A Movie Recommendation System

Claudia Valdeavella

14 January 2019

Introduction

The objective of this project is to create a movie recommendation system using a subset which consists of 10M ratings from the MovieLens dataset.

Each observation consists of 6 variables:

- userId
- movieId
- rating
- timestamp
- title
- genres

A couple of years ago, Netflix sponsored a contest that awarded a million dollars to the person or group who can improve the Netflix movie recommendation algorithm by 10%. The winning strategy for the estimation of the movie rating involves setting a baseline value, capturing the main effects from the movie and the user on the ratings, then having the model predict the remainder. While estimating the latter requires more sophisticated algorithms and modeling breakthroughs, capturing the main effects is straightforward.

This project is limited in scope to the contribution of the movie and the user to the estimation of the rating.

Analysis

The dataset consists of 10M observations, of which 10% was set aside as the validation set. The user+movie matrix is sparse and we do not want to set the missing elements to zero.

This project will follow the strategy of decomposing a rating into several parts:

- The baseline rating which is the overall average
- The movie effect which takes into account the fact that some movies may be more appealing than others
- The user effect which captures the fact that some users rate movies higher than others
- The specific user+movie interaction which accounts for the remainder of the rating

The overall average is the baseline value, 3.5.

Next we estimate the effect of the movie on the rating and write the model as

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

where b_i is the deviation of the movie rating from the average. A positive b_i means that the movie is liked better than average, with 1.5 corresponding to the highest possible rating. In the same manner, a negative b_i means that a movie has a below average rating.

The estimates, b_i,will be calculated as the mean of the difference, $Y_{u,i} - \mu$, for each movie. This is preferred over executing the lm() function for each movie, as this would have taken a lot of time.

In a similar manner, we will calculate the user effect on the rating. We will write the model as follows

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

and estimate b_u as the average over $Y_{u,i} - \mu - b_i$.

At this point, we can calculate the predicted ratings, including movie and user effects, but without the specific user+movie interaction.

Regularization

When a movie is rated infrequently, the average rating is unreliable as a measure of the true average. The magnitude of the b_i term can be large as a result of the rating of an individual user with strong feelings about a movie. Regularization is an approach that penalizes large b_i, which will be done next.

First, we will apply the concept to the estimation of the b_i's. The approach involves minimizing the following equation

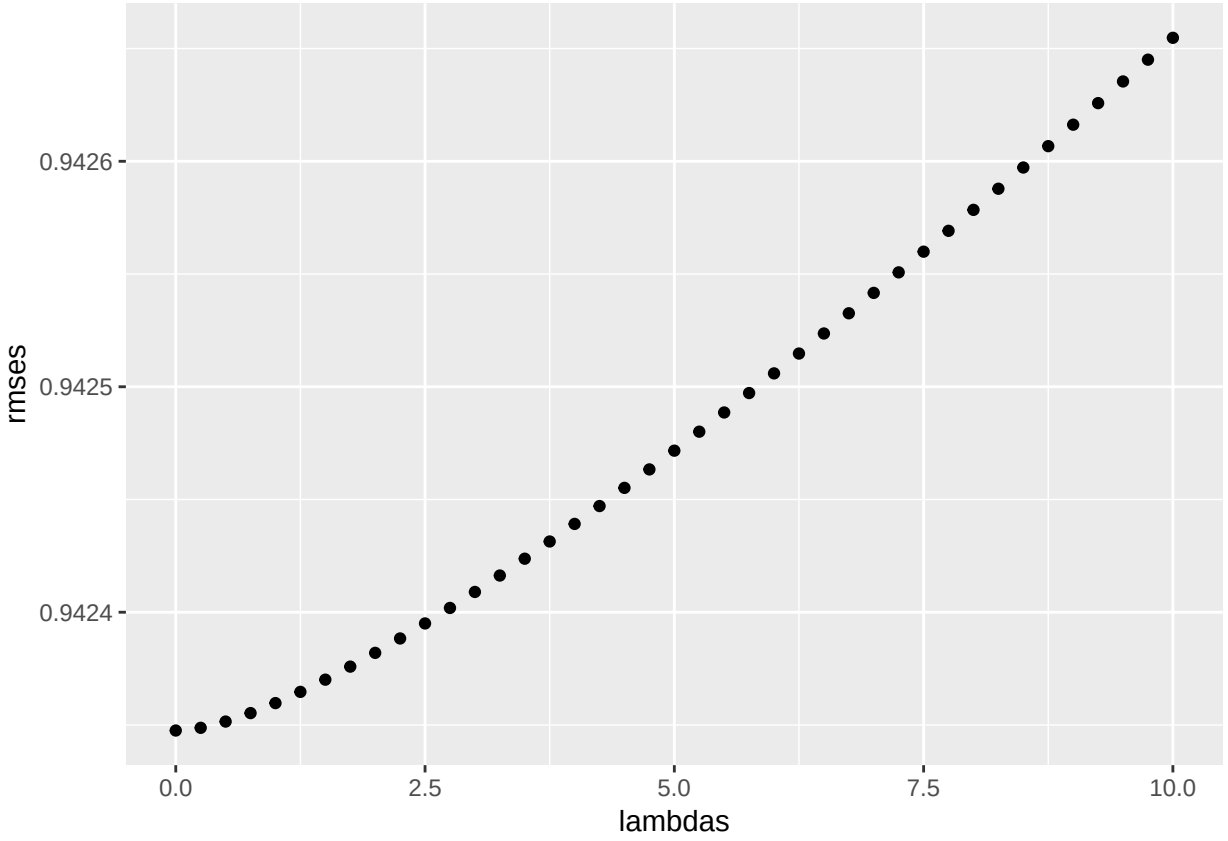
$$\frac{1}{N}\sum_{u,i}(y_{u,i} - \mu - b_i)^2 + \lambda \sum_i b_i^2$$

