# Seminar Report
# Recommender Systems using Collaborative Filtering

Varun B Patil
8th sem B.E CS & E
SJCE, Mysore

March 12, 2012

## Abstract

Recommender systems (also known as Recommendation engines) are systems that recommend objects of interest to users based on their past activities and the activities of other users in the same social environment. In this seminar, I will talk about Recommender systems for an online movie streaming website, where it can be used to recommend top rated movies to the user based on his/her preferences, past movies viewed and also based on the ratings by other users with the same psychology. The algorithm used here is called "Collaborative Filtering" and as the term "collaborative" suggests, it takes into account the movie ratings from other users as well as the user's own movie ratings.

## 1 Introduction

Recommender systems have become an integral part of almost every e-commerce website these days. That is how these e-businesses coerce people to buy more stuff and that is how they make more money. The recommendations made should not disappoint the customers. Thus it is very important that the recommendation engine is spot on in its predictions and is able to learn user preferences quickly and efficiently. Some famous examples of highly sophisticated recommendation engines are those from Amazon for book recommendations, Netflix for movie recommendations, YouTube for video recommendations, Pandora radio for music recommendations, etc. The most important aspect to note here is that, nowadays, social media is a big industry and everybody wants to buy things that other users have bought; everybody wants to watch movies that other users have watched and liked. E-commerce websites are cashing in on this craze by providing recommendations to users. Thus, where a user would normally have bought one movie DVD online, the user will now be coerced to buy another two with the same theme, thus raking in the moolah fast and easy for the e-business. And of course, they are relying on the human psyche where it is almost impossible to resist buying the recommended items, especially if there are huge discounts that come along with it.

## 2 Collaborative Filtering

As the term "collaborative" suggests, this machine learning algorithm considers movie ratings (in the context of this seminar) of all other users in the social environment while predicting movie ratings for a particular user. The movie rating predictions made by the collaborative filtering algorithm can then used to recommend top rated (predicted ratings) movies for a user.

### 2.1 Advantages

There are several advantages of Collaborative filtering algorithm compared to other algorithms and that is exactly the reason for considering this particular algorithm for the seminar. Some of these advantages are

- Can start with a completely empty movie ratings database.
- New movies need not be reviewed by some person for it to be recommended to some user ( i.e, new movie with no ratings in the database ).
- Recommendations can be provided to a user even if he/she has not rated even a single movie in the database.

Other algorithms that can be used for building recommendation engines are content-based filtering, hybrid recommender systems, K-nearest neighbor (k-NN), Pearson correlation, Rocchio Relevance Filtering, etc.[1]

## 3 Inputs

The only input that is required to run this algorithm is a matrix of movie ratings which is a n(m) X n(u) matrix. It is however possible that entries in this matrix can be empty which corresponds to a particular user not rating a particular movie. In the extreme case, the entire matrix may be empty. This is where Collaborative Filtering aces other algorithms. It can predict movie ratings even in these less-than-ideal conditions.

## 4 Learning

The Collaborative filtering algorithm learns two entities. A feature vector 'X' for each movie and a parameter vector 'Theta' for each user in the database such that it meets a certain criteria or optimizes a particular function. The function in question here is called a Cost function represented by 'J'. In the case of Collaborative Filtering, the cost function is nothing more than sum of squared error between the predicted ratings and the actual ratings. Now, arises the question of calculating the predicted ratings which are required to calculate the Cost function. The predicted ratings is a n(m) X n(u) matrix, just like the original ratings database and is obtained by the product of the feature vector 'X' and the transpose of the parameter vector 'Theta'.

We start out with randomly initialized feature and parameter vectors and in each iteration we continue to make modification to these vectors so that the cost function is minimized after each iteration. After a certain fixed number of iterations, we expect the algorithm to converge to a particular value for 'X' and 'Theta' at which the cost function is minimum. It is important to note that simply increasing the number of iterations beyond a certain value will not lead to more accurate learning or a lesser cost function value, but rather, it increases the learning time with no more better results than before. Thus, choosing the right value for the number of iterations is important. The question now is how to modify the feature and parameter vectors in each iteration so that after each iteration the predicted ratings get closer to the actual ratings in the database.

