Toward a Social Graph Recommendation Algorithm: Do We Trust Our Friends in Movie Recommendations?*

Ali Adabi and Luca de Alfaro**

University of California Santa Cruz, Computer Science Department
Santa Cruz, CA, USA, 95065
{aadabi,luca}@soe.ucsc.edu

Abstract. Social networks provide users with information about their friends, their activities, and their preferences. In this paper we study the effectiveness of movie recommendations computed from such communicated preferences. We present a set of social movie recommendation algorithms, which we implemented on top of the Facebook social network, and we compare their effectiveness in influencing user decisions. We also study the effect of showing users a justification for the recommendations, in the form of the profile pictures of the friends that caused the recommendation.

We show that social movie recommendations are generally accurate. Furthermore, 80% of the users that are undecided on whether to accept a recommendation are able to reach a decision upon learning of the identities of the users behind the recommendation. However, in 27% of the cases, they decide *against* watching the recommended movies, showing that revealing identities can have a negative effect on recommendation acceptance.

Keywords: Social Recommendation, Friend Network.

1 Introduction

Web 2.0 is about joining communities, and connecting people to each other through social networks. Web 2.0 is making available in computational form a throve of information about the preferences and behavior of users, as well as their social connections. Recommender systems leverage this information to provide personalized recommendations for users, helping them to discover content that they might find of interest [2,4].

Taste in movies is both sophisticated and personal: the reaction of a user to a movie is a complex outcome of movie features, user preferences, previous movies seen, and so forth, that is difficult to summarize in a few simple criteria.

* This Work was supported in Part by a Gift from Google, Inc.
** This author has a financial interest in Google, Inc.

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 637–647, 2012.

For instance, one might like *Usual Suspects* and dislike *Seven* even though both movies belong to the crime-thrillers. In order to provide users with meaningful recommendations, some algorithms rely on collaborative filtering and user reviews [9,18], while others rely on content-tags [6,22].

We analyze the problem of generating movie recommendation and investigate ways to leverage the social graph in producing relevant recommendations for users. In the past approaches (*e.g.* Netflix), the item recommendations are computed based on the taste similarity. We will explore in this work *social* recommendations that are a function also of what the friends of a user like or dislike. We believe social recommendations are particularly effective for movies since most people like to watch movies in company: thus, suggestions based on a user's friend network may also facilitate the task of finding friends with whom to watch recommended movies.

We have built a social movie recommendation site on top of the Facebook social network, and we have compared the efficacy of several social recommendation algorithms. Our study was motivated by the following two hypotheses:

– The knowledge of the social graph of the user and the preference of their friends will enable us to make precise predictions.
– Communicating to a user, in addition to the recommendation, also the list of friends on which the recommendation is based, increases the trust of the user in the recommendations, and increases the likelihood that the recommendations are followed.

The first hypothesis was first put forward in [3], and validated in experiments in [5], among others. We find that, for movies as well, the use of social information leads to high-quality recommendations. As for the second hypothesis, the results were somewhat surprising. On the one hand, we found that providing information on the friends that were the source of the recommendations leads users to place increased trust in the recommendations themselves, in line with [8]. Indeed, in 80% of instances, users who were undecided on whether to accept a movie recommendation were able to make up their minds once they learned the identities of the friends who liked the movies. Surpisingly, we found that in 27% of instances, undecided users actually decided *against* watching recommended movies upon learning the identities of the friends who liked them, showing how identity information can have also an adverse effect on recommendation acceptance. This indicates that "friends" in social network are well aware of their (real or perceived) difference in content tastes, and modeling such "acceptance similarity" explicitly could lead to better recommendations.

2 Collaborative Filtering

One approach in designing recommendation systems is Collaborative Filtering (CF). In CF, users are compared according to their similarity in past preferences, and recommendations for new items are built on the basis of the preferences of similar users [12,18,21].

