Netflix Movie Rating Prediction

Chi-Jen Chien
North Carolina State University
Raleigh, North Carolina, USA
cchien@ncsu.edu

Danielle Hancock
North Carolina State University
Raleigh, North Carolina, USA
dnhancoc@ncsu.edu

Eddy Huang
North Carolina State University
Raleigh, North Carolina, USA
whuang25@ncsu.edu

ACM Reference Format:
Chi-Jen Chien, Danielle Hancock, and Eddy Huang. 2021. Netflix Movie Rating Prediction. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

This project is to predict the users' ratings according to the attributions of a movie based on the data set of users' previous ratings for numerous movies. The input data is joined by two data sets. One is Netflix Prize Data, which contains only user ratings of various movies and the movie titles with years. The other is IMDb data set, which provides additional information about the movies, such as genre, director, actors, etc. We use KNN, Random Forests, Gradient Boosted Trees, Logistic Regression, and SVM to predict the ratings and evaluate the results to select the best model on movie rating prediction.

2 BACKGROUND

A common challenge in machine learning is rating prediction and recommendation: Based on a customer's rating or purchase history, predict how much they will like other items, in order to offer effective recommendations. Netflix uses machine learning to suggest new movies and series to its customers based on those they have already watched, in the hopes of keeping customers on Netflix. In 2006, in an effort to improve their recommendation system, Netflix released a data set of over 100 million user-generated ratings of 18,000 movies, from nearly 500,000 different users. This data was released as part of the Netflix Prize Challenge, a competition with the goal of producing a reduction of at least 10 percent in the root mean squared error of rating predictions, as compared to Netflix's existing recommendation system, Cinematch [3]. Such collaborative filtering models use information about the whole group to make predictions about individuals [4].

In 2009, the team BellKor's Pragmatic Chaos (a coalition formed from the teams BellKor, Pragmatic Theory, and BigChaos) won the million-dollar prize, with a 10.06 percent improvement over Cinematch [9]. This team used a complex ensemble method that combined many diverse models, including matrix factorization, neighborhood models, Restricted Boltzmann Machines, gradient boosted decision trees, clustering models, linear regression, and neural networks. [6] [11] [1]. The winners stated that combining various models was by far the most effective approach, and that even a hopelessly inferior model could improve the ensemble's performance if it was not highly correlated with the other models [4].

One common approach to rating prediction is a K-Nearest-Neighbor (KNN) algorithm. KNN classifiers for this purpose often measure *user* similarity (based on rating history), with the premise that similar users will rate movies similarity [5]. Such an approach was necessary in the case of the Netflix Prize challenge, as the data set provided no information about movies other than the title, year, and ratings by various users. This user-based approach can be effective, but is often very computationally expensive, as a data set may contain thousands of ratings by thousands of users [5]. A potentially less computationally expensive approach is to measure *movie* similarity, focusing on the movies that the individual in question has rated in the past, rather than the entire rating history of all users. Sarwar et al. suggest such an approach [13].

According to "A data mining approach to analysis and prediction of movie ratings" by Saraee, White, and Eccleston [8], some features are more relevant than others for predicting ratings. Their results show that by far the most significant factors are the actors and actresses appearing in the movie, with over 90% relevance. The second most important factor is the director, with approximately 55% relevance. The budget is also fairly significant at around 28%. The certification the movie receives and its genre play only a small part, less than 5% each.

3 DATA PREPROCESSING

3.1 Join Two Data Sets

The data we want to use is a combined data set with movie titles, sufficient movie attributions, and corresponding customers' ratings. To obtain this, we combined the Netflix Prize Data [10], which contains only user ratings and the IMDb movie data, which provides additional information about the movies [7].

3.2 Feature Engineering

We selected 8 out of 17 features, which including genre, duration, country, language, director, writer, production company, and actors based on the data completeness. Other attributes, such as budget, were eliminated because a large portion of the data was missing. For space and efficiency reasons, we chose to store only the top three actors. Because we expect the Movie Id and the Customer Id are nominal features, we cast them from integer to sting. To manage the string feature, we encoded them to unique integers and did One-hot encoding. And we will implement different feature selectors in the training part based on different requirements of the methods.