which yields the following estimate for b_i

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

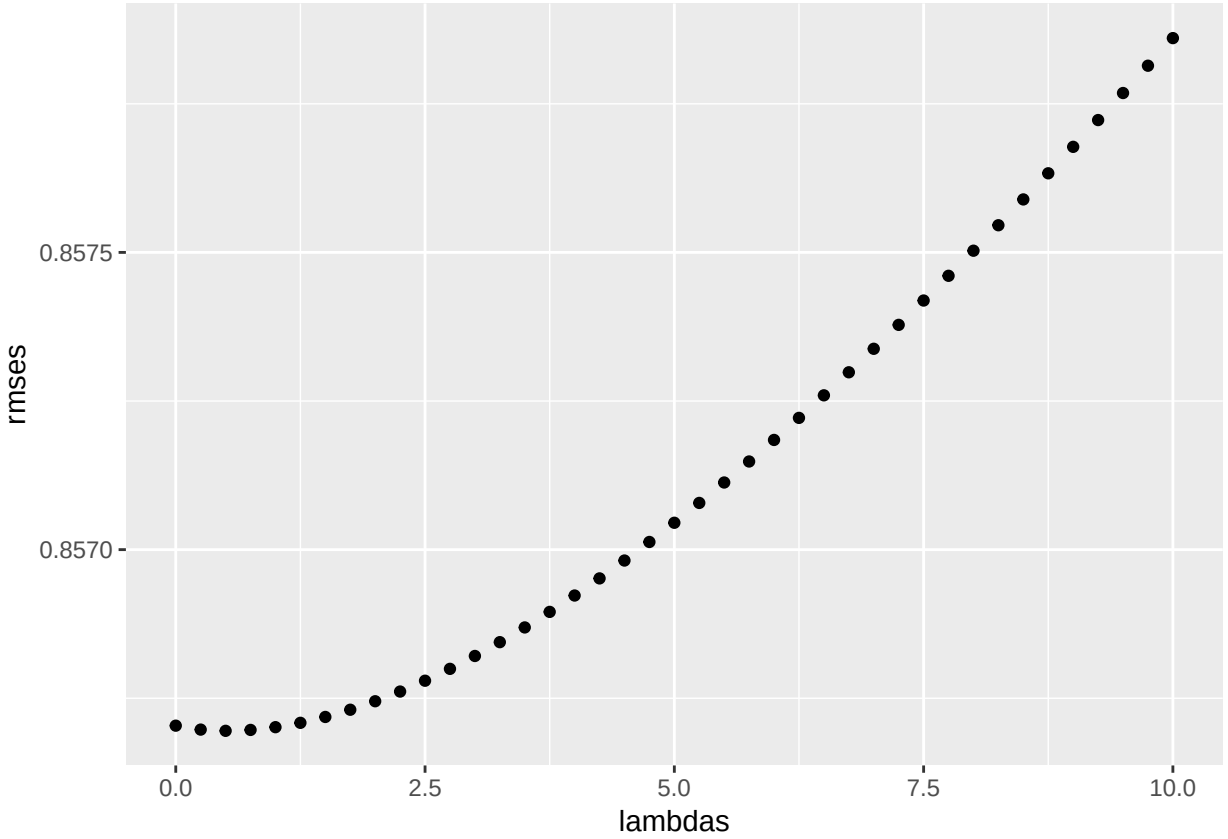
The effect of λ is minimized when n_i is large. Otherwise, the effect of λ is to shrink the $b_i(\lambda)$ to zero.

