Modeling Movie Recommendations

Andrew Hayes

10/3/2019

Contents

1 Overview

This is a rating prediction project based off of the Netflix Prize, in which participants were challenged to surpass Netflix's accuracy in predicting users' movie ratings on the test set. The goal of the competition and of this project was to minimize the prediction error, specifically the root mean-squared error (RMSE). I implemented a series of increasingly complex models on a subset of the MovieLens dataset and observe significant decreases in RMSE from a trivial, baseline algorithm.

2 Methods

2.0.1 Data Preparation

The 1 million ratings used for this project are a subset of the full 10 million rating MovieLens dataset. Using the script provided in class, the data was imported from flat files and split into training and test datasets called *edx* and *validation* respectively. Users or movies that occured only in the test set were removed and added back to the training set, to avoid making predictions on net new movies or users.

2.0.2 Measuring Effectiveness

The main measure of effectiveness across various models was root mean squared error (RMSE) between actual ratings and predicted ratings. Other aspects of the ratings, like the appropriateness of resulting movie recommendations, were also considered.

2.1 Models

The first three models - simple average, movie and user effects, and regularized movie and user effects - are implementations of the models demonstrated in the edx course *Data Science: Machine Learning.* The last three are exploratory and use the recommenderlab package.

2.1.1 Simple Average Model

As a baseline model, we minimize RMSE by predicting the global average. In other words, the model assumes that all movies have the same true rating and all variation is independent of movie i or user j:

$$r_{ij} = \mu + \epsilon_{ij}$$

2.1.2 Movie and User Effects Model

In this model we assume each movie and user introduces a deviation or "effect" from the global true rating. The movie effect m_i is the average deviation of movie i from the global mean, and allows for what we already know: that some movies are better than others (or at least perceived as better). The user effect u_j is the average deviation of user j from the average rating of each movie they watched, to adjust for the fact that some users rate movies on average more highly than others. Our resulting model is:

$$r_{ij} = \mu + m_i + u_j + \epsilon_{ij}$$
$$m_i = \frac{1}{n_i}\sum_k (r_{ij_k} - \mu) \qquad u_j = \frac{1}{n_j}\sum_k (r_{i_k j} - m_{i_k} - \mu)$$

where n_i and n_j represent the number of ratings that a movie or user has.

The movie and user effects model can generate predictions greater than 5 or less than .5, so out-of-range predictions were rounded to 5 or .5.

2.1.3 Regularized Movie and User Effects Model

An extension of the Movie + User Effects Model, the regularized extension recognizes that a simple mean to determine movie or user effects can have negative results in prediction on the test set due to small sample sizes in the training set. For example, if one and only one person rates movie i in the training set and that rating is a 5, that m_i is going to indicate a very good movie when it may not be. To regularize for small sample sizes, we introduce terms λ_m and λ_u when calculating effects to reduce them disproportionately for small samples:

$$m_i = \frac{1}{n_i + \lambda_m}\sum_k (r_{ij_k} - \mu) \qquad u_j = \frac{1}{n_j + \lambda_u}\sum_k (r_{i_k j} - m_{i_k} - \mu)$$

The values of the λ's were determined by minimizing RMSE on the training set with cross-validation ($k = 10$). When creating the folds, users or movies who appeared in the test folds but not their respective training sets were removed to the training sets, just as was done with the actual test set during data preparation. After applying the model to each fold, the mean of the 10 RMSE's was calculated and minimized with Nelder-Mead, R's default optimization algorithm.

2.1.4 Recommender Lab: UBCF, IBCF, and SVD

The recommenderlab R package was used to build more personalized models. The three previous models, while they attempt to minimize the RMSE, cannot be used for personalization. The two effects models differentiate movies but only insofar as one is more highly rated than another, not that Person A might prefer Movie X while Person B might prefer Movie Y. **Note:** These three approaches were not finely tuned, and the only tuning that took place was on the *test* set. This was due to the computational intensiveness of these approaches which makes testing them a very lengthy process. Truly training these models with tune grids and cross-validation would require a more powerful machine in a hosted environment.