We rely on CF algorithms that perform these two steps:

1. Calculate the similarity *sim(u,v)* between two users u and v, on the basis of preferences expressed by u and v.
2. Produce a prediction value p_u for user u on the basis of the preferences expressed by users similar to u.

Several approaches have been used to compute similarity between users. One of the most common approaches is *cosine-similarity*: the two users u and v are treated as two vectors in n-dimensional space, and the cosine of the angle across the two vectors is computed:

$$sim(u,v) = \frac{\sum_{i \in I} r_{u,i}\, r_{v,i}}{\sqrt{\sum_{i \in I} r_{u,i}^2}\sqrt{\sum_{i \in I} r_{v,i}^2}} \,. \tag{1}$$

where I is the set of items, $r_{u,i}$ is the rating of user u for item i, and $r_{v,i}$ is the rating of user v for item i.

Once user-to-user similarities are computed, recommendations are obtained by analyzing the preferences of users similar to the target user. We use neighborhood-based CF algorithms, where a subset of nearest neighbors (in the cosine-similarity metric) of a user is chosen, and a weighed average of the preferences of these nearest-neighbour users is used. Precisely, to make a prediction $p_{u,i}$ for user u on a certain item i, we take a weighted average of all the nearest-neighbour ratings on that item according to the following formula [13,18] :

$$p_{u,i} = \bar{r}_u + \frac{\sum_{v \in U_i} sim(u,v) \cdot (r_{v,i} - \bar{r}_v)}{\sum_{v \in U} sim(u,v)} \,. \tag{2}$$

where $\bar{r}_u$ and $\bar{r}_v$ are the average ratings for the user u and user v on all other rated items, and where $sim(u,v)$ is the similarity between the user u and user v. The summations are over all the users U_i who have rated the item i.

3 Recommendation via Friends

Social networking websites such as Facebook have became a prominent source of information sharing recently: for instance, Facebook had 901 million users in March 2012, and users shared over 3.2 billion "likes" per day on the site [7].

In this work, we rely on the Facebook API to get information on the social network of users, and on the movie preferences of the individual users (via the "likes" that users have expressed for the movies). On the basis of these expressed social connection and movie preferences, we compute personalized movie recommendations for users. We compare several algorithms for computing personalized preferences, as described below.

Algorithm 1. Basic Social Recommendation (BSR):

```
1: Input: A user u ∈ U.
2: For all movies i ∈ I, set c(i) = 0.
3: for all v ∈ F_u do
4:     for all i ∈ L(v) do
5:         c(i) := c(i) + 1
6:     end for
7: end for
8: Sort the movies list I in decreasing order of c(i), i ∈ I.
9: return Top movies in the ordering.
```

3.1 Friends Network Algorithms

Basic Social Recommendation (BSR) BSR finds the most frequent recommended movies by friends. For a user $u \in U$, denote by F_u the set of friends of u in the social network, and let $L(u)$ be the set of movies that u has liked.

BSR scans all movies users' friends liked, and generates a list of highly recommended movies among each user's social network.

General Stranger Recommendation (GSR). In this algorithm, we compute movie recommendations by tallying likes across all users. In our experimental setting, however, the total number of users is small, and for many users u we have that $U \setminus F_u$ is not much larger than F_u: that is, the friendship neighborhood of a user comprises a significant portion of the overall user base. In order to minimize the effect of a user's friends neighborhood on the recommendation and better model the behavior of a true friendship-independent algorithm in the large scale, we exclude from the recommendation the friends of each user. Thus, the GSR algorithm is identical to the BSR algorithm above, except that in Step 2, we consider all $v \in U \setminus F_u$ rather than $v \in F_u$.

Explanation Social Recommendation (ESR). Many recommender systems are providing no transparency into the working of the recommendation. Explanations provide that transparency, exposing the reasoning and data behind a recommendation. The ESR algorithm explains to the user the origin of the social recommendations by displaying the faces and names of the friends that caused a recommendation next to the recommendation itself.

