# Movie Recommender Using User Sentiments and Collaborative Filtering(CF)

**Janani Krishna**
University of Massachusetts, Amherst
jkrishna@cs.umass.edu

## Abstract

Often on-line personalized recommendation systems helps to improve customers' satisfaction and needs. In general, a recommendation system is considered as a success if customers purchase the recommended products. The act of purchasing itself does not guarantee satisfaction and a truly successful recommendation system should be one that maximizes the customer's satisfaction.

One of the most popular techniques for recommendation systems is collaborative filtering. This algorithm comes in two flavors: item based and user based. A common practice is to consume only the ratings given by users in the recommendation algorithm. Though this works in practice, this approach does not use the context like user reviews. In this project, we propose one simple approach to incorporate context, namely user reviews into the collaborative filtering algorithm. The experiments below demonstrate the gains with our new proposed approach.

## 1 Introduction

Personalization of product information has become one of the most important factors that affect customer's product selection in today's competitive and challenging market. Personalized recommendations require e-tailers to understand customers and offer goods or services that meet their needs.

Amazon [1] stays ahead of the curve in the eCommence industry by personalized recommendation of items. Google news generates click through rates by showing relevant content to readers. TripAdvisor provides different hotel rankings for different users. Netflix achieves two-thirds of its movie views by recommendations. Concisely, recommendation systems are decision aids that analyze customer's prior online behavior and present information on products to match customer's preferences. Through analyzing the customer's purchase history or communicating with them, recommendation systems employ quantitative and qualitative methods to discover the products that best suit the customer. Most of the recommender systems employ content-based filtering (CBF), collaborative filtering (CF), and other data mining techniques. Most recommender system deal with the rating of the user and ignore the fact that users have given review text that is different from the rating that they have assigned for the product. The aim of this project is to build a recommendation system based on collaborative filtering (CF) method like the user-based collaborative filtering along with sentiments of the user's reviews in collaboration with the rating they have provided. Sentiments score acts as vital role in recommending the right set of movies and thereby reducing the bias of predicting the set of movies solely based on rating. In order to perform collaborative filtering, the project uses the most common metric the Pearson's correlation coefficient that gives a score for the user-user and item-item relationship.

## 2 Related work

The methods and procedures in our recommendation system are used widely, not only in movies,but also in various other areas such as music, news and e-commerce. Companies like Facebook, Twitter and LinkedIn also use such methods to recommend friends/followers/connections [2]. As mentioned by Hu et al. [3], one of the most common approaches to collaborative filtering is that of neighborhood models. The underlying assumption is that users with similar ratings on some items will have similar ratings on the others (an analogous assumption is made for items that share similar ratings for many users). Another set of methods that has shown promise recently relies on lowrank matrix factorization, which seeks to uncover the most important factors governing movie choices. These two approaches are the ones we will be focusing on in the rest of this project. More work has related to collaborative filtering are been performed like in netflix.[4]

In some specific scenarios the search for neighbors among a large user population generally poses an issue. Thus, item-based algorithms [5] avoid this bottleneck by exploring the relationships between items first, rather than the relationships between users. Recommendations for users are computed by finding items that are similar to other items the user has liked. Furthermore, user based algorithms like clustering techniques work by identifying groups of users who appear to have similar preferences. After the clusters are created, the predictions for an individual can be made by averaging the opinions of other users in that clusters. Some clustering techniques represent each user with partial participation in several clusters. The prediction is then an averaged across the clusters, weighted by degree of participation. The clustering technique can be used in combination to nearest neighbor to shrink the candidate set helping to reduce the computational cost. Recently there have been considerable use of sentiments in predicting the user rating and recommendation. There are algorithms on sentiment analysis of reviews and opinions about products in particular to movie reviews. The methods include a mixture of machine learning and NLP techniques [6].

## 3 Methodology

Product reviews have a wealth of data with attributes like user name, date, time, ratings, title and review text. It makes lot of sense to incorporate the review text into the item-based and user-based collaborative filtering algorithm. In this project, we propose a simple method to incorporate review text. More specifically, the problem we intend to tackle is remove inconsistent ratings from the data using review text.

Consider these two examples from our dataset:

- product/productId: B000M86HPC
  review/userId: A27DHFFWMH042Y
  review/profileName: Cathleen M. Walker "geminiwalker"
  review/score: 1.0
  review/summary: A pathetic disappointment
  review/text: I so wanted to love this movie. How could it fail?The dialogue was pointless, inarticulate and barely audible, no matter how high I turned up the volume. The story meandered endlessly, yet went no where. I still don't quite understand more than I already knew - once there was a woman who was murdered and the case was never solved.
- product/productId: B00004NKCQ
  review/userId: A27DHFFWMH042Y
  review/profileName: Cathleen M. Walker "geminiwalker"
  review/score: 3.0
  review/summary: Disappointing
  review/text: I was intrigued by the title of this movie because one of my favorite songs has the same name and a science fiction based lyric. Unfortunately, that is all they have in common, and the song has a better storyline. I really had high expections for this film, it had so much potential, even for its time. Anyway, I was disappointed, and I did have a hard time maintaining interest in the long, drawn out story that really went no where. I wish I had felt differently.<br /><br /><a href="http://www.amazon.com/gp/product/B0000542C6">The Emerald Forest</a>

As we can see from the above examples there are some consistency/inconsistency between the user rating and the text review. For example in the second review the user has criticized the movie and yet still has given a rating of 3 which adds a bias when recommending this movie to another user.

