Modelling Movie Ratings

Report for HarvardX PH125.9x: Data Science: Capstone

Samuel Bates

June 01, 2021

Overview

In this report, I describe my exploration of a dataset of movie ratings made by many people over the course of fourteen years, and my construction of several linear models for predicting ratings. The ratings were taken from the MovieLens website and made available by Grouplens Research.[1] The models were constructed from and tested on a training set containing 90% of the ratings. The remaining 10% of the ratings were held out in a validation set, which was not used until I had selected a final model among those I constructed. I report how well each model performed on the training set, and how well the final model succeeded in predicting the ratings in the validation set.

Exploration

The dataset contains 10,000,054 ratings of movies by people (hereinafter referred to as *users*). There are 10,677 different movies rated and 69,878 different users. The ratings were taken between January 9, 1995 and January 5, 2009. The dataset was provided as two files:

- `ratings.dat`, containing 10,000,054 observations, each with 4 variables: `userId`, `movieId`, `rating` (a number between 0 and 5), and `timestamp`;
- `movies.dat`, containing 10,681 observations, each with 3 variables: `movieId`, `title` (which contains the movie title and the year it was released), and `genres` (which contains a list of genres separated by `|` symbols).

Tables 1 and 2 show the results of counting how many movies each user rated. Half of all users rated 69 or fewer movies, and 9 out of 10 rated 335 or fewer movies. Every user rated at least 20 movies, and roughly 1% of the users rated more than 1000 movies.

[1] https://grouplens.org/datasets/movielens

Table 1: Number of movies rated by users

0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
20	24	31	40	52	69	94	129	195	335	7359

Table 2: Number of movies rated by high-volume users

90%	91%	92%	93%	94%	95%	96%	97%	98%	99%	100%
335	360	388	419	462	512	575.92	665.69	806	1058.23	7359

It is intuitively obvious that a movie's rating depends on the particular movie being rated and the particular user rating it. This intuition is reflected in the data. Figure 1 shows that there is a positive correlation between a movie's average rating and the number of times it was rated. A reasonable explanation is that more people will see more popular movies. Figure 2 shows that there is a negative correlation between a user's average rating and the number of movies rated by that user. One can imagine that such a user is harder to impress, having seen so many movies.

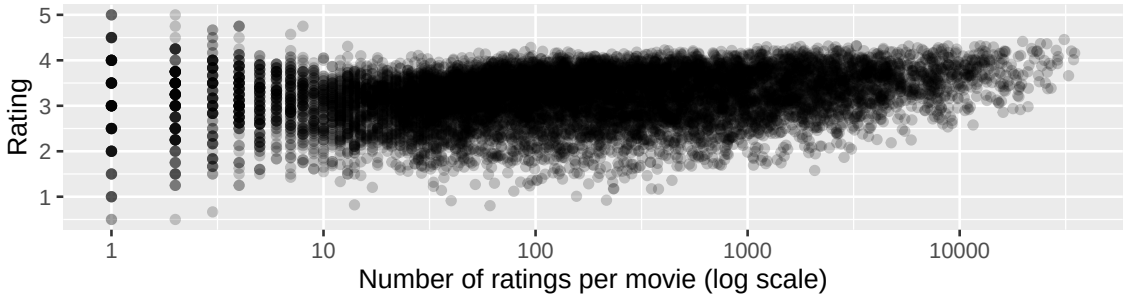

Figure 1: A movie's average rating increases as the number of ratings increases (correlation is 0.2129).

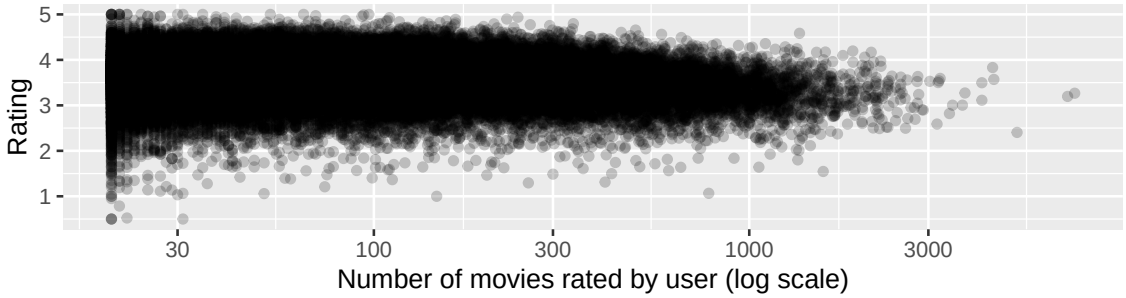

Figure 2: A user's average movie rating decreases as the number of movies rated increases (correlation is -0.1561).

Lastly, a movie's ratings may also change over time. Figure 3 shows how the average rating per month of the movie "Jerry Maguire" decreased over the 14 years after it was released.