For all three recommenderlab models, the data was prepared in the same way. After normalizing and regularizing ratings, the residuals from the 1000 most rated movies and the 1000 users with the most ratings among those movies were used to form a 1000×1000 ratings matrix. A second matrix of the 1000 most rated movies and *any* user who had rated *any* of those movies was also created (69871×1000), to input to recommenderlab's predict function which attempts to fill in the unknowns in a provided ratings matrix.

A Singular Value Decomposition (SVD), Item Based Collaborative Filtering (ICBF), and User Based Collaborative Filter (UCBF) model were all trained on the 1000×1000 matrix. Then, those models were used to predict missing ratings in the 69871×1000 matrix. Because Recommender Lab only predicts ratings for items (here movies) it has already seen, they did not make predictions for all ratings in the test set. In the case of ICBF and UCBF, sometimes no ratings were predicted even for movies included in the model when not enough user or movie data was available. Results from the "Regularized Movie and User Effects Model" were interpolated to make a full prediction. All of the recommenderlab modeling can be thought of as an extension, not an alternative the previous model. The recommenderlab predict function is not aware of the limits on our rating scales of 0.5 - 5.0, so ratings outside of that range were adjusted to the border. For more information on the algorithms involved, see the paper from one of recommenderlab's creator, Michael Hahsler. Singular Value Decomposition was also covered in the edx course *Data Science: Machine Learning*.

3 Results

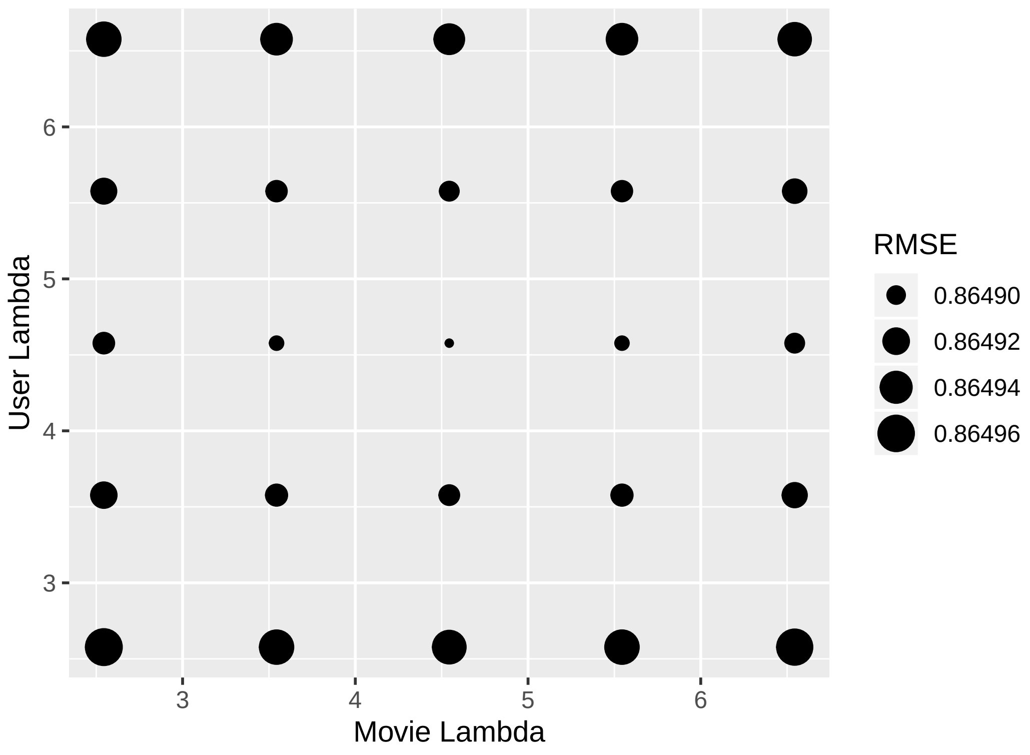
The RMSE's for the various models are:

Name	RMSE
Simple Average	1.06120
Movie + User Effects	0.86516
Regularized Effects (MUFR)	0.86471
MUFR + UBCF	0.86470
MUFR + IBCF	0.85295
MUFR + SVD	0.85019

3.1 Including Effects

The largest improvement by far is from the simple average model to the movie and user effects model. Clearly both movies and users have substantial effects that we benefit by accounting for.