For this, we need to define gradients for each of 'X' and 'Theta'. Using these gradient definitions and cost function definition, certain open-source optimization libraries like fmincg or fminunc in Octave can calculate 'X' and 'Theta' that optimizes the cost function.

Another important consideration is the number of features for a movie or the number of parameters for a user (must always be equal). Movie features may correspond to the movie genre like comedy, action, drama, thriller, etc. Parameters for the user might correspond to his/her liking of a particular movie genre (although there might be no direct correspondence between the movie genre and the feature or parameter values). More number of features of parameters might be good for accuracy of the predictions, but just like choosing the number of iterations, increasing it beyond a certain point will only be detrimental to the algorithm's run-time rather than improve accuracy.

# 5 Making recommendations

Once the predicted ratings are available, it is a trivial function to recommend movies with the highest ratings for a particular user to that user. There is no machine learning involved here. It is a simple task of sorting rating values and printing them in descending order.

# 6 Algorithm Improvements

There is no doubt the above described algorithm will work for many cases, but there are a few simple improvements that could be made so that it always works and is more efficient and fast at it. Some of these simple improvements are describes below.

## 6.1 Regularization

[2]It is quite common for every machine learning algorithm to overfit the training samples which means that the predicted values come very close to the actual values in the training set, but fail to predict correctly for samples outside the training set, and Collaborative filtering is no exception to this. However, the cost function definition and the gradient definitions may be modified suitably to overcome this. Care should be taken so that you do not overdo the corrective measures so much so that you end up with an algorithm that underfits the data.

## 6.2 Feature scaling

[2]When some features have a very high range of values compared to other features, this high valued feature has a greater influence on the learning algorithm. To prevent this domination by features, prior to running the learning routine, we ensure that all features fall in the same range. This process is called Feature scaling.
As an example consider that we are predicting house prices based on two features - area of the plot and number of bedrooms. It is obvious that the area feature is very much greater than the number of bedrooms feature. A 5000 sq. feet plot may have just 3 bedrooms. This results in the are feature having a greater influence on the learning process than the number of bedrooms feature. Since we want to avoid such a thing, we scale the features by subtracting from mean and then dividing by their variance. Now all features fall in the same range.
The same concept applies to the movie feature vectors and user parameter vectors in the Collaborative filtering algorithm.

## 6.3 Mean normalization

[2]If a user has not rated any movie and still wants movie recommendations, the original Collaborative filtering algorithm will end up predicting 0 rating for all movies in the database. This is clearly undesirable. It

would be more intuitive if the user is given the average rating for the movie considering all the ratings by other users for that movie in the database. However, in doing so, we should not end up ruining the algorithm for the normal cases. One way to accomplish what is intuitive for the special case and what is correct for the normal cases is Mean normalization where all the ratings in the database for each movie in the database in subtracted by the mean rating for that movie so that the ratings for any movie in the database has a mean of 0 and a variance of 1. This small modification allows the algorithm to work correctly for every case and does not require the algorithm to handle special cases separately from the normal cases.

## 7 Conclusion

Collaborative filtering is a simple but powerful algorithm to build recommendation engines that are becoming so popular and mature these days due to massive socializing of the user's community. Incorporating socializing features in online businesses is the name of the game these days and are the way forward for massive expansion of online businesses and massive profits for the business and at the same time, prioritizing customer satisfaction and focusing on gathering a massive customer base. A very good example in the Indian e-commerce space is the hugely popular "Flipkart". Where people were once overly hesitant to conduct transactions online, due to fear of substandard quality and online frauds, Flipkart has transformed the Indian buyers mindset and has quite intelligently adopted recommendation systems like the ones above to gain a huge following. Thus, there is no doubt that recommendation systems are the way forward for online businesses.

## References

[1] Wikipedia.org, *Recommender Systems.*

[2] Andrew Ng, *Collaborative Filtering for E-commerce*, Stanford University online course, 2011.