Clustering Social Recommendation (CSR). The CSR algorithm identifies in the social network of a user other friends who are similar in their movie tastes, and uses the preferences of such similar friends to generate recommendations for the target user. Given a target user u and a friend v, we compute their similarity simply as:

$$sim(u, v) = \frac{|L(u) \cap L(v)|}{|L(v)|} .$$

CSR calculates $sim(u, v)$ for all friends v of u, and sorts these friends in order of decreasing similarity. CSR then recommends to u the movies that these friends have liked, but u has not seen yet.

Algorithm 2. Explanation Based Recommendation (ESR) using BSR:

Input: A user $u \in U$.
For all movies $i \in I$, set $c(i) = 0$.
for all $v \in F_u$ **do**
    for all $i \in L(v)$ **do**
        $c(i) := c(i) + 1$
    end for
end for
Let M be the list of movies in I, sorted in decreasing order of $c(i)$, for $i \in I$.
For each movie $i \in I$, consider the friends $F_i = \{v \in U \mid i \in L(v)\}$ that liked i.
return The list M, with the pictures and names of the friends in F_i for each movie $i \in M$.

Algorithm 3. Clustering Social Recommendation (CSR):

1: **Input:** A user $u \in U$
2: **for all** $v \in F(u)$ **do**
3:     Compute $sim(u, v)$
4: **end for**
5: Sort F_u in decreasing order of $sim(u, v), v \in F_u$, obtaining F'.
6: $M =$ empty list.
7: **for** $v \in F'$ **do**
8:     Append the movies that v liked, and u has not seen, to M.
9: **end for**
10: **return** M

Clustering Based Recommendation (CBR). CBR is a generic form of CSR: instead of going through user friends, CBR goes through all strangers (not-user-friends), similarly to how GSR relates to BSR. In particular, CBR differs from CSR in Line 2, where $v \in F_u$ is replaced by $v \in U \setminus F_u$. The CBR serves as a base case for CSR in identifying the benefits or shortcomings of social network recommendations.

4 Experiments on the Effectiveness of Social Algorithms: BSR, GSR, ESR, CSR, and CBR

A series of experiments have been implemented on the social algorithms BSR, GSR, ESR, CSR, and CBR. The test subjects were 20 college students ages between 20 to 28. The size of the movie database in this study was 5000 titles. Users were asked to login into a Facebook application using their credentials. Upon successful authentication, the Facebook application was able to extract user and his or her friends' movie information. Users were shown a series of recommendations (BSR, GSR, etc.) and their answers were recorded.

4.1 BSR vs. GSR

In this experiment, 5 GSR movies were combined randomly with 5 BSR movies and have been offered to the users. Users were asked to pick if they like or dislike any of the 10 recommendations provided(so the results are not skewed). The displayed results gave users equal chances of choosing BSR or GSR movies. This study tested if users liked BSR or GSR movie recommendations without prior knowledge of their friends movie preferences.

As figure 1 shows, a majority of users (83%) liked either BSR or GSR movies, with BSR edging GSR. It is notable to consider that 17% of the recommended movies were disliked by the users (or users found them irrelevant). The reason that the values of BSR and GSR in figure 1 are similar could be due to the fact that the user sample is uniformly chosen from college students (age 20-28) and perhaps a non-uniform sample could change the results toward the BSR recommendations.

The overall BSR vs. GSR results showed that since a user's friend are generally similar in terms of age, culture, and geographical location, an overlap in interests is visible. Although these interests are not uniform since friends are not uniform, the integration of all friends' interest can be generally used as the user's preference as well.

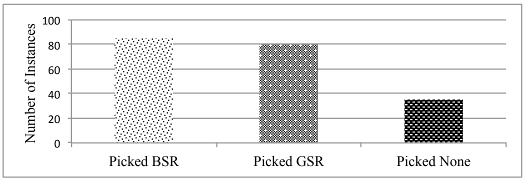

Fig. 1. BSR vs. GSR

Gender Attribute Applying a gender classification filter to the recommendations can lead to more personalized results. The gender filter was applied to the recommended movies which eliminated opposite sex recommendation and resulted figure 2.