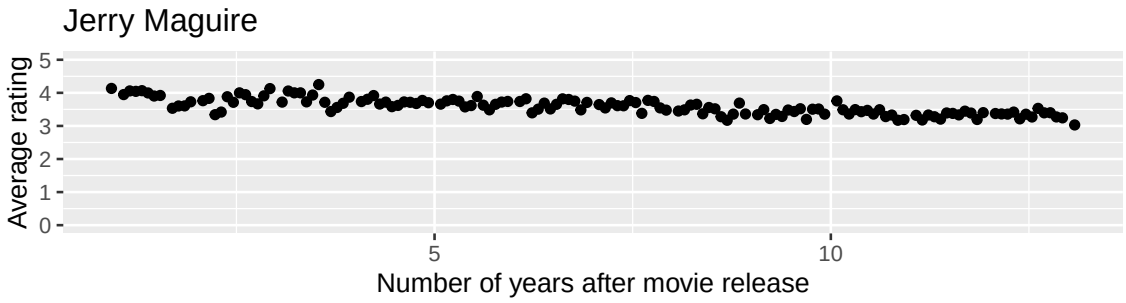

Figure 3: A movie's rating can change over time.

Methods and Analysis

Data Wrangling

The `movies.dat` table has two variables that were transformed in order to be used in models.

- The release year is contained in the `title` variable. This was separated into a new `year` variable in order to model the effect of time on ratings.
- The `genres` variable is a list of `|`-separated genres. There are 19 different genres, plus a "(no genre specified)" option. The variable was separated into 20 variables, each named for a genre. A genre variable contains a 1 if the movie has the variable name in its `genres` variable, and a 0 otherwise.

The `ratings.dat` table has a `timestamp` variable that is a second count from 12:00 am on January 1, 1970. It was converted into a `POSIXct` variable so that the time between a movie's release year and a rating's creation time can be calculated.

I created a training set containing 9,000,055 ratings and a validation set containing 999,999 ratings. All exploration and modelling was done exclusively on the training set.

My exploration was done in two phases. In the first phase, I tested models that excluded the genres of the movies. In the second phase, I subtracted the best model's predicted values from the actual values, and used the genres in a subsequent model.

Models without Genre

The models without genre that I created were all linear models on some subset of the variables `userId`, `movieId`, `year`, and `timestamp`.[2] I included the user and movie variables in all of them.[3] Rather than including the timestamp variable directly in a model, I calculated an age variable. The age is an approximate measure of how old the movie was when the user rated it. I calculated it as `(year(timestamp) + month(timestamp) / 13) - year`. I divided by 13 instead of by 12 so that the integral part of the age would equal the year of the rating (otherwise, ratings in December would

[2] I left out `title` as I could not think of a useful way to convert a title into a number that would make up part of a rating.

[3] Although the models are not included in the final code, I did create a model and generate an RMSE value for each pair of variables. If only one of the movie and user variables was in the model, the RMSE was at least .94.

have the integral part equal to the next year). Another advantage is that every age is non-zero. I chose the month for the fraction rather than the day or week of the year, as it exhibited a smoother rate of change.

Furthermore, I added regularization to different variables in the models. The movie variable was always regularized, since there are many movies with 4 or fewer ratings.

The result of these decisions was a set of eleven models:

- Two models with movie and user effects: both regularized or only movie effects regularized ($\hat{Y} = \mu + e_m + e_u$);
- Four models with movie, user, and year effects: all three regularized and user or year or both effects not regularized ($\hat{Y} = \mu + e_m + e_u + e_y$);
- Four models with movie, user, and age effects: all three regularized and user or age or both effects not regularized ($\hat{Y} = \mu + e_m + e_u + e_a$);
- One model with movie, user, year, and age effects, with only movie effects regularized ($\hat{Y} = \mu + e_m + e_u + e_y + e_a$). This model was added after seeing the results of the previous ten models.

Here, the constant μ is the average rating over all movies and all users. For each model, I calculated an optimal tuning parameter λ for the regularization and then averaged the results from 5-fold cross validation on the training set. I then chose the model that resulted in the lowest RMSE on the entire training set.

Models with Genre

The second phase was to explore the effects of genre after removing the best estimate from the first phase. I could have created another model using the `genres` variable directly, each combination of genres being a separate category. There are two problems with this approach:

- Genre combinations that share genres would be considered entirely separately. For example, a user's rating of a movide with genre "Comedy|Romance" would have no relation to the user's rating of a movie with genre "Adventure|Comedy|Romance," which we intuitively know is not likely.
- The model would perform no better than a non-genre model for a movie with a genre combination previously unrated by a user.

Therefore, I chose a linear combination of individual genre variables for the model. As described above, I separated the `genres` variable into 20 separate variables, one for each of the genres in Table 3. For display purposes, I abbreviated each genre to two characters. Each genre variable has a value of 1 if the genre appears in the movie's `genres` variable and has a value of 0 otherwise.

Mathematically, the model can be written as

$$\hat{Y} = \sum_{g=1}^{20} x_m^g e_g^u$$

Table 3: The twenty movie genres and their abbreviations.

