An Approach to Movie Rating Predictions

Lizz Judge

6/16/2019

Introduction

The instructor-provided data set for this project consists of movie rating records similar to what one might find in a Netflix or Amazon Prime data base. The training set has a little over nine million records, and the test set is about 1/10th that size. The data collected for each movie are a numerical identifier for the user making the rating (userID), a numerical identifier for the movie (movieID), a numerical rating between 0.5 and 5 given by the user with userID to the movie with movieID (rating), a timestamp (timestamp), the title of the movie being rated (title), and genre of the movie being rated (genres). The goal of the exercise is to use the available data to predict the rating of a particular movie by a particular user while minimizing the loss function.

Methods

The loss function for the exercise is the root mean square error (RMSE), defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{effect}(\hat{y}-y)^2},$$

where $\hat{y}$ is the predicted rating, y is the actual rating, and $effect$ is the feature used to make the prediction (user, movie, genre).

The most naive, or zeroth order, data-based approach to predicting a movie's rating is to use the average of the training set (a naive non-data-based approach might be to assign an arbitrary rating):

$$\hat{y}_n = \text{mean}(rating) = \mu.$$

The value of the loss function for the Naive Model is

$$\text{RMSE}_n = 1.0603.$$

Degree of improvement upon this loss function is the metric of success for the current analysis.

The naive approach does not take into account that some movies are more popular than others, some genres are more popular than others, or that users bring a particular style to their movie ratings. An example of the former effect is that some movies are designed to have broader mass market appeal than others and will thus have a larger percentage of good ratings, bringing the over all average up. A model that does not take this into account will have artifically low predictions for popular movies and artificially high predictions for movies that were widely panned, and similarly for genres and users. The next order corrections to the model include the effect of movie, genre, and user:

$$\hat{y} = \mu + b_m + b_u + b_g,$$

where b_m, b_u, and b_g are the correction terms for movie, user, and genre, respectively.

We must ensure that all data used to validate the model contains identical factors to the data used to train the model. That is, movies and users that are present in the training data must be present in the test data. From here, analysis including movie, genre, and user effects was straightforward.

The author noticed that in the class example, the User Effect was layered on top of the Movie Effect and was never calculated on its own. The current analysis includes a stand alone calculation of the movie, user, and genre effects in order to compare the size of each effect separately. The terms are then layered in different orders to determine the effect of calculating corrections in different orders.

The different correction terms were then regularized. Regularization is an analytic technique to constrain the total variability of the effect sizes. It does this by introducing a correction term λ to the effect terms b.

$$b_x(\lambda) = \frac{1}{\lambda + n_x} \sum_x (y - \mu) \,,$$

where x is any of the effects. Regularization can be calculated on any of the previous effects separately or on all of them. Regularizing the movie, user, and genre effects separately gives an idea of what the impact on each is.

A total of 13 models are compared:

number	method
	Naive Model
1	Movie Effect
2	User Effect
3	Genre Effect
4	Movie + User Effects Model
5	User + Movie Effects Model
6	Movie + Genre Effects Model
7	User + Genre Effects Model
8	Movie + User + Genre Effects Model
9	Movie Regularization Model
10	User Regularization Model
11	Genre Regularization Model
12	Grouped Regularization Model
13	Individual Regularization Model

Example code to calculate correction from just one effect (the movie effect):

```
movie_avgs1 <- train_set %>%
  group_by(movieId) %>%
  summarize(b_m = mean(rating - mu))

predicted_ratings_m <- mu + test_set %>%
  left_join(movie_avgs, by='movieId') %>%
  .$b_m

rmse <- RMSE(predicted_ratings_m, test_set$rating)
```

The user and genre effect corrections were calculated similarly.

Example code to calculate corrections for two effects (movie followed by user):

```
user_avgs <- test_set %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mu - b_m))

predicted_ratings_mu <- test_set %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  mutate(pred = mu + b_m + b_u) %>%
  .$pred

rmse <- RMSE(predicted_ratings_mu, test_set$rating)
```

Other paired effects were calculated similarly.