The improvement from regularization was much smaller, less than 10^{-2}. Lambda was optimized via cross-validation at $\lambda_m = 4.54399$ and $\lambda_u = 4.5771$. The tuning is visualized here with a tuning grid constructed after the fact, centered at the optimal value:



You can see that the RMSE was relatively more senstive to λ_u than λ_m and the difference between all choices shown are all small, well within $< 10^{-4}$. The regularization did, however, have an enormous impact on our movie rankings. See the top 20 movies for each model, and the bottom 20 movies for each model, in the tables below. Without regularization, most of the best and worst movies have only a few ratings. With regularization many more familiar movies rise to the top of the charts.

We can also visualize the shift in movie effects by plotting the effects against the number of ratings. Initially there is a large spread in movie effects near $n = 0$, which is reduced with regularization.

Table 1: Best Movies w/o Regularization

	Title	n
1	Hellhounds on My Trail	1
2	Satan's Tango (Sátántangó)	2
3	Shadows of Forgotten Ancestors	1
4	Fighting Elegy (Kenka erejii)	1
5	Sun Alley (Sonnenallee)	1
6	Blue Light, The (Das Blaue Licht)	1
7	Who's Singin' Over There? (a.k.a. Who Sings Ove...	4
8	Human Condition II, The (Ningen no joken II)	4
9	Human Condition III, The (Ningen no joken III)	4
10	Constantine's Sword	2
11	More	7
12	I'm Starting From Three (Ricomincio da Tre)	3
13	Class, The (Entre les Murs)	3
14	Mickey	1
15	Demon Lover Diary	1
16	Life of Oharu, The (Saikaku ichidai onna)	3
17	Valerie and Her Week of Wonders (Valerie a týde...	1
18	Testament of Orpheus, The (Testament d'Orphée)	1
19	Power of Nightmares: The Rise of the Politics o...	4
20	Kansas City Confidential	1

Table 2: Best Movies w/ Regularization

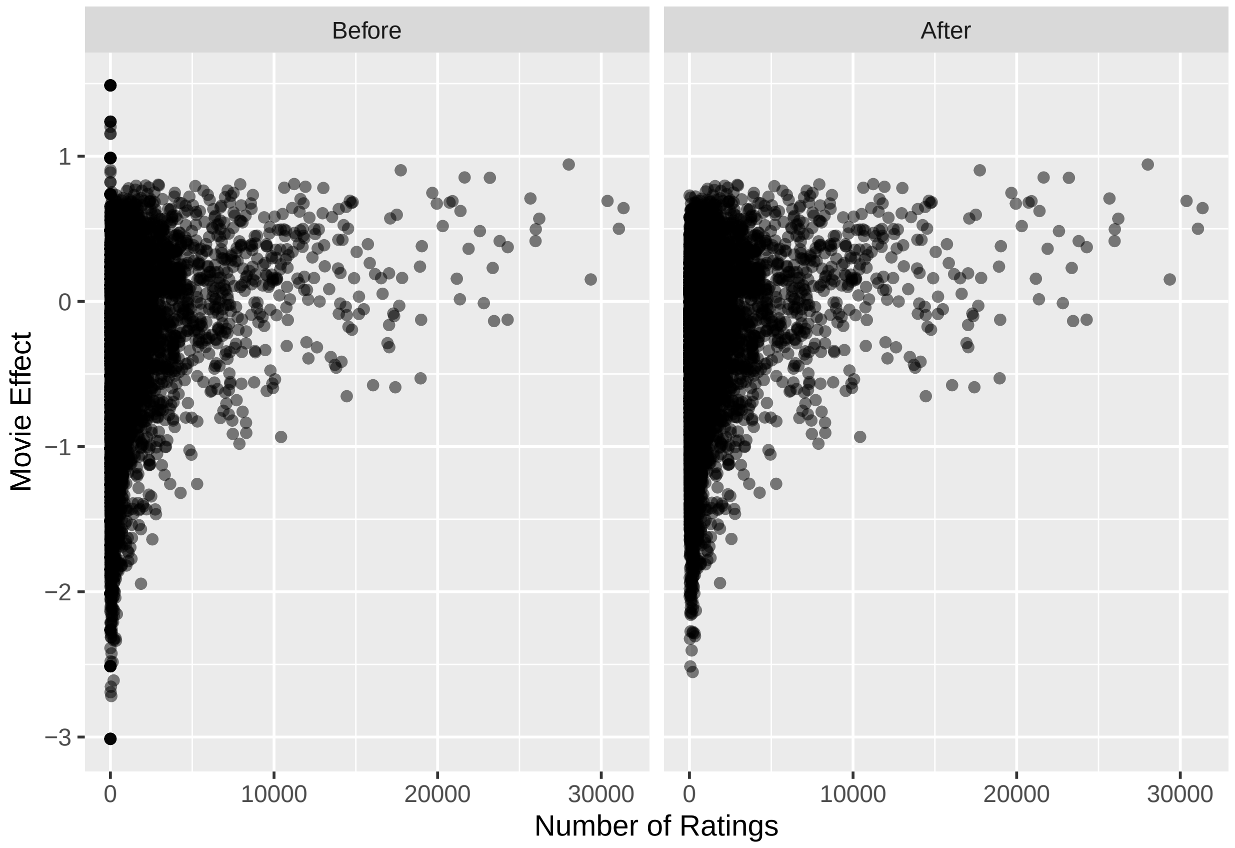
	Title	n
1	Shawshank Redemption, The	28015
2	Godfather, The	17747
3	Usual Suspects, The	21648
4	Schindler's List	23193
5	Casablanca	11232
6	Rear Window	7935
7	Sunset Blvd. (a.k.a. Sunset Boulevard)	2922
8	Third Man, The	2967
9	Double Indemnity	2154
10	Paths of Glory	1571
11	Seven Samurai (Shichinin no samurai)	5190
12	Godfather: Part II, The	11920
13	Dark Knight, The	2353
14	Dr. Strangelove or: How I Learned to Stop Worry...	10627
15	One Flew Over the Cuckoo's Nest	13014
16	Lives of Others, The (Das Leben der Anderen)	1108
17	Yojimbo	1528
18	Wallace & Gromit: The Wrong Trousers	7167
19	Wallace & Gromit: A Close Shave	5690
20	M	1926