We train a binary Logistic regression classifier which classifies the review text as consistent or inconsistent relative to the rating. We consider all the reviews with ratings 3/4/5 as label 1 and reviews with ratings 1 and 2 as 0. The features are based on sentiment scores from the reviews.
Firstly, we perform sentiment analysis over a set of distinct user reviews using the nltk Vader polarity. Over the iteration of the sentiment analysis the polarity score of the each sentence is calculated and categorized as either positive, negative or neutral sentence. We aggregate these scores over the review text to form our feature vector.

A Logistic regression model is trained on this data and tested over the remaining users. We perform 5-fold cross validation to get the best set of hyperparameters. Using this LR model, we filter out the inconsistent reviews from our data. So our data consists of following columns:
**<User id, product id, actual rating, filterOrNot>**

We perform collaborative filtering on this data. The basic premise here is that ratings without consistencies should improve the performance of collaborative filtering.

### 3.1 Recommender System Method - Collaborative Filtering

In Collaborative Filtering (CF) implementation, stores the data in a 2-D matrix. Thus each user in a row will have ratings for each product in the column. After getting the 2D dataset matrix, we implement two kinds of recommendation system models: item-based and user-based collaborative filtering. For both models, we need to compute the similarity and prediction score. In the following part,we will explicitly show how we build our item-based and user-based recommendation system, and compare different models in the following subsection.

#### 3.1.1 Item Based Collaborative Filtering

Foremost step in recommender system is computing the similarity between items so that we can recommend similar items to customers based on what they bought before. The basic idea of similarity computation between two items i and j is to firstly cluster the users who have rated both of these items and then to apply a similarity computation technique to determine the similarity $sim_{i,j}$. This similarity function called the pearson co-relation is defined as follows:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_j)^2}} \tag{1}$$

The above equation denotes that the similarity between two items is measured by computing correlation $corr_{i,j}$ where the set of users who both rate $i$ and $j$ is denoted as $U$, $R_{u,i}$ denotes the rating given by user $u$ for movie $i$, $\bar{R}_i$ denotes the mean rating of the movie $i$.

#### 3.1.2 User Based Collaborative Filtering

In case of user based CF, it is similar to item based CF. Instead of computing the similarity between items here the similarity is computed between two users. The method used for computing the similarity between two customers u, v is the same as item-based method. The similarity is denoted as $s_{u,v}$.

#### 3.1.3 Prediction Computation

The prediction on an item for a user u is calculated by computing weighted sum of different users ratings on item i. The prediction $P_{u,i}$ is given by:

$$P_{u,i} = \frac{\sum_v (r_{v,i} s_{u,v})}{\sum_v (s_{u,v})} \tag{2}$$

where $r_{v,i}$ is the rating given by user v on item i.

# 4 Dataset

This project uses subset of Amazon movie dataset [6] to develop a recommender system that provides recommendation to the customer. The dataset contains 7,911,684 reviews for about 889,176 distinct users and 253,059 distinct products. The dataset includes the following features:

- product/productId:for example, amazon.com/dp/B00006HAXW
- review/userId: id of the user, for example A1RSDE90N6RSZF
- review/profileName: name of the user
- review/helpfulness: fraction of users who found the review helpful
- review/score: rating of the product
- review/time: time of the review
- review/summary: review summary
- review/text: text of the review

The dataset was pre-processed using the sentiment analysis, in which the scores in the dataset were grouped for positive and negative sentiments. For experimental purpose a subset(about 5,00000 reviews) of the reviews were used.

The training dataset looks as below:

**<UserID, ProductID, NegativeScore, NegativeCount, NeutralScore, NeutralCount, PostiveScore, PostiveCount, Bias, Rating, New Rating>**
<A1XIOOXV9LAZXZ, B0028O9UR0, 0.027785714285714285, 0, 0.7847142857142858, 14, 0.1875, 0, 1, 4.0, 1>

# 5 Experiments and Results

The first step towards the experiments is sentiment analysis. A subset of the dataset(5,00000 user reviews) is taken and split into training and test set (80-20). The Logistic regression model is trained with the train set and test set is predicted. The accuracy of the model came upto **86%**. The actual rating and the predicted rating is compared and the reviews that were inconsistent with the user rating is discarded. The final result containing the userid, productid ,user rating, sentiment rating(0's and 1's). We rely on the ratings in the train data for our labels. Ideally we would like to assign a consistent/inconsistent label to each review manually. The assumption we make while using the ratings as is :given enough data, the LR model should be able to capture the consistent ratings from the inconsistent ones.