Code to calculate the Movie + User + Genre Effect Model:

```
genre_avgs <- test_set %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  group_by(genres) %>%
  summarize(b_g = mean(rating - mu - b_m - b_u))

predicted_ratings_mug <- test_set %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  left_join(genre_avgs, by='genres') %>%
  mutate(pred = mu + b_m + b_u + b_g) %>%
  .$pred

rmse <- RMSE(predicted_ratings_mug, test_set$rating)
```

Example code to calculate regularization effects:

```
lambdas <- seq(0, 10, 0.25)

# calculate the sum and count of ratings to build to the regularized average
just_the_sum <- train_set %>%
  group_by(movieId) %>%
  summarize(s = sum(rating - mu), n_m = n())
# now calculate rmse as a function of regularization factor, lambda
rmse_lambda <- sapply(lambdas, function(lam){
  predicted_ratings <- test_set %>%
    left_join(just_the_sum, by='movieId') %>%
    mutate(b_m = s/(n_m+lam)) %>%
    mutate(pred = mu + b_m) %>%
    .$pred
  return(RMSE(predicted_ratings, test_set$rating))
})

lambda_m <- lambdas[which.min(rmse_lambda)]

# use lambda to correct for (regularize) movie effect term b_m1
```

```
movie_avgs_r <- train_set %>%
  group_by(movieId) %>%
  summarize(b_mr = sum(rating - mu)/(n()+lambda_m), n_m = n())

rmse <- rmse_lambda[which.min(rmse_lambda)]
```

Results

In the class exercise, validating the training set and the validation set reduced the number of records in the validation set, indicating removal of records in the validation set for which there were not corresponding users or movies in the training set. When performing the same steps on the validation data provided, it becomes clear that the data have been pre-cleaned to ensure that movies, genres, and user IDs match between training and validation data.

The improved RMSEs for just movies, just users, and just genres are:

method	RMSE
Naive Model	1.0603
1. Movie Effect Model	0.9439
2. User Effect Model	0.9783
3. Genre Effect Model	1.018

Inclusion of each individual effect is an improvement over the Naive Model. However, it is clear that some effects are stronger than others. The movie effect is stronger than the user effect, and both are stronger than the genre effect. If one were to stop at a first order correction, the Movie Effect Model would be the best choice.

The next order corrections come from combining two or more effects. The results of combining two effects are:

method	RMSE
4. Movie + User Effects Model	0.8292
5. User + Movie Effects Model	0.8767
6. Movie + Genre Effects Model	0.9435
7. User + Genre Effects Model	0.9422

Models 4 and 5 make it clear that the order of calculating correction terms matters. Calculating the user effect per movie results in a better prediction than the other way around. Adding in a genre effect made almost no different to the Movie Effect model (Model 1) and a small difference to the User Effect Model (Model 2). The next level of corrections is to add a genre effect to Model 4, the Movie + User Effect model.

method	RMSE
8. Movie + User + Genre Effects Model	0.8285

Adding the genre effect in had a small impact, as one might have expected from looking at the impact of genre alone and of genre on either movie or user separately.

The next effects calculated were of regularization. The effect of λ on RMSE is visualized in Figures 1–4.