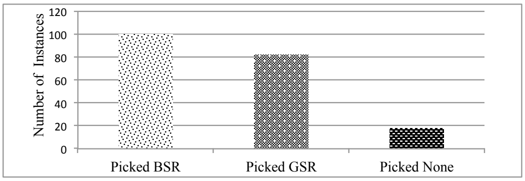

Fig. 2. BSR vs. GSR after applying a gender filter (Number of total instances was 200)

Figure 2 shows that adding a gender classifier can enhance the recommendations performance however anomalies can also be created. One of the visible issue was the elimination of some independent and not mainstream movies, which have been discarded, in the recommended movies specifically in the male participants results.

4.2 CSR vs. CBR

CSR finds people with similar taste within the friends' network. The experimentation was conducted to compare the effectiveness of recommendations of similar users between two distinct groups: Friends of a users (CSR) and strangers (CBR). Figure 3 shows the CSR and CBR recommendation comparison among users.

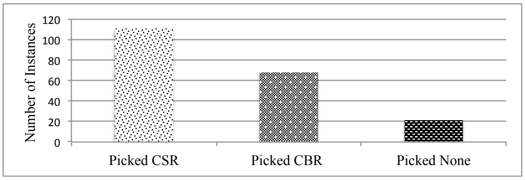

Fig. 3. CSR vs. CBR for 200 instances

Users preferred CSR results 21% more than CBR. This is because if a user u has a friend v, u already has some similarities to v, which may be cultural, geographical, or age-related. Searching the global network for similar users with CBR resulted in quality outcome as well. In our experiment, the overall number of users is small, but in large user bases, there is a higher probability of finding good matches for users (at the cost of an increased amount of computation).

4.3 ESR

In a separate experiment, the ESR algorithm was compared with BSR. ESR is similar to BSR, except that it also shows the recommenders' faces along with the results. This experiment measures the effect on movie recommendation acceptance of showing the identities of the friends on which the recommendations are based. We performed two experiments, termed A and B:

- In Experiment A, we studied the effect of adding, below the recommendations, the identities of 5 friends that contributed to the recommendations. When more than 5 friends supported a recommendation, we selected 5 such friends at random. Movies with less than 5 supporting friends were excluded from the experiment.
- In Experiment B, we proceeded as in experiment A, but we also we studied the effect of varying the number of faces displayed alongside the recommendations.

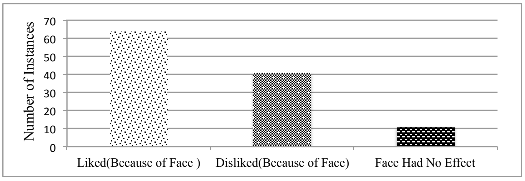

Fig. 4. ESR with constant face count effect calculated for 200 intances

Experiment A: Fixed Number of Recommenders. Figure 4 clearly illustrates that showing the identities of friends supporting a recommendation had a strong effect on people's decision to watch a movie. However, to our surprise, the effect was not all in the positive direction: in many cases, users were actually put off from the recommendation upon learning the identities of the friends on which the recommendation was based. Evidently, the trust that people may feel for their Facebook friends does not blankly extend to their taste for movies! Participants in our study valued a group of friends they recognized as individuals with great taste, or "tastemakers". When users saw the selected group of tastemakers recommended a specific movie, they instantly wanted to follow other movies they suggested.

Experiment B: Variable Number of Recommenders. In Experiment B, we varied the number of friends displayed next to the recommended movies from 1 to 10. We asked users if they would like to watch a movie they have not watched because a number of their friends recommended it.