The optimum value of λ can be determined by cross-validation. For the model with the movie effects only



We see from the analysis that the optimum value for λ is zero, which implies that this correction is unnecessary for the above model.

Regularization can be done on the combined movie and user effects as well. Using cross-validation to find the optimum λ in this model



λ, in this case 0.5, is small but nonzero. The movie and user effects, can then be updated based on this value of λ.

As mentioned in the introduction, this analysis will not go into further enhancements of the model to predict the remainder term, $\epsilon_{u,i}$. The primary reason is that the author does not have the compute power to execute more sophisticated approaches.

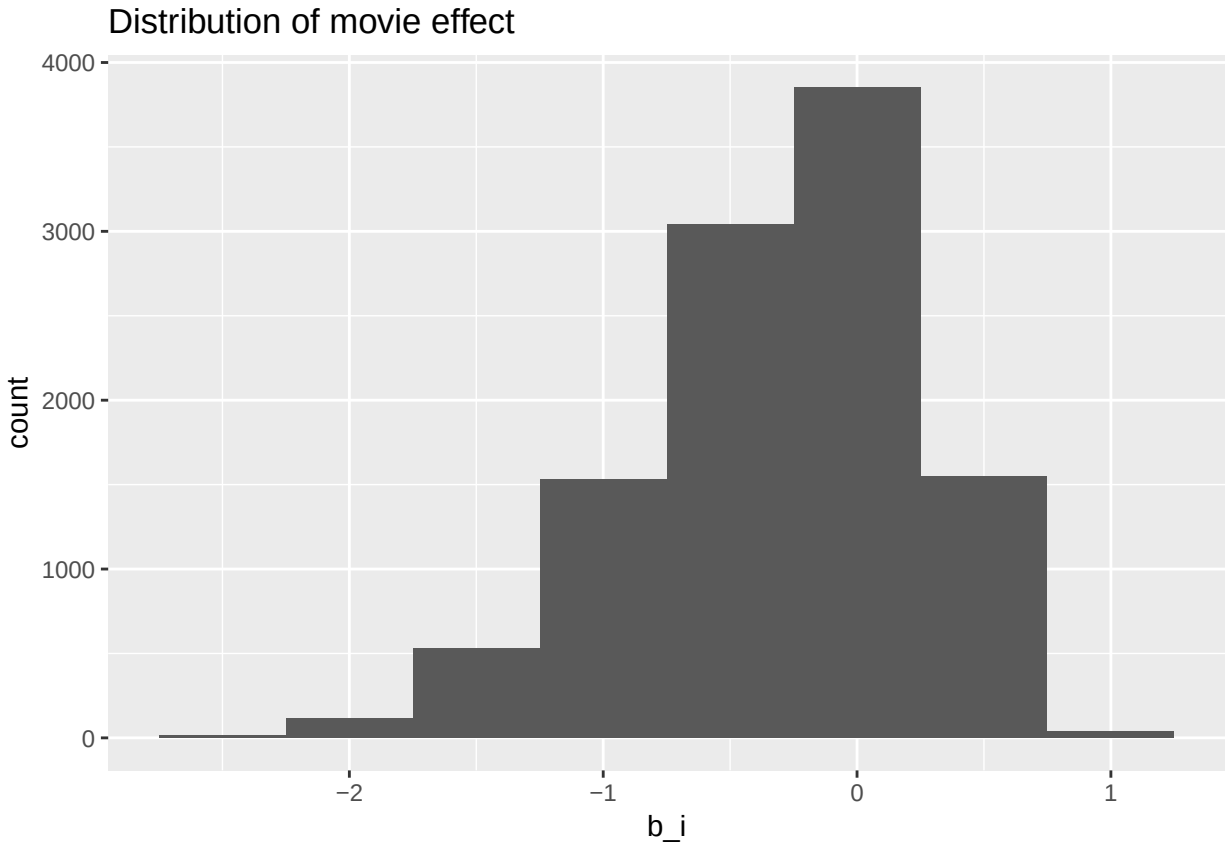
Results

The following table lists the basic models that were described in the previous section and the corresponding root mean squared errors.