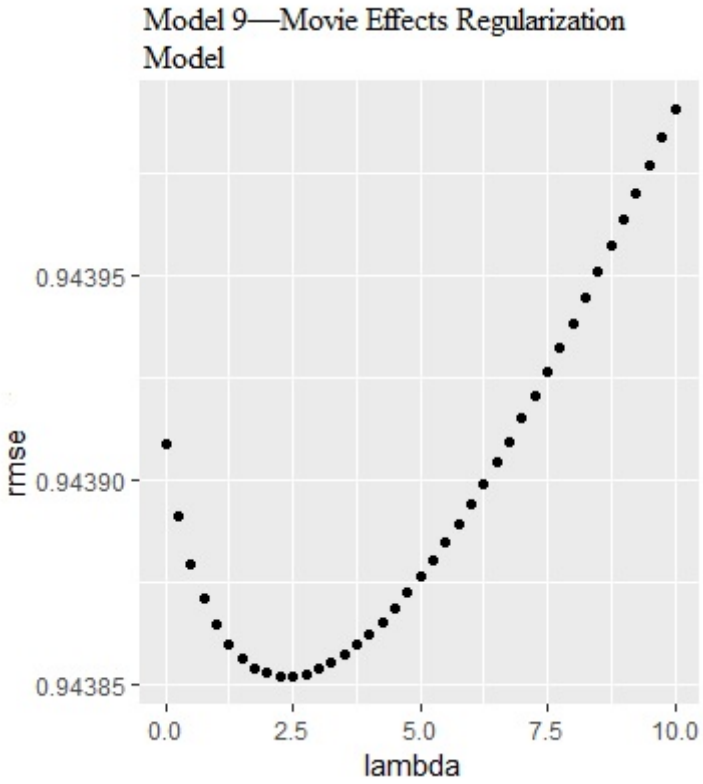

Figure 1: RMSE vs. λ for regularization of the movie effect

method	RMSE
9. Movie Effects Regularization Model	0.9439
10. User Effects Regularization Model	0.9779
11. Genre Effects Regularization Model	1.018

The additional impact of regularization on RMSE for each individual effect can only be seen in the fourth decimal place. Of course, the greatest impact to RMSE comes from combining effects. We expect that combining regularization with a combined effect model will lead to the greatest improvement in RMSE. We take two approaches: find a single λ that regularizes all effects and use individualized λs on each effect separately.

method	RMSE
12. All Effects Regularization Model–Single lambda	0.8644
13. All Effects Regularization Model–Multiple lambdas	0.8645

The impact of regularizing all effects on RMSE is around 8%. The difference between using a single λ to calculate all the effects vs. using the optimal λ for each effect is negligible. However, regularization actually increases the RMSE over the non-regularized models. When reviewing all the models, it is clear that the Movie + User + Genre Effects Model without regularization is optimal, showing approximately 18% improvement over the Naive Model.

Conclusion

The optimal data-based model to predict movie ratings corrects for popularity of movies (movie effect), ratings patterns of users (user effect), and popularity of genres (genre effect) and does not include regularization. This model provides an RMSE of 0.8285. See https://github.com/lizzjudge/HarvardX-PH125.9x-Data-Science-Capstone.git

