Towards a hybrid approach to Netflix Challenge

Abhishek Gupta, Abhijeet Mohapatra, Tejaswi Tenneti

March 12, 2009

1 Introduction

Today Recommendation systems [3] have become indispensible because of the sheer bulk of information made available to a user from web-services(Netflix, IMDB, Amazon and many others) and the need for personalized suggestions. Recommendation systems are a well studied research area. In the following work, we present our study on the Netflix Challenge [1]. The Neflix Challenge can be summarized in the following way: "Given a movie, predict the rating of a particular user based on the user's prior ratings". The performance of all such approaches is measured using the RMSE (root mean-squared error) of the submitted ratings from the actual ratings. Currently, the best system has an RMSE of 0.8616 [2].

We obtained ratings from the following approaches:

1. The global or the average rating of a movie
2. The rating obtained by user's preferences of genres, directors and the respective ratings (Content Based Approach)
3. The rating influenced by a set of śimilarúsers based on their preferences of genres(Collaborative filtering)
4. The weighted sum of the 3 ratings above

Section 2 describes the data sources that we have used. Sections 3, 4 and 5 describe the first three approaches respectively. In Section 6, we present our results. In section 7, we discuss some interesting findings which lead to the hybrid approach (iv). Section 8 deals with conclusion and future work.

2 Data Sources

We used the Netflix Dataset provided at [1]. To build movie profiles we obtained the rating, genres and director information from IMDB. It was interesting to note that most of the Netflix data pertaining to movie names did not correlate with those in IMDB (about 7000 movie names). This led us to use Google for querying the IMDB data on these movies by their names and year of release. This matched most of the movies (about 16,500 out of 17770). On querying Google with only the movie names, we were able to match 17500 out of 17770 movie records in IMDB.

3 Global Rating

For the cases, in which we don't have much data to build a comprehensive user profile or find a sufficiently similar set of users, we intend to give a high weightage to the global rating of the movie while predicting ratings of the test set. The global rating for a movie can either be considered as the average of the ratings given by the Netflix users or the IMDB rating. Currently, we take a decision based on the following heuristic:

Algorithm 3.1: GLOBALRATING($movie\_id$)

```
if $NumVotes(movie\_id) >= 100$
  then { if ($NeflixAvg(movie\_id) >= 2.5$)
           and
           ($NetflixAvg(movie\_id) <= 3.5$)
           and
           ($NetflixStdDev(movie\_id) >= 1.3$)
           $mrating = (IMDBAverage(movie\_id))/2$
         else $mrating = NetflixAvg(movie\_id)$ }
  else $mrating = NetflixAvg(movie\_id)$
return ($mrating$)
```

For movie ratings between 2.5 and 3.5, if the standard deviation of rating is more than a certain threshold then we assume that the Netflix average rating for that movie is not reliable. Further, in cases where the number of ratings of a movie is below a certain threshold then we take IMDB rating to be the global rating of the movie.

4 Content Based Approach

Based on the movies that the user has already seen, we build user's profile. To this end we use the genre and director information extracted from the IMDB data. The motivation for content based approach stems from the fact that a certain set of users may not share common interests with the rest of the users despite having a lot movies in common. We believe that genre information and the director of a movie are most indicative of a user's liking for that movie. For every user, we compute the average rating for individual directors and genres. Further, we also maintain the rank of a director or a genre specific to user. The weights of the ratings based on director

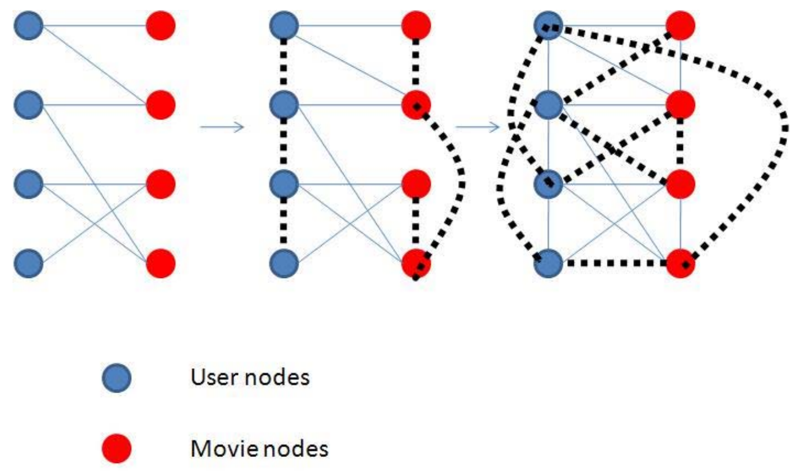

Figure 1: Growth of SimRank for user-movie pairs

and genre information are decided by the relative importance of the director or genre fields to the user which we compute heuristically.