Conference'17, July 2017, Washington, DC, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

4 PROPOSED METHODS

4.1 KNN

First, we will perform correlation analysis, using Analysis of Variance (ANOVA) to determine which movie attributes are most highly correlated with Netflix ratings. And we will implement a KNN classifier, with attributes weighted according to the results of the correlation analysis. Unlike previous KNN approaches, we will measure movie similarity, in terms of its attributes, rather than user similarity. Nominal attributes, such as genre, will simply have a distance of 0 if they are the same, or a distance of 1 otherwise. So, to predict a user U's rating of some movie M, we will look at the movies U has rating in the past, find the K most similar movies to M, and average their ratings. The hyperparameter K will be tuned using the validation data set.

4.2 Random Forests And Gradient Boosted Trees

Decision tree classifier, Random forests classifier, and Gradient-boosted tree classifier are all classical tree methods for doing classification. Random Forests can improve the performance of the Decision tree by using many trees with a random sample of features chosen as the split. For the Gradient-boosted tree, we used decision trees as the weak learner and created an additive model to add weak learners to minimize the loss function. Trees are added at a time, and existing trees in the model are not changed. Besides, a gradient descent procedure is used to minimize the loss when adding trees.

We implemented Chi-Squared feature selector to do the feature selection. And we will use the in-build tree functions in Apache Spark to build our tree models. Hyperparameters such as the maximum number of trees, the maximum depth of trees, and so on will be set at default values.

We will evaluate the tree models by the root mean squared error (RMSE). Since it is a multi-classification prediction, we will also use f1-score as the metric to consider both recall and precision to prevent distortion by label unbalance. And we will tune the hyperparameters based on the results of the validation set.

4.3 Logistic Regression

According to the study by Adam Sadovsky and Xing Chen[12], both logistic regression with L1 and L2 regularization performed well on the Netflix Prize dataset[10]. We will use multinomial logistic (softmax) regression[2] to train the training data. The conditional probabilities of the outcome classes $k \in 1, 2, \ldots, K$ are modeled using the softmax function.

$$P(Y = k|\mathbf{X}, \boldsymbol{\beta}_k, \beta_{0k}) = \frac{e^{\boldsymbol{\beta}_k \cdot \mathbf{X} + \beta_{0k}}}{\sum_{k'=0}^{K-1} e^{\boldsymbol{\beta}_{k'} \cdot \mathbf{X} + \beta_{0k'}}} \tag{1}$$

And we minimized the weighted negative log-likelihood, using a multinomial response model with elastic-net penalty to control for overfitting.

$$\min_{\beta,\beta_0} - \left[\sum_{i=1}^{L} w_i \cdot \log P(Y = y_i|\mathbf{x}_i) \right] + \lambda \left[\frac{1}{2} (1 - \alpha) ||\boldsymbol{\beta}||_2^2 + \alpha ||\boldsymbol{\beta}||_1 \right] \tag{2}$$

We implemented Chi-Squared feature selector to do the feature selection. And we will use the in-build multinomial logistic regression function in Apache Spark to build our model. Hyperparameters such as the maximum number of iterations, the elastic net parameter (α), and the regularization parameter (λ) will be set at default values.

4.4 SVM

Although Support Vector Machine (SVM) is frequently used for binary classification, it is one of the most popular algorithms widely used for building predicting models due to its high performance of accuracy. In the case of this project for predicting movie ratings, the SVM Regression Model is adopted. SVM Regression is designed to find a function that has the largest deviation from the actual obtained target for all the training data regardless the errors. Finding the function will be implemented by applying linear kernel function to map the data from the input space to the feature space via performing linear regression. In the training phase, all the features such as the director, the actor, the genre of the movies that need to be studied will be used to make a N-dimensional hyper-cube, then the model will attempt to find the maximum margin to separate the classes of data. In the testing phase, we will use the model to predict the movie ratings from the user. Finally, we will evaluate the prediction accuracy.

