Time Influence on Ratings in the Netflix Prize Dataset

Timothy Ozimek
North Carolina State University
teozimek@ncsu.edu

1 Background

The online movie service, Netflix, gives a user the choice of watching amongst several thousand movie titles. While not all titles are available at all times, Netflix will rotate selections, and add new ones, during three month windows. Netflix offers a recommendation system that suggests movies that suits a user's tastes. This feature works especially well after a user has recorded a rating after they watch a movie. The more ratings they score, the better the suggestion becomes.

One way of understanding the recommendation system is through everyday word-of-mouth conversation. Friends and acquaintances tend to share common opinions over a variety of topics. Likes and dislikes accordingly tend to coincide. The field of data mining and machine learning has observed this relationship and formalized the task as a recommender system (need citation). Such systems tend to employ the technique of collaborative filtering. In short, this method scours data, looking for commonalities in a request. Items deemed "close enough" to a request are filtered and used to predict. This filtered group may contain ratings not in the initial request. Based on these ratings an estimate is constructed for how a given request might rate that item. In this way, recommendation systems offer new knowledge.

The Netflix Prize dataset was supplied by the Netflix movie corporation in 2006 for use in a prediction contest. The objective of the contest was to predict how a user would rate a movie and compare through the root-mean-square-error (RMSE) metric at how well the scheme worked. The data set is vast, occupying over 100 million ratings from 480,000 users on 17700 movies. The objective was to beat Netflix's proprietary system, *Cinematch*, by 10 % on the RMSE. metric. Part way through the contest a Progress Prize was awarded because of an 8.43 % improvement. This result was produced after 2000 hours of work with the use of a combination of 107 algorithms[1].

Netflix's services have since changed from mail-order DVDs to a streaming service. As a result, the fundamental recommendation system required new adaptations; the final prize's full algorithm was not used because it was applicable to the old service. However, two main algorithms, singular-value decomposition (SVD) and restricted Boltzmann machines (RBM) found useful roles under Netflix's expanding recommendation system[1].

The objective of this paper is less ambitious than the Netflix Prize. It will explore user-based collaborative filtering (UBCF) through the *recommenderlab* R framework and custom C++ software. It will attempt to implement some variations described in the Netflix Prize winners' preliminary paper found in [3]. The dataset only provides the following data: user id, movie id, user rating, user rating data, movie title, and movie release date. Movie title offers too little information for an approach such as bag-of-words to help categorize data as many of the titles involve proper names, thus these are discarded.

2 Method

Model 1: Load data, give rating to user of average ratings for a movie.

Model 2: Employ user ratings filter via cosine metric. Average matching results.

Model 3: Explore with nearest neighbor setting.

Model 4: Combine model 1 and model 2. For cold start request, weight global rating more than filter, up to a threshold. If global rating has low variance, apply even more weight.

Model 5: Search for changing bias and use as filtering indicator. This requires time gets involved. Use cutoff to see if bias exists. Will require timestamps.

3 Experiment

4 Conclusion

Under Construction.

References / Papers to Read

[1] Xavier Amatriain, Justin Basilico. *Netflix Recommendations: Beyond the 5 Stars (Two Parts)*. The Netflix Tech Blog, 2012. https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429. 11/5/2017.

[2] Robert M. Bell, Yehuda Koren. (2007) *Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights*. Seventh IEEE International Conference on Data Mining. pp. 43-52. IEEE.

[2] Linyuan Lu, Matus Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, Tao Zhou. (2012) *Recommender systems*. In Physics Reports pp. 1-49, Elsevier B. V. 0370-1573.

[3] Michael Hahsler (2017). recommenderlab: Lab for Developing and Testing Recommender Algorithms. R package version 0.2-2. http://lyle.smu.edu/IDA/recommenderlab/

[4] Arkadiusz Paterek, (2007) *Improving regularized singular value decomposition for collaborative filtering*. KDDCup.07 pp. 39-42. ACM 978-1-59593-834-3/07/0008