Regularization of User, Movie, and Genre in Movie Recommendation

Roland Bennett

12/12/2019

Introduction

In this report I used the R programming language to develope an algorithm that computes a prediction for how a user will rate movies on a scale from 0 to 5. The algorithm is developed incrementally; the first model simply takes the average rating assigned to a movie and uses that as the prediction. The fifth and final model incorporates and regularizes effects due to users, movies and genres. This model achieved a root mean squared error (RMSE) of 0.85772, which achieves the goal of getting an RMSE of under 0.86490.

I used the "ml-10m.zip" data from http://files.grouplens.org/datasets/movielens/. This data consists of 10,000,054 ratings of 10,677 movies by a total of 69,878 users. The data is divided into three sets: a training set (edx_train) a test set (edx_test) and an evaluation set (validation). The algorithm was developed using the training set and then optimized using the test set. In order to prevent overtraining, the validation set was only used for computing the final RMSE.

Methods

The validation set consists of 10% of the data (999,999 ratings). The remaining data was split into 80% training data (7,200,096 ratings) and 20% testing data (1,799,959 ratings). Each data set was constructed so that every user and movie present in the test and validation sets were also present in the training set. Due to the incredible size of the training data (936 megabytes), no premade machine learning functions (such as random forest, knn, or regression) could be run in a timely manner on the entire data set. Functions such as principal componenct analysis (pca) also could not be executed on the entire dataset. I wanted to use the entire training set for training, rather than just a subset of it, so I opted to not use any such functions.

I did a preliminary assessment where I assigned the average of all ratings ($mu = 3.51$) as the prediction for all further ratings. This yielded an RMSE of 1.06049. The next method, Model 1, accounted for discrepancies between different movies. I calculated b_i, the average amount that each movie was above the average, mu, and added that to mu to get the prediction (b_i is negative for a movie whose average is below mu and therefore the prediction is less than mu). The RMSE improved to 0.94436.

The second model added a correction for user bias. I assumed that some users are more likely than others to give higher ratings. I created the term b_u for each user that is equal to the average amount that that user rated movies above each movie's average. The predicted rating is then equal $mu + b_u + b_i$. The RMSE for this model improved to 0.86668.

The third model was the first to incorporate regularization. If a movie is not rated many times, then the b_i value is not as as reliable and therefore should recieve less weight. The regularized equation for b_i is

$$Equation\ 1 : b_i = \sum_{n=1}^{N} \frac{rating - mu}{N + lambda}$$

for N ratings of movie i. The RMSE was calculated using values of lambda between 0 and 10 in increments of 0.25. The results are shown in Figure 1. The minimum RMSE value of 0.94429 was found when lambda was 2.5. This RMSE was higher than in the previous model; however, it did not include any user effects.

Model 4 regularized user and movie effects. b_u for was caluclated in a similar fashion to b_i:

$$Equation\ 2 : b_u = \sum_{n=1}^{N} \frac{rating - mu - b_i}{N + lambda}$$

for N rating by user i. Initially, in Model 4a, the RMSE's were calculated for each of the same values of lambda as before with the b_i term used before (lambda = 2.5) also included. This yielded a minimum RMSE of 0.86597 at lambda = 5. Now the algorithm was run again (Model 4b), but this time fixing $lambda(b_u)$ at 5 and retesting different values for $lambda(b_i)$. Now, an minimum RMSE of 0.86594 was found when $lambda(b_i) = 4.5$.

Model 5a was the first to include genre effects. Each movie is assigned one or more genres (there were some movies where the genre column was empty). I assigned this collection of genres to be the single genre for the movie. For example, if a movie is considered to be "Action, Adventure, and Sci-Fi," then it would be categorized as sharing a genre with and only with other movies that also have those exact genres. My algorithm would not recognize it as having any similarity to a movie with the genres "Action, Adventure, Sci-Fi, and Comedy" or "Action, Adventure, and Comedy." I initially calculated g_u values for each users-genre combination as

$$Equation\ 3 : g_u = \sum_{n=1}^{N} \frac{rating - mu - b_i - b_u}{N}$$

for N ratings by user u of genre g.