In the CF experiments, we vary the size of the training set and the test set by splitting randomly with equal probability the ratings written by the users in the whole dataset. So about half of the ratings written by the users in the whole dataset will be put into the training set and the other half will be put into the test set.

Furthermore, we filter user-recommendation pairs which do not have enough support. Specifically, in user based CF, we impose a restriction that number of common users who have rated on a product to be atleast 'n'. Likewise, in product based CF, we impose a restriction that the number of common products between two users is at least 'n'. We choose n (over n items bought by a particular user) to be 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. In case of user based CF 'n' signifies the number of common items between the user where as item based 'n' signifies the number of common users between two items

For above setting of 'n', item-based CF and then item-based CF with the sentiment analysis is performed. The result is shown in (Figure 1). Similar experiment is performed for user based CF for n ranging from 10 to 50 and the results can be seen in (Figure 2). User-based collaborative filtering recommendation system consumes more time on calculating the similarity and rating scores for related items and it works well when the scale is not so large and data refreshment is relatively frequent.

Also an experiment is performed to check how relevant the recommended score for a particular item for a user is to the rating the user has provided. The percentage of accuracy is tabulated below (Table 1) for item-based CF and for item-based CF with sentiment analysis. The same was not experimented

Table 1: Comparing the predicted recommended ratings with the actual ratings

| Model | % of rating matches | % difference=1 | % difference >=2 ($\mu$m) |
|---|---|---|---|
| Product Based CF | 63.03% | 15.06% | 21.91% |
| Product Based CF with Sentiment Analysis | 82.14% | 10.28% | 7.58% |

over user-based CF as the computation complexity of the user-based is more and takes more time to run.

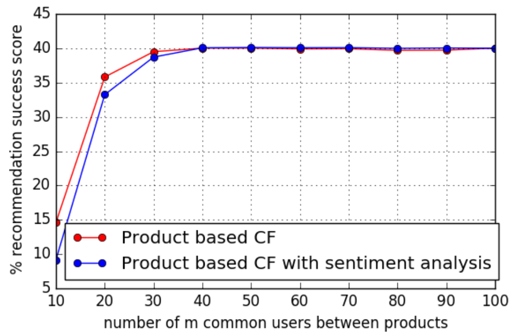


Figure 1: item-item CF and sentiment analysis

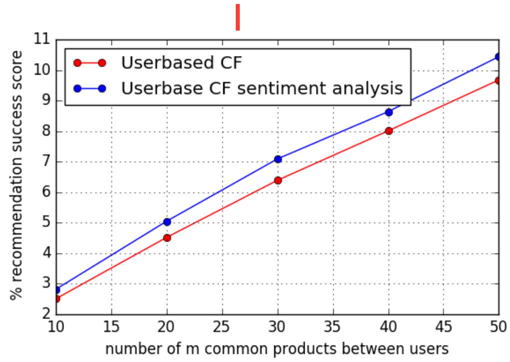


Figure 2: user-user CF and sentiment analysis

## 6 Discussion and Conclusions

From the series of experiments as mentioned and from Figure 1 and Figure 2 it is clear that upon sentiment analysis there is about 2-4% improvement in case of item-based with sentiment analysis and user-based sentiment analysis in comparison to default collaborative filtering. Also from Table 1 shows that the accuracy of CF with sentiment analysis is better when compared to the default CF. Thus filtering inconsistent ratings using sentiment analysis provides two advantages:
1. We require lesser number of common users to get effective product similarity
2. The predicted recommendation score are more consistent with the actual test predictions
Also item-item CF seems to perform faster when compared to user-user CF due to the matrix formation which is costlier in case of user based CF. In future the performance of the system can be made better by extracting features from the review text provided by the user and perform more experiments on those features.

## References

[1]Linden, Greg, Brent Smith, and Jeremy York. "Amazon.com recommendations: Item-to-item collaborative filtering." Internet Computing, IEEE 7.1 (2003): 76-80.

[2] Paul B Kantor, Lior Rokach, Francesco Ricci, and Bracha Shapira. Recommender systems handbook. Springer, 2011.

[3] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer-Verlag.Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In In IEEE International Conference on Data Mining (ICDM 2008, pages 263–272, 2008.

[4] Paterek, Arkadiusz. "Improving regularized singular value decomposition for collaborative filtering." Proceedings of KDD cup and workshop. Vol. 2007. 2007.

[5]Sarwar, Badrul, et al. "Item-based collaborative filtering recommendation algorithms." Proceedings of the 10th international conference on World Wide Web. ACM, 2001.

[6] K. Dave, S. Lawrence, & D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings WWW 2003, 2003.

[7] http://snap.stanford.edu/data/web-Movies.html