Table 3: Worst Movies w/o Regularization

	Title	n
1	Carnosaur 3: Primal Species	68
2	Roller Boogie	15
3	Pokémon Heroes	137
4	Criminals	2
5	Mountain Eagle, The	2
6	Stacy's Knights	1
7	Dog Run	1
8	Monkey's Tale, A (Les Château des singes)	1
9	When Time Ran Out... (a.k.a. The Day the World ...	1
10	Dischord	1
11	Relative Strangers	1
12	From Justin to Kelly	199
13	Disaster Movie	32
14	Hip Hop Witch, Da	14
15	SuperBabies: Baby Geniuses 2	56
16	Besotted	2
17	Hi-Line, The	1
18	Accused (Anklaget)	1
19	Confessions of a Superhero	1
20	War of the Worlds 2: The Next Wave	2

Table 4: Worst Movies w/ Regularization

	Title	n
1	Faces of Death 4	85
2	Alone in the Dark	177
3	Prom Night IV: Deliver Us From Evil	90
4	Pokémon 3: The Movie	239
5	Carnosaur 2	92
6	House of the Dead, The	209
7	Turbo: A Power Rangers Movie	394
8	Faces of Death 6	79
9	Faces of Death: Fact or Fiction?	58
10	Son of the Mask	165
11	Yu-Gi-Oh!	80
12	Carnosaur 3: Primal Species	68
13	Barney's Great Adventure	208
14	Pokemon 4 Ever (a.k.a. Pokémon 4: The Movie)	202
15	Gigli	313
16	Glitter	339
17	Disaster Movie	32
18	Pokémon Heroes	137
19	SuperBabies: Baby Geniuses 2	56
20	From Justin to Kelly	199



3.2 Personalization with Recommender Lab

Recommending the best rated movies is valuable in itself, but ideally recommendations are personalized. The recommenderlab package enabled some personalization and reduced RMSE substantially. **Reminder:** Tuning for these models, what little did take place, was on the test set for expediency and should be interpreted with a grain of salt.