Scheme used for calculating content based rating:

Algorithm 4.1: ContentBasedRating(u, m)

```
$Rating(u,m) \leftarrow \frac{w_1.r_1.R_1+w_2.r_2.R_2}{w_1.r_1+w_2.r_2}$
$(w_1 = Importance\ of\ directors\ for\ user\ “u”$
$w_2 = Importance\ of\ genres\ for\ user\ “u”$
$r_1 = Inverse\ of\ rank\ of\ \ director\ of\ the\ movie$
$r_2 = Inverse\ of\ rank\ of\ genre\ of\ the\ movie$
$R_1 = Average\ rating\ of\ the\ director\ of\ movie$
$R_2 = Average\ rating\ of\ the\ genre\ of\ movie\ \ )$
```

The rank of a director/genre is computed from the average rating of the directors/genres whose movies user has seen. To compute w_1 and w_2, we filter the list of genres and directors keeping those whose average rating, as given by the user, exceeds 3.5.

- w_1 = total count of the ratings given to directors in this list
- w_2 = total count of the ratings given to genres in this list

5 Collaborative Filtering

It is generally the case in social networks(Facebook, Twitter etc.) that users tend to show common patterns of behavior as their *buddies*. This applies even to the domain of movie ratings, since the number of users are far greater than the number of movies, we expect users to have common movies and futhermore, exhibit similar responses to movies. In this section, we explore two collaborative approaches, namely *Simrank* and *Locality Sensitive Hashing*. Detailed description of these approaches are given below.

5.1 SimRank

It is pretty intuitive that if two users are similar then they would rate similar movies similarly and vice versa. This assumption causes the Netflix ratings problem to fall very neatly into the framework of Simrank [5]. We define the following gragh $G(V, E)$ where $V = \{$set of user ids and movie ids$\}$. The graph initially looks as follows:

- A user-id, movie exists and has the weight $r/5$ if the user had previously seen the movie and rated it r.
- No user-user edge exists if users are distinct. Else weight of a user-user edge is 1.
- A movie-movie edge exists with weight calculated by the Jaccard similarity based on genres

When we run Simrank on this graph, we gradually get user-movie edges for those movies the user had not seen earlier. At the end of each iteration, we get new edges with neighbors that are separated by one hop count. At the end of sufficient number of iterations (say 5), we multiply the weight of a user-movie edge with 5 to get the predicted rating of user u for the movie m.

The novelty of this approach is that it is simple to implement and we can get the user-movie rating directly from this graph. We do not require additional computations to study the trend of similar users over the set of movies as done in traditional collaborative filtering techniques. However, this approach has a disadvantage when it comes to time complexity. Simrank runs naively in $O(n^3)$ time and at the best, we can bring it down to $O(k^2.n^2)$ where k = maximum number of neighbors for a

Table 1: Experimental Results for the Netflix Probe Set

Sl No.	w_{Collab}	$w_{Content}$	w_{Global}	RMSE	# of u-m pairs
1	1.0	0.0	0.0	0.9602	992,681
2	0.0	1.0	0.0	0.6955	172,641
3	0.0	0.0	1.0	1.0537	1,408,395
4	1.0	1.0	0.0	0.7853	121,722
5	1.0	0.0	1.0	0.9674	992,681
6	0.0	1.0	1.0	0.8564	172,641
7	1.0	1.0	1.0	0.8368	121,722
8	1.5	1.0	0.5	0.8339	121,722
9	1.0	1.5	0.5	0.7741	121,722
10	1.0	0.5	1.5	0.9008	121,722
11	1.5	0.5	1.0	0.8961	121,722
12	0.5	1.5	1.0	0.7885	121,722
13	0.5	1.0	1.5	0.8469	121,722

given vertex. We had tried to use this approach to compute the user-movie rating predictions, but were awed by the sheer time complexity.

5.2 Locality Sensitive Hashing (LSH)

Since the number of possible neighbors turns out to be very large, Locality Sensitive hashing [4] seemed an obvious choice to find nearest neighbors. We define a user as a vector containing his average ratings for the set of popular genres. Dimension of this vector was chosen to be 13, although there are 28 distinct genres in all. Two users are deemed similar if the *distance* between their ratings in respective dimensions(or genres) is less. Genres are the ideal choice of signatures since they are highly representative of the content of a movie. Hence, the average rating in each genre, closely correlates with the user's preferences. We have used the LSH implementation of the E2LSH package[1].

Initially, we computed 10 nearest neighbors for every user using LSH. The user's predicted rating for a movie was then caculated using the weighted average rating of his nearest neighbors. This did not yield satisfactory results. We suspected that this might be due to the fact that a small deviation in the ratings of few nearest neighbors might amplify the error in the predicted rating. So in order to ammortize such variations across all sets of neghbors we then used 100 nearest neighbors for every user. Our results significantly improved with this change. We believe that with even larger sets of neighbors per user the results can be further improved. A detailed description of the results is presented in *Section 6*.