5 EXPERIMENT

We will use a simple holdout approach, with 3 sets: training (60%), validation (20%), and testing (20%), to evaluate our classifiers. Unfortunately, the qualifying (test) data from the Netflix Prize Challenge was not provided, so we are unable to test on the same data as the competition participants. Accuracy will be evaluated by the root mean squared error (RMSE), as this is the metric used in the Netflix Prize Challenge.

We will test the models individually, as well as an ensemble method which linearly combines the results of the other models, with weights determined by the model's accuracy. Based on the findings of BellKor's Pragmatic Chaos, our hypothesis is that the ensemble method will perform better than any individual model. As a baseline for comparison, we will use a simple average of all ratings for a given movie in the training set, regardless of user. We expect all of our models to exceed the baseline.

Additionally, we want to determine which of a movie's attributes are most highly correlated with its ratings. for this, we will use Analysis of Variance (ANOVA), since most attributes are nominal. Based on the study by Saraee, White, and Eccleston [8], we expect that actors will be the most important attribute, followed by director.

6 RESULTS

6.1 KNN

6.2 Random Forests And Gradient Boosted Trees

6.3 Logistic Regression

6.4 SVM

7 MEETING SCHEDULE

(1) 4/7 (Wednesday) 8:30 - 10 pm
Participants: Chi-Jen Chien, Danielle Hancock, Eddy Huang
(2) 4/12 (Monday) 4-5 pm
(3) 4/19 (Monday) 4-5 pm
(4) 4/26 (Monday) 4-5 pm

REFERENCES

[1] Michael Jahrer Andreas Toscher and Robert M. Bell. 2009. *The BigChaos Solution to the Netflix Grand Prize.* TheBigChaosSolutiontotheNetflixGrandPrize

[2] Apache Spark 2021. *Multinomial logistic regression.* https://spark.apache.org/docs/2.3.0/ml-classification-regression.html#multinomial-logistic-regression

[3] Robert M. Bell and Yehuda Koren. 2007. Lessons from the Netflix Prize Challenge. *SIGKDD Explor. Newsl.* 9, 2 (Dec. 2007), 75–79. https://doi.org/10.1145/1345448.1345465

[4] Robert M. Bell, Yehuda Koren, and Chris Volinsky. 2010. All Together Now: A Perspective on the Netflix Prize. *CHANCE* 23, 1 (2010), 24–29. https://doi.org/10.1080/09332480.2010.10739787 arXiv:https://doi.org/10.1080/09332480.2010.10739787

[5] Ted Hong and Dimitris Tsamis. 2006. Use of KNN for the Netflix Prize. (2006). http://cs229.stanford.edu/proj2006/HongTsamis-KNNForNetflix.pdf

[6] Yehuda Koren. 2009. *The BellKor Solution to the Netflix Grand Prize.* https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf

[7] Stefano Leone. 2020. *IMDb movies extensive dataset.* https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset

[8] S. White M.H. Saraee and J. Eccleston. 2004. A data mining approach to analysis and prediction of movie ratings. (2004). http://usir.salford.ac.uk/18838/

[9] Netflix. 2009. *Netflix Prize: Forum.* https://www.netflixprize.com/community/topic_1537.html

[10] Netflix. 2019. *Netflix Prize data.* https://www.kaggle.com/netflix-inc/netflix-prize-data

[11] Martin Piotte and Martin Chabbert. 2009. *The Pragmatic Theory solution to the Netflix Grand Prize.* https://www.netflixprize.com/assets/GrandPrize2009_BPC_PragmaticTheory.pdf

[12] Adam Sadovsky and Xing Chen. 2006. Evaluating the effectiveness of regularized logistic regression for the Netflix movie rating prediction task. (2006). http://cs229.stanford.edu/proj2006/SadovskyChen-NetflixL1LogReg.pdf

[13] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web* (Hong Kong, Hong Kong) *(WWW '01)*. Association for Computing Machinery, New York, NY, USA, 285–295. https://doi.org/10.1145/371920.372071