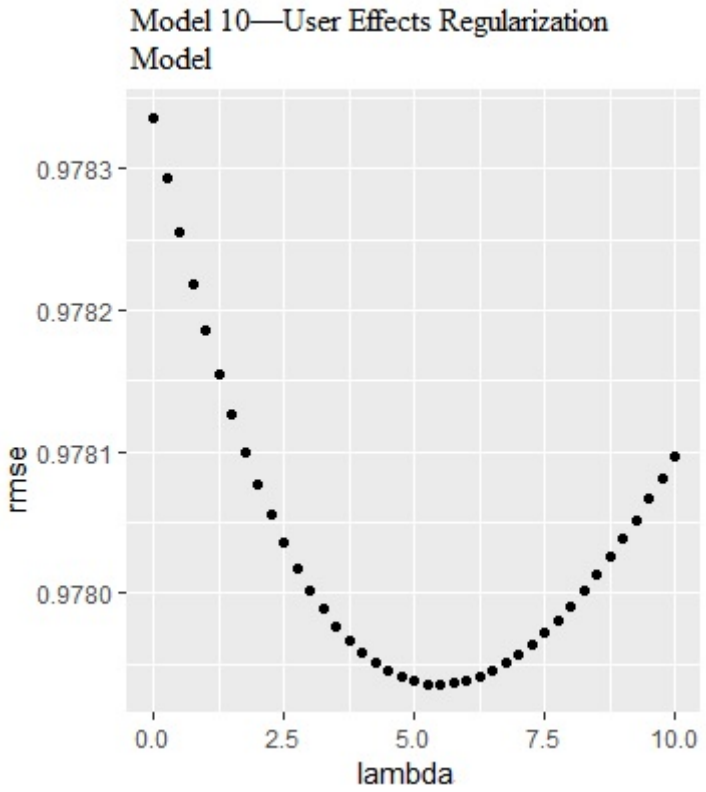

Figure 2: RMSE vs. λ for regularization of the user effect

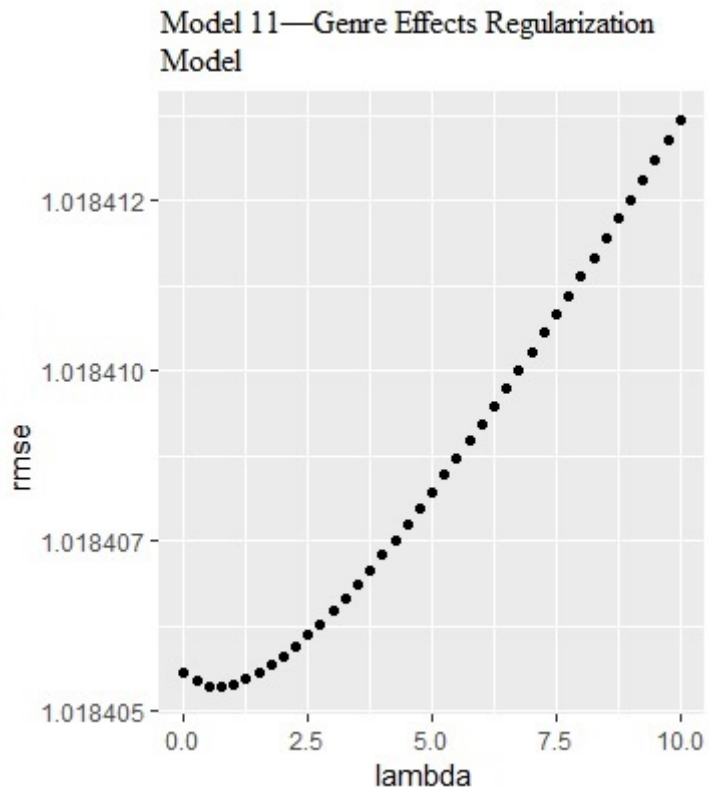

Figure 3: RMSE vs. λ for regularization of the genre effect

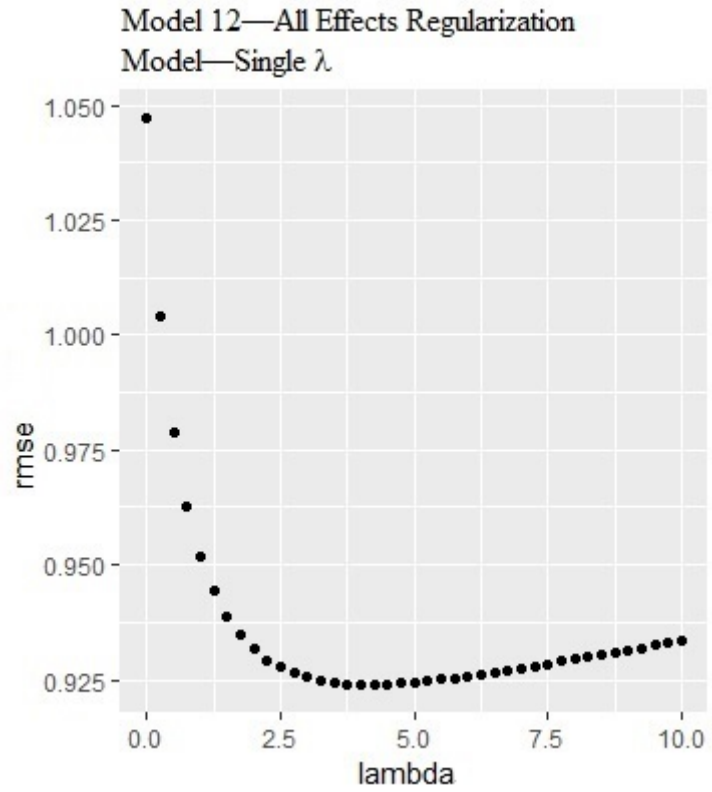

Figure 4: RMSE vs. λ for regularization of all effects with a single value of λ