[1]We are thankful to Alex Andoni for providing E2LSH package http://www.mit.edu/ andoni/LSH/

6 Results

The experiments were conducted on the Aster nCluster, on the Netflix dataset with 48,106 users and 17,770 movies. The probe-set for predicting ratings had about 1.4 million user-movie pairs. Out of these 1.4 million user-movie pairs, we were able to get content-based ratings for only 121,722 user-movie pairs. However, we were able to get ratings from the collaborative approach for 992,681 user-movie pairs. We use different combination schemes, with varying weights for the ratings obtained from the three approaches discussed. These are summarized above in Table 1.

While computing the average ratings from the Neflix data-set we observed that most of the ratings lay between 3 and 4 (about 12,000 of them) and most of them had a standard deviation of around 1. This implies that very few people rate movies either 1 or 5. They settle for a modest value between 2 to 4.

7 Towards a hybrid approach

We observed that the collaborative filtering (using LSH) gave a RMSE value of 0.9602. However, it gave a RMSE value of 0.8012 for those cases where the actual user ratings were greater than 2 (867,095 cases), and a RMSE of almost 1.6793 in cases where actual user ratings were 1 or 2(125,586 cases). When we tried out a simple average of the content based user ratings and the ratings obtained using the collaborative approach we got a RMSE value of 0.8197! This led us to a hybrid scheme where we obtain the weighted mean of the ratings obtained from the previous three approaches. The weights are decided by a number of factors(Table 2), identified in [6]:

Table 2: Factors affecting the influence of the three aproaches

Factor	Content-based	Collaborative	Global Rating
# of movies rated	+	+	-
Deviation of ratings from average	+	-	-
Rarity of ratings	+	N.A	N.A

1. Number of movies rated: As the number of movies seen by a user increases, so should the weightage of the content-based ratings. The weightage of the collaborative ratings should also be increased since more movies imply more movies in common with other users. In this case, we should rely less on the global rating of the movie.

2. Deviation of ratings from average: If a user's ratings deviate a lot from the movie's average ratings, this implies that the user has different preferences. In this case we should give more weightage to the content based ratings.

3. Rarity of movies: If a movie is seen by small fraction of users then it implies that those fraction of users have different preferences than others. Therefore, in this case too the weightage of the content based ratings should be increased.

Due to limitations of time, we were not able to evaluate the performance of the hybrid scheme based on these parameters. However, we did manage to compare the RMSE values obtained by different orderings of the weights of the three ratings. The corresponding results are described in Section 6.

Getting a lower RMSE value when actual user ratings lie between 3 - 5 is advantageous since, in a generic recommendation engine we would want to recommend movies that a user would like and not otherwise! This also proves that RMSE is not always the best metric to evaluate the performance of recommendation engines since it penalizes errors on lower and higher ratings equally. If we, gave more weightage to the collaborative ratings as opposed to the global ratings for values where the global rating > 3 and lesser weightage to it in cases when global rating ≤ 2, we hope to get a better precdiction for the rating.

8 Conclusion and Future Work

One of the important observations is from the hybrid approach is that errors do *cancel out* when we combine the ratings derived from the collaborative and content-based approaches. This is consistent with our initial motivation for relative weighting of the three schems. Another important observation was that the collaborative approach gave a lesser RMSE when we increased the number of nearest neighbors from 10 to 100. Thus, we believe that even a larger set of neighbors, if considered, would give lesser RMSE values. Another interesting observation from our experiments is that giving a higher weightage to Content Based Rating in our final score calculation leads to lower RMSE values. Infact, we expected this bias towards content based ratings as is evident from Table 2. However this implies that either the influence of similar users on a given user's ratings is lesser than his own discretion while rating the movie, or that better features are needed to capture the notion of similarity between users and the influencing factors among them.

There were, however, some pieces of information that we did not utilize, the date of release of a movie and the date of rating of a user. In the spirit of the idea that more data leads to better results we feel that studying the variations in the rating patterns of a user over a period time might prove to be a useful feature in improving the prediction results. Moreover, building movie profiles using their release date as an additional feature (say, clustering together movies in a decade) might give us deeper insight into the tastes of the user. Keeping time as parameter also helps us to study the user profile in different time-windows.

We believe that we could have further improved our collaborative approach by incorporating the standard deviation of the ratings of a user along with his average ratings.

References

[1] Netflix challenge. `http://www.netflixprize.com/`.

[2] Netflix prize. `http://en.wikipedia.org/wiki/Netflix_Prize`.

[3] Gediminas Adomavicius and Er Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17, 2005.

[4] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *VLDB*, 1999.

[5] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002.

[6] Al Mamunur Rashid, George Karypis, and John Riedl. Influence in ratings-based recommender systems: An algorithm-independent approach. SDM, 2005.