# Global Blending on Sparse Ratings in the Netflix Prize Dataset

**Timothy Ozimek**
North Carolina State University
teozimek@ncsu.edu

## 1 Background

The movie service, Netflix, offers users the choice of watching several thousand movies on demand. While the service now offers a highly popular online, on-demand, streaming service, in the past, operations were run through a mail DVD system. A user of the service could request a certain number of movies at any given time up to a limit. After watching and returning the DVDs a customer could receive new movies for as quickly as the mail delivery service could deliver them. Along with the movies, Netflix provided a list of recommended movies based on a user's viewing preferences. A viewer can voluntarily rate movies they've seen with the understanding that their rating would guide Netflix in suggesting new movies. The more ratings they gave, the better the recommendations became.

To provide this service, Netflix wrote the software tool, *Cinematch* . This tool used a combination of machine learning and data mining algorithms to perform its duties. One technique is prominently featured: recommender systems. The intuition behind these systems is based on how word-of-mouth conversations spread news of a product. These conversations occur between friends and acquaintances who 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 similarities in rating patterns, much like word-of-mouth works among similar friends, but at a far larger scale. The result of this similarity seach will produce or filter a group of similar users from which a ratings estimate can be computed.

A variety of algorithms have been adopted with two standing large, item-based collaborative filtering (IBCF) and user-based collaborative filtering (UBCF). The names of each imply their method of search. IBCF finds similarity of items, in this case movies, and recommends based how similar a movie is to another. Likewise UBCF establishes similarity through the intuitive word-of-mouth approach by comparing similar user ratings.

To stimulate research in improving their recommender system, Netflix offered the Netflix prize in 2006. The objective was tantalizingly simple: be the first team to improve recommendations by 10% over *Cinematch* and receive $1 million. Without delving too far into details, contestants were given a dataset of over 100 million user ratings of 17700 movies. Against this dataset algorithms could be devised, tested, and submitted to Netflix for testing. Netflix would run the submission against a hidden testset. A simple root-mean-square-error would be computed per tested rating to arrive at a final score. This score was the metric compared against *Cinematch* to determine the winner.

The prize was captured in September 2009 by the team "BellKor's Pragmatic Chaos", a group of professional statisticians and machine learning experts. Their methods were sophisticated, entirely new and were comprised of 107 predictor algorithms run as an ensemble to produce ratings. At their core, however, they were recommender systems that had advanced ways of producing better similarity metrics.

The objective of this paper is less ambitious than the Netflix Prize. The combined efforts of the winning team included well over 2000 hours of work for their preliminary solution where even their simplest algorithms took 45 minutes to run; several searches spanned over many hours. The scope is beyond what can be provided in less than a semester's time.

It was noted in "BellKor's Pragmatic Chaos"'s work that data sparsity was the main adversary that needed to be overcome. Along these lines, this project will focus on exploring remedies to this issue by using more information from the dataset. In particular, along with similarity of ratings, when sparsity offers little new information, global information on the movie such as average ratings and variance, as well as user bias, will be considered in weighing a score. The further idea is that the sparser the data the more the algorithms will rely on global metrics and will taper-off as more similar information is provided. Time of rating may also be considered to see if it offers better estimates.

The setup will rely on the *recommederlab* tool suite. The purpose of this suite to to expose a sandbox experimental environment for recommender systems. Support is provided for UBCF and IBCF systems as well as variants of singular value decomposition (SVD) and hybrid solutions. Aside from user movie ratings, also were provided date of rating, title of movie, and DVD release data of movie. Briefly, a bag-of-words approach was considered for identifying movie genres but was rejected outright because of too many proper names. Additional support will be provided by C++ routines used to clean data, and provide training and test sets.

## 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