method	RMSE
Baseline	1.0603
Movie Effect Model	0.9423
Movie + User Effects Model	0.8567
Reg. Movie + User Effects Model	0.8567

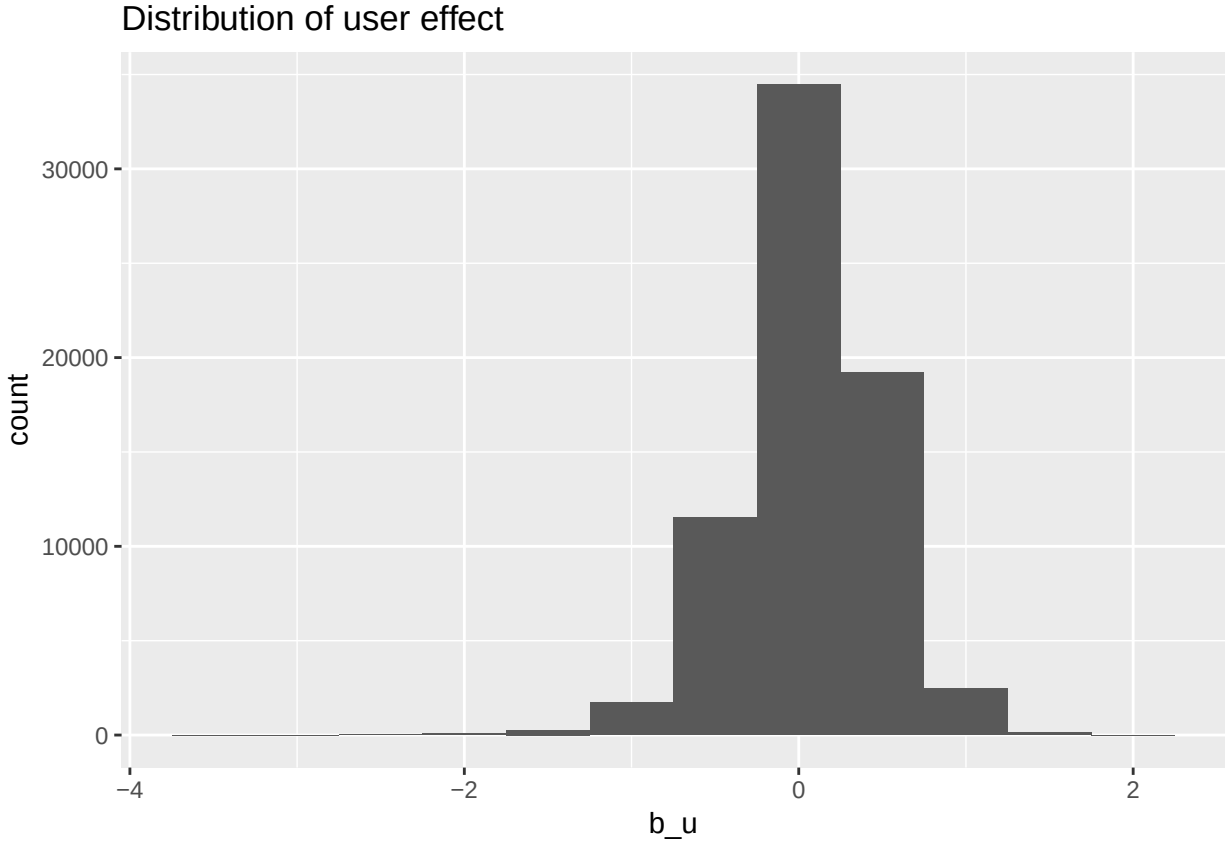
The above metrics were calculated during training. The rmse is frequently used as the metric for the quality of the fit. The results show that accounting for just the baseline predictors can significantly improve the fit.

The distribution of the b_i term is shown below



The movie effect varies substantially with the peak at 0 corresponding to the average movie rating.

Similarly, we can see from the distribution of the b_u term



that the user effect spreads throughout the entire range of possible ratings with the peak at the average.