3.2.1 Tuning and RMSEs

UBCF: UBCF was the only algorithm that did not significantly outperform MUFR. Its results were improved by calculating distance between users with Pearson's rather than Cosine's. UBCF is commonly used and known as a top recommendation algorithm, but it is more memory and computationally intensive than SVD or IBCF when making predictions. It took much longer to run and likely needs to be scaled up to be effective.

IBCF: The IBCF model outperformed the Regularized Movie and User Effects Model (MUFR). In its default settings IBCF performed quite poorly, but was improved greatly by a) calculating distance with Pearson's rather than Cosine's and b) dramatically increasing k from the default 30 to 400. k in this algorithm indicates the number of movies similar to movie i whose ratings are considered when predicting a rating for movie i.

SVD: The SVD model also outperformed MUFR. The only tuning that took place was to increase k from 10 to 20, where k is the number of most significant vector pairs used to estimate the residual. Increasing k beyond that yielded some reduction in RMSE but not much, suggesting that the higher-numbered primary components become less relevant and represent noise rather than actual user properties.

3.2.2 Changed Recommendations

Like with regularization, we can also see a change in movie rankings in addition to RMSE. The new models will recommend different top movies (from the top 1000 movies) for each person from the list of movies

Table 5: Top 5 recommendations for User 368 across algorithms.

	MUFR	UBCF	IBCF	SVD
1	Shawshank Redempt...	Shawshank Redempt...	Shawshank Redempt...	One Flew Over the...
2	Schindler's List	Godfather: Part I...	Rear Window	American Beauty
3	Rear Window	Pulp Fiction	One Flew Over the...	Rear Window
4	Sunset Blvd. (a.k...	Rear Window	Wallace & Gromit:...	Schindler's List
5	Third Man, The	Third Man, The	Schindler's List	Léon: The Profess...

Table 6: Top 5 recommendations for User 2 across algorithms.

	MUFR	UBCF	IBCF	SVD
1	Shawshank Redempt...	Usual Suspects, The	Schindler's List	Shawshank Redempt...
2	Godfather, The	Godfather, The	Pan's Labyrinth (...	Godfather, The
3	Usual Suspects, The	Casablanca	Fargo	Schindler's List
4	Schindler's List	Godfather: Part I...	Godfather, The	Godfather: Part I...
5	Casablanca	One Flew Over the...	Deliverance	Usual Suspects, The

they have not rated. In tables 5 and 6, the top five recommendations for two users are arranged by model. Most of the recommendations are familiar from the overall top 20 movies. The models as tuned reorder recommendations but often respect the overall ranking of movies - in another implemention, this could be changed by weighting the predicted residual higher than the movie effect. In Table 7, we can see that for the 250 users on which we have the most training data, the median movie rank of their recommendations increases with the personalized models as expected.

Table 7: Median movie ranks of top 5 recommendations for 250 users

	MUFR	UBCF	IBCF	SVD
1	1	3	14	2
2	3	5	16	4
3	4	6	23	5
4	5	8	30	7
5	6	8	36	11

4 Conclusion

By modeling the differences in movies, user averages, and user preferences, we substantially decreased our RMSE. Modeling movie and user effects reduced the RMSE from a baseline 1.061 by over 18% to 0.8651. Regularizing effects granted a further reduction to 0.8647 and incorporated sample size into our movie rankings for better recommendations. The Regularized Effects model has the benefit of scaling well but unfortunately does not offer personalized recommendations. It would recommend all users the same top movies. A user could just as easily look up the top movie list on IMDb. There is an industry imperative to personalize recommendations based on the residuals of the Regularized Effects model.

An exploratory step in that direction was taken by using the recommenderlab package to test different models for user preferences. Substantial improvements in RMSE were seen with relatively small models and little tuning. The next step would be to test and tune with recommenderlab in a more robust fashion. First, the modeling should move to a hosted environment (rather than a local machine) with more computational power and in-memory capacity. Second, parameter tuning should take place on the training set via cross-validation rather than on the test set. Specifically, the parameters to tune are the number of movies in the model, the number of users in the model, and the parameters specific to each algorithm. Third, an ensemble of the various methods should be built and optimized to further reduce error.