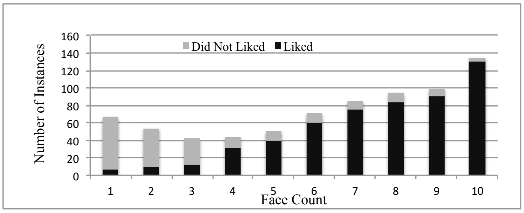

Fig. 5. ESR Face Count Effect

As figure 5 shows, recommender numbers has a direct effect on the user decision. This could be because the majority of the movies that have been liked by users, are in fact popular movies among users social network and therefore are relevant to the users. In recommendations with lower number of recommenders (ex. for 3 recommenders), when some of the users saw a specific friend watched the movie they disliked it. This is in par with the assumption made in Experiment A. However once the face count of the recommenders increase and pass a threshold, users tend to prefer the movies with higher count of friends faces.

5 ESR vs. BSR

For this experiment, the target user is given 10 BSR movie recommendations. Once the user has examined the recommendations, the user is showed the faces of the users that contributed to the recommendations, and is asked whether she is able to make a decision in the cases where she was undecided.

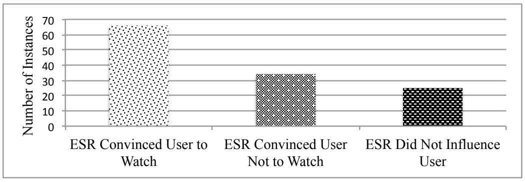

Fig. 6. *ESR* influence for 200 instances

As figure 6 shows, 80% of the users that were undecided on whether to watch a movie were able to reach a conclusion upon learning the identities of the users that were at the root of the recommendation. This means the explanation based recommendation has an important effect on users decision. However, as figure 6 illustrates, in 27% of total of cases, users decided not watch a movie after knowing some of their friends watched a movie. This exposes a new side of social recommendation, which is their negative effect on user's decision. We remarked that this negative effect is more significant for movies below a certain threshold of popularity. For little-known movies, if the user is doubtful, the user has roughly a one-in-three chance of deciding not to watch the movie when made aware of the identities of friends who liked the movie.

6 Conclusion

Our results show that social movie recommendations are effective in helping users decide which movies to watch — and, if the friend identities underlying the recommendations are revealed, also in helping users decide which movies *not* to watch. These results were partially expected, based on previous work by Suri and Watts [17]. This can be explained by the fact that users trust or distrust in some of their friends' taste as suggested by Jøsang, et al. [11] (correlation between taste and trust).

One of the problems with the current methodology of collaborative filtering is that when a new movie comes out, there is not sufficient rating data about the movie. A particular movie that a friend likes has a higher chance of being recognized by similar users. Abdul-Rahman and Hailes [1] showed that in a predefined context, such as movies, users develop social connections with people who have similar preferences. Ziegler and Lausen[23] extended these results in work that showed a correlation exists between trust and user similarity in an empirical study of a real online community. Clearly, recommendations only make sense when obtained from like-minded people [23].

Social recommendations used in the studies above show that the overall advantage of this method is relevancy of the result and trust that user have established with the recommender through their social network. As *Papagelis et al.* [20] states, retrieving data from the user and his social graph intrinsically alleviates the sparsity problem. Another advantage of social recommendation algorithms, over general collaborative filtering, is that these algorithms need to examine only the social neighbourhood of a user to produce recommendations for that user. Under the assumption that the size of such social neighbourhood is constant (very few users have over a few hundred friends), this leads to a computationally more tractable problem, compared to algorithms that scour the global set of users for users with similar tastes. Nevertheless, social movie recommendations face challenges regarding insufficiency of data, data normalization, and cold start:

- Insufficiency of data: Some users may be part of a sparse social networks, or they may be unwilling to disclose movie preferences, making it difficult to compute recommendations.
- Data normalization: Some users might not declare their interests in their profile, while others might overly exaggerate about their interest, creating anomalies.
- Cold Start: Users might not initially like a lot of movies this problem can be resolved if the user inputs movies they liked into their Facebook profile.