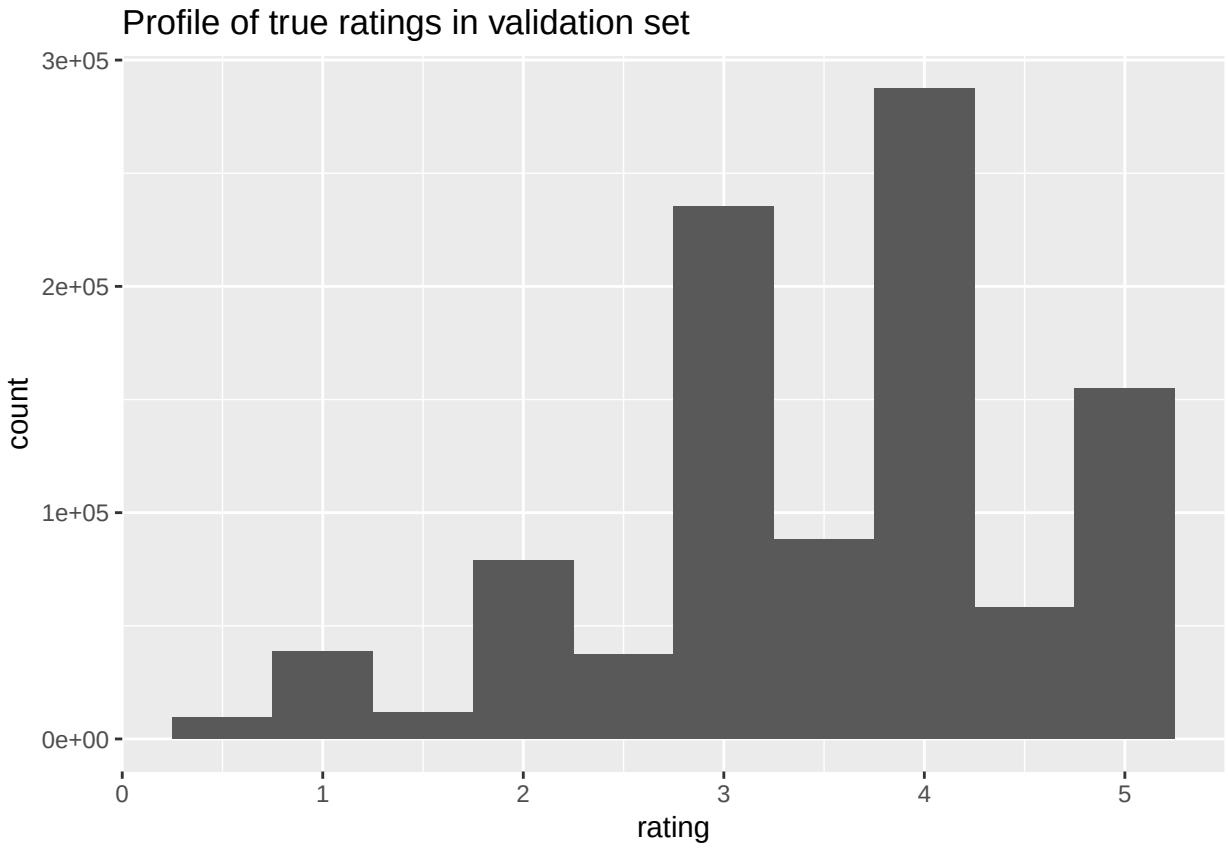
This project will be graded based on the rmse. The following table shows two metrics calculated for the validation set. The rmse is good but the accuracy on the predicted ratings for the validation set is poor.

