data will be used for PCA.

4. To perform PCA, we use spread() to expand the train data set to a data frame with columns equal to movie IDs. Then we cast it into a matrix and remove the row means and column means. The result will be residuals which cannot be explained by the movie or user effect. We use the function prcomp() to perform PCA.

The pca result shows that, with about 400 principal components, it explained over 50% of variance.



5. We first train the model using regularized movie and user effect. We do not include PCA effect at this moment because the PCA result is very large and the computer speed is slow when tuning the lambda. We train the model and find the lambda that minimizes RMSE. Then we can use the best lambda in the model of regularized movie and user effect plus PCA effect.

The word 'regularized' means that we penalize small data sample by dividing the sum by (lambda + no. of sample).

```
b_i <- train_set %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(l + n()))
```

After tuning, the RMSE is 0.86555 and the best lambda is 5.

6. With the best lambda, we can now combine the regularized movie and user effect and PCA effect. We predict using the test set to obtain the RMSE. As our model suggests, we assume: $Y_{ui} = mu + b_i + b_u + p_{u1} * q_{1i} + p_{u2} * q_{2i} + ...$

From the coding, the prediction is made by adding the mu, b_i, b_u and res. The 'res' is from the PCA. It is the residual explained by the first 400 principal components, which account for over 50% of variance.

```
predicted_ratings <- test_set %>%
  left_join(b_i, by = "movieId") %>%
  left_join(b_u, by = "userId") %>%
  left_join(pca_df, by = c("movieId", "userId")) %>% #Left join the PCA result
  mutate_at("res", ~replace(., is.na(.), 0)) %>% #Replace the NA values by 0
  mutate(pred = mu + b_i + b_u + res) %>%
  pull(pred)
```

The RMSE from the test set is 0.85308. This seems better than the original regularized movie and user effect (0.86555). We can now fit the model using the whole edx data set.

7. In previous steps, we divide the edx data set into train and test data set to tune lambda. Now we use the whole edx data set to fit the model. We use the best lambda found (lambda = 5). We calculate the average rating of each movie and that of each user and compute the PCA matrix.

Results

We predict the validation using the fitted model. The result RMSE is 0.84628. This is better than the RMSE of regularized movie and user effect model alone, which is 0.86482. This is better than guessing with sample average, with RMSE = 1.06120.

Conclusion

The combined model of regularized movie and user effect plus PCA effect is better than the regularized movie and user effect model. With the combined model, we have a RMSE of 0.84628. With the regularized movie and user effect model alone, we only have a RMSE of 0.86482. The improvement is obvious. The PCA effect definitely explains some noises which cannot be explained by movie and user effect alone.

While the RMSE is satisfactory, there are some limitations in the model. As the edx data size is too large, the computing speed is slow if we use all the data to do PCA. The computer that we use is 2 core CPU + 16GB ram. Sometimes RStudio will prompt message that we are over the memory limit. We have to filter out users and movies with few ratings, so that we can do PCA. If we have a more powerful computer, we can include more data to do PCA.

When tuning parameter for the model, we did not use cross validation. We only used one training set and test set for tuning. We think cross validation can be done but it will also add complexity to the project. We did not include PCA effect in tuning because the speed was too slow. We could only test the PCA effect after we got an optimized lambda. This is not ideal.

When we used the test set to get an RMSE, we did not use cross validation. This may be done to get a more accurate RMSE.