Genre	Abbr.	Genre	Abbr.	Genre	Abbr.	Genre	Abbr.
Adventure	Ad	Romance	Ro	Horror	Ho	War	Wa
Animation	An	Drama	Dr	Mystery	My	Musical	Mu
Children	Ch	Action	Ac	Sci-Fi	SF	Film-Noir	FN
Comedy	Co	Crime	Cr	IMAX	IM	Western	We
Fantasy	Fa	Thriller	Th	Documentary	Do	(no genres listed)	No

where m is a movie, u is a user, and x_m^g is the genre variable described above. Constructing the model is then equivalent to calculating the term e_g^u, the effect of genre g on ratings by user u, for each user and genre.

I calculated the terms in two steps:

1. For each residual rating in the training set, I gave equal fractions to each of the movie's genres. Thus, if a movie with genres {Ad, Co, Ro} has a residual rating of 0.3, I assigned a rating of 0.1 to each of the three genres.
2. For each user, I gathered the fractional ratings calculated in Step 1 and computed the average for each genre. For example, if a user rated 3 movies with genres {Ad, Co} and 5 movies with genres {Co, Ro}, the Ad genre would get an average of 3 values, the Co genre would get an average of 8 values, and the Ro genre would get an average of 5 values.

The result is a matrix with 69,878 rows and 20 columns: 1 row for each user and 1 column for each genre. The rating for a movie is then computed by summing the values in a row that correspond to the movie's genres. When I tested this, the simple sum gave lower ratings than the actual ratings, so I introduced a tuning factor λ_2. I then calculated the RMSE for $\lambda_2 \cdot \hat{Y}$ for values of λ_2 in a range, and chose the value of λ_2 that produced the lowest RMSE on the training set.

To compute $\hat{Y}$ efficiently, I noted that the sum above can be rewritten as the dot product of the vectors $(x_m^1, \ldots, x_m^{20})$ and $(e_1^u, \ldots, e_{20}^u)$. The first vector is the genre categorization of movie m and the second vector is the row of the above matrix for user u. Therefore, multiple ratings can be calculated by computing the product of two matrices:

- Define M to be a matrix with a row for each movie to be rated, each row containing x_m^1 through x_m^{20}.
- Define U to be a matrix containing with a row for each user whose ratings are to be calculated, each row containing e_1^u through e_{20}^u. Row i of U is for the user whose rating is desired for the corresponding row i of M.
- The product $U \cdot M^\intercal$ has the average ratings on the diagonal. The diagonal entries are then multiplied by λ_2 to get the predicted ratings.

This is essential for timely calculation of ratings, since R's matrix algebra operators are much faster than multiplying two vectors and summing the result. I wrote code that calculates roughly 10,000 ratings at a time. Despite this optimization, calculating the genre-based ratings took almost 600 times as long as calculating the non-genre-based ratings.

Table 4: Results of non-genre models on the training set

Model name	λ	RMSE
Regularized(Movie + User) Effects	5.0	0.85705
Regularized(Movie) + User Effects	4.5	0.85670
Regularized(Movie + User + Year) Effects	4.6	0.85671
Regularized(Movie + Year) + User Effects	4.4	0.85638
Regularized(Movie + User) + Year Effects	4.6	0.85671
Regularized(Movie) + User + Year Effects	4.4	0.85638
Regularized(Movie + User + Age) Effects	5.0	0.85640
Regularized(Movie + Age) + User Effects	4.0	0.85609
Regularized(Movie + User) + Age Effects	5.0	0.85640
Regularized(Movie) + User + Age Effects	4.0	0.85609
Regularized(Movie) + User + Year + Age Effects	**4.0**	**0.85593**

Table 5: Result of combined models on the training set

Model name	λ	λ_2	RMSE
Regularized(Movie) + User + Year + Age Effects + Genre Effects	4	1.8	0.79548

Results

The models using only non-genre variables all returned very similar results. Table 4 shows each model name, the optimal tuning value λ, and the RMSE on the training set, with the lowest RMSE shown in bold. The following patterns are observed:

- *Regularizing the user variable reduces accuracy:* For each pair of models that differ only in whether the `user` variable is regularized (indicated by stripes), the model with the unregularized variable has a lower RMSE.
- *The age of a movie is a better predictor than its release year: Each* model with the `age` variable has a lower RMSE than the corresponding model with the `year` variable.
- *The best non-genre model is only slightly better than the worst non-genre model:* The improvement is only 0.13%.

Adding the genre effects with $\lambda_2 = 1.8$ to the best non-genre model substantially reduced the RMSE to 0.79548 (Table 5), an improvement of 7.06%. Finally, combining the two models and running them on the validation set yielded an RMSE value of 0.85888.

Conclusion

The preceding analysis shows that the most significant variables for a linear model of movie ratings are the user, movie, and genre variables. Including the release year or age of the movie at rating time had a much smaller effect on accuracy. Furthermore, determining the effect of individual genres on a user's rating allows a rating to be predicted for a movie whose combination of genres has not been previously rated by a user. This is especially important for a model that will be used as the basis for a recommendation system.