metric	validation
rmse	0.8771
accuracy	0.2483

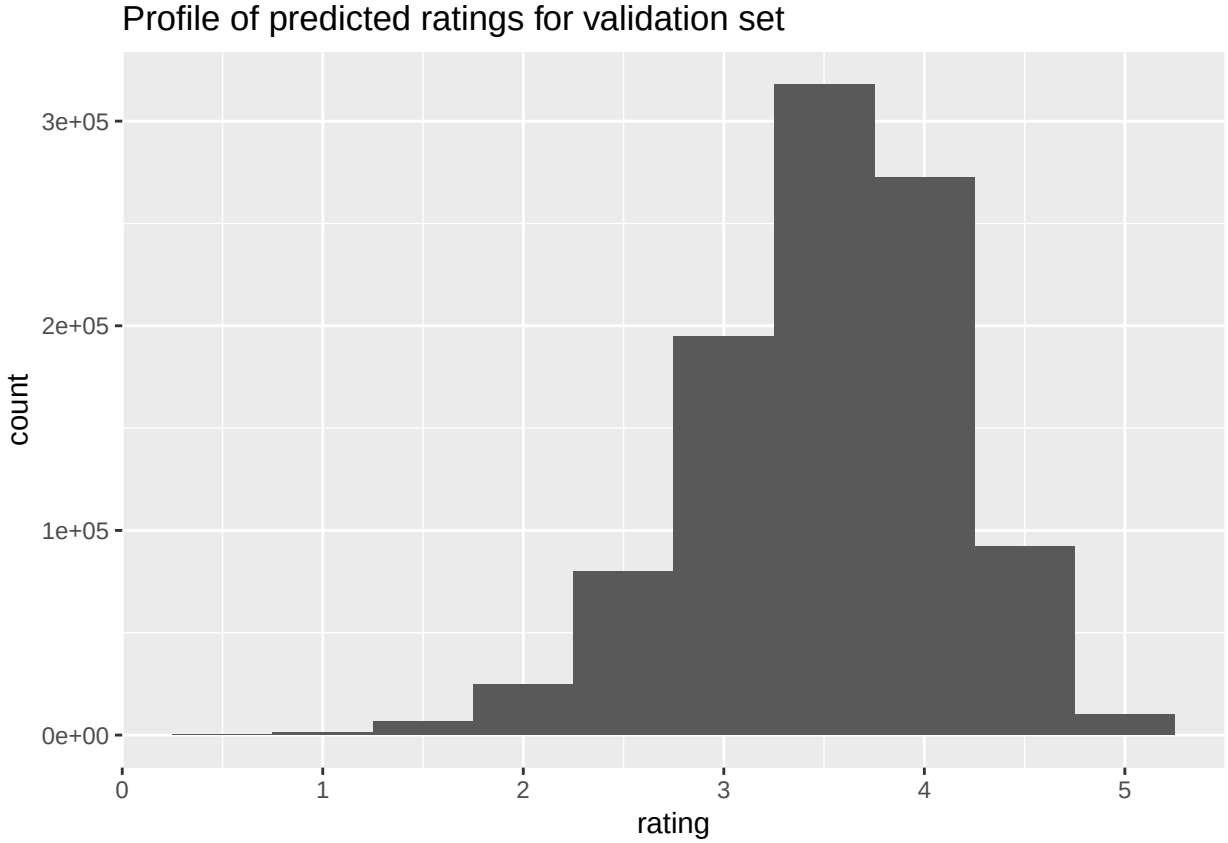
These metrics were calculated after the predicted ratings were rounded off to the nearest integer or half integer. Furthermore, the author applied the bounds on the ratings, that is, predicted ratings below 0.5 were set equal to 0.5 while predicted ratings above 5 were set to 5.

The rmse is a better metric than the accuracy to quantify the fit of the model to the data. The accuracy score does not take into account the magnitude of the deviation of the predicted rating from the true rating. If the true rating is 3.5 for example, then a prediction of either 4 or 5 counts the same.

A pattern that can be observed from the movie ratings is that half integer ratings are so much less than the integer ratings, for example, the number of 3.5 ratings are markedly less than 3 or 4 ratings.



The models that were generated here do not have a way of capturing this pattern.



The dataset has another variable, genres, that the author did not use in the models. There are 797 distinct values for this field, and upon examination, this field is a concatenation of all the genres into which a movie could be classified. There is much overlap between the field values for the genres field to be useful as a descriptor.

It seems that to improve the accuracy in the models, the specific user+movie interactions has to be factored into the model.

According to the literature, among the models that were submitted for the Netflix Prize, those based on matrix factorization are most accurate. The typical way to carry out matrix factorization is by SVD or singular value decomposition. However, since the ratings matrix (ie. users on the rows, movies on the columns of the matrix) is sparse and we do not want to set the missing elements to 0, we can't do the standard SVD as in linear algebra. The stochastic gradient descent method is the approach used to carry out SVD on a sparse matrix.

Matrix factorization would have allowed us to write the residuals as a sum of terms

$$Y_{u,i} = \mu + b_i + b_u + p_{u,1}q_{1,i} + p_{u,2}q_{2,i} + ... + p_{u,n}q_{n,i} + \epsilon'_{u,i}$$

There are additional parameters, p's and q's, to be estimated but the number of parameters is much reduced compared to the dimensions of the original user+movie matrix. The contribution to the residuals drop with succeeding terms, with the first one or two terms accounting for most of the variability in the data. Only after this step, is it possible to effectively declare that the remaining $\epsilon'_{u,i}$ term in the above equation represents random error.

Conclusion

A strategy for predicting how a specific user will rate a movie is presented. This strategy focused on calculating the contributions made by baseline predictors, that is, overall average, movie effect, and user effect, to the movie rating. This model has an rmse of 0.8772.

It is proposed that matrix factorization be done to capture specific movie+user interactions and improve the predictions.