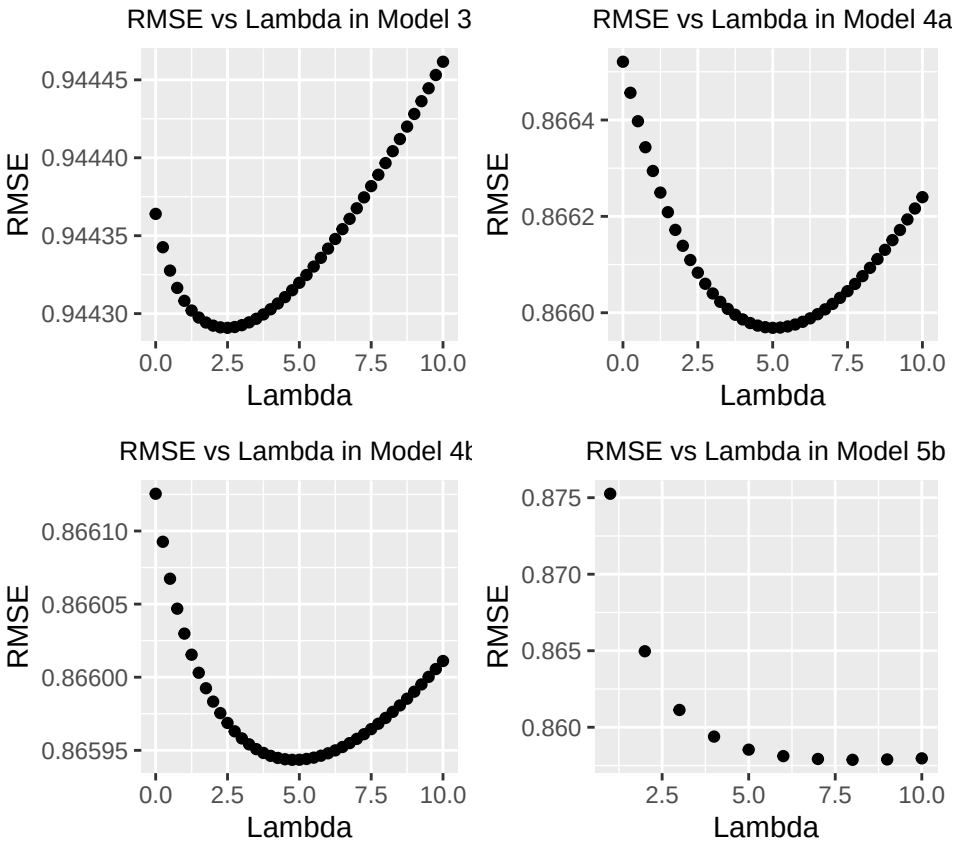

Figure 1: Scatter plots of RMSE vs Lambda in Models 3, 4a, 4b, and 5b

This model was improved upon using regularizarion in Model 5b. g_u was regularized by finding the RMSE

for values of $g_u(lambda)$ calculated using the following:

$$Equation\ 4 : g_u = \sum_{n=1}^{N} \frac{rating - mu - b_i - b_u}{N + lambda}$$

for each group with N ratings. Integer values between 0 and 10 were initally used for lambda. The minimum RMSE was found at lambda = 8. In Model 5c ran the algorithm again on lambdas from 7 to 9 with an interval of 0.1. Lambda = 8.1 yielded a slightly improved RMSE.

Results

The lowest RMSE achieved on any set was 0.56653 and occured when the non-regularized genre effects algorithm was run on the training set; however, this was due largely to overtraining. When the same algorithm was used on the test set, the resulting RMSE was 0.92187. There are many user-genre groups with only one or a few ratings. When there is only one rating in a group, the prediction will match the rating exactly when used on the same set used to train. The training set was not used to evaluate any of the models. The RMSE values for each model are shown in Table 1 below. The RMSE value reported for each model was calculated using the final version of each model.