Future Work

The approaches outlined in this paper can be extended in several directions. The collaborative-filtering approach taken in this paper could be combined with a content-based approach, where movies are represented by feature vectors, and movie similarity is considered alongside user similarity. To alleviate the cold-start and data-insufficiency problems, a learning algorithm could be developed to identify the movie tastemakes among groups of friends, since tastemakers have a huge effect on decision of users.

References

1. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Hawaii International Conference on System Sciences. Maui (2000)
2. Adomavicius, G., Tuzhili, A.: Towards the Next Gen. of Recommender Systems. IEEE Transactions on Knowledge and Data Engineering 17, 634–749 (2005)
3. Asch, S.E.: Opinions and social pressure. Scientific American 193, 31–35 (1955)
4. Burke, R.: Hybrid Recommender Systems. User Mod. and User-Adap 12, 331–370 (2002)
5. Centola, D.: The Spread of Behavior in an Online Social Network Experiment. Science 329, 1194–1197 (2010)

6. Durao, F., Dolog, P.: A Personalized Tag-Based Recommendation in Social Web Systems. In: Intelligent Web and Information Systems.CoRR (2012)
7. Key Facts Statistics, http://newsroom.fb.com/content/default.aspx?NewsAreaId=22
8. Golbeck, J.:Generating Predictive Movie Recommendations from Trust in Social Networks, iTrust (2006)
9. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: ACM Conference on Computer Supported Cooperative Work, pp. 241–250. ACM, New York (2000)
10. Huang, Z., Zeng, D., Chen, H.: A Link Analysis Approach to Recommendation under Sparse Data. In: Americas Conference on Information Systems (2004)
11. Jøsang, A., Quattrociocchi, W., Karabeg, D.: Taste and Trust. In: Wakeman, I., Gudes, E., Jensen, C.D., Crampton, J. (eds.) Trust Management V. IFIP AICT, vol. 358, pp. 312–322. Springer, Heidelberg (2011)
12. Resnick, P., Lacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An Open Architecture for Collaborative Filtering. In: ACM Conference on Computer Supported Cooperative Work, New York (1994)
13. Ricci, F., Rokach L.: Recommender Systems Handbook. Springer, New York (2011)
14. Sarwar, B.M., Karypis, G., Onstan, J.A., Riedl, J.: Recommender Systems for Large-scale E-Commerce. In: ICCIT (2002)
15. Social Recommender Systems Methods and User Issues, http://hci.epfl.ch/teaching/advanced-hci/slides/2011.5.23_Yu.pdf
16. Smith, B., Briggs, P., Coyle, M., O'Mahony, M.: Google Shared.A Case-Study in Social Search. In: International Conference, UMAP, 17 (2009)
17. Suri, A., Watts, D.J.: Cooperation and Contagion in Web-Based, Networked Public Goods Experiments. PLoS ONE, 6 e16836 (2011).
18. Su, X., Khoshgoftaar, T.: A Survey of Collaborative Filtering Techniques. In: Advances in Artificial Intelligence, 17 (2009)
19. Pazzani, M.J.: A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Review, 393–408 (1999)
20. Papagelis, M., Plexousakis, D., Kutsuras, T.: Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. iTrust, Heraklion (2005)
21. Melville, P., Sindhwani, V.: Recommender Systems. Encyclopedia of Machine Learning, New York (2010)
22. Vatturi, P.K., Geyer, W., Dugan, C., Muller, B.B.: Tag-based ltering for personalized bookmark recommendations. In: 17th ACM Conference on Information and Knowledge Mining, pp. 1395–1396. ACM, New York (2008)
23. Ziegler, C., Lausen, G.: Analyzing Correlation Between Trust and User Similarity in Online Communities. In: International Conference on Trust Management, vol.(2) (2004)