Table 1: Resulting root mean aquared error (RMSE) of each model

method	RMSE
Just the Average	1.0604925
Model 1	0.9443640
Model 2	0.8666769
Model 3	0.9442909
Model 4	0.8659435
Final RMSE	0.8577153

Once regularization was applied to Model 5, the RMSE value dropped considerably to 0.85787. User u's rating of a movie i of genre g is calculated in the final algorithm using the equation

$$Equation\ 5 : Rating(u, i, g) = mu + b_i + b_u + g_u$$

With b_i as defined in Equation , b_u as defined in Equation , and g_u as defined in Equation . When this algorithm was then used on the validation set the final RMSE of 0.85772 was achieved.

Conclusion

This method of using regularization for predicting movie ratings proved quite proficient, as the final RMSE was below the target by 0.00718. A major concern is the high variability between the predicted rating and expected rating in user-genre groups with a small number of ratings in them. This is shown in Figure 2 below. This is due to how specific the genres are. A movie can only be used to predict the genre effect in movies with all of the same movie tags, which can be rather specific since a movie can have many tags. Many users only rate a single movie in a genre group. The more ratings in a user-genre group, the more reliable the g_u value is and therefore, generally, the lower the residual is. The first fifteen groups are shown in Table 2 below. Notice also that the groups defined by more ratings generally have fewer ratings and therefore contribute less to the overall RMSE. Recognizing correlations between movies with some but not all of the same genre tags would be a logical next step in improving this algorithm.

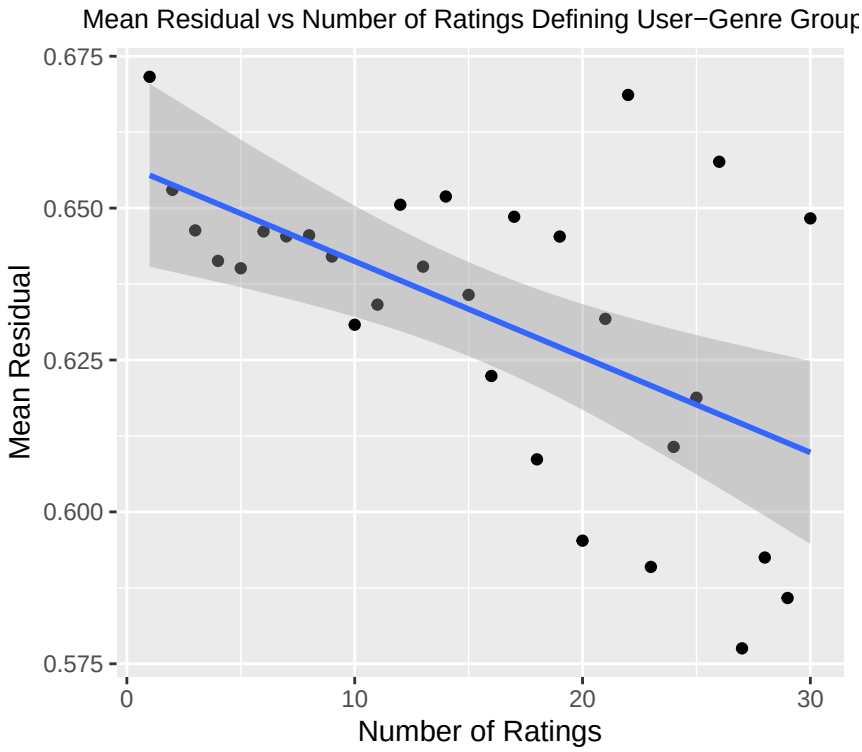

Figure 2: Line graph of relationship between size of user-genre group and mean residual for groups defined by less than 30 ratings

Table 2: Mean residual and size of different user-genre groups for first 15 groups

Number of Ratings Defining User-Genre Group	Size of Group	Mean Residual
1	655761	0.6716118
2	71636	0.6530058
3	20631	0.6463343
4	8951	0.6413138
5	4285	0.6401019
6	2628	0.6461655
7	1618	0.6453471
8	1077	0.6455279
9	819	0.6420227
10	572	0.6308020
11	423	0.6341128
12	296	0.6505605
13	244	0.6403632
14	199	0.6519121
15	139	0.6357179