A Socially Augmented Hybrid Recommendation Engine

A PROJECT REPORT

Submitted By

Shivendra Soni (IIT2010027)
Prateek Porwal (IIT2010035)
Kshitij Mittal (IIT2010040)

Under the guidance of

Dr. Sonali Agarwal

in partial fulfillment for the award of the degree

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

ALLAHABAD

November, 2012

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Certificate

This is to certify that the dissertation entitled “A Socially Augmented Hybrid Recommendation Engine” submitted by Shivendra Soni (IIT2010027), Prateek Porwal(IIT2010035), Kshitij Mittal (IIT2010040) is approved for the award of Degree - Bachelor of Technology in Information Technology. We hereby declare that it is an authenticated record of our original work carried out from July to November, 2012 under the guidance of Dr. Sonali Agarwal. Due acknowledgements have been made in the text to all other material used.

Place:
IIIT Allahabad
Jhalwa Campus
Date: 29th November, 2012

Project Mentor
Dr. Sonali Agarwal

Acknowledgements

We would like to convey our deepest gratitude to Dr. Sonali Agarwal, who guided us through this project. Her keen inspiring personality, admirable guidance, and constant encouragement are the motive force behind this project work.

Shivendra Soni
(IIT2010027)

Prateek Porwal
(IIT2010035)

Kshitij Mittal
(IIT2010040)

CONTENTS

1. ABSTRACT

Social influence plays an important role in product marketing. However, it has rarely been considered in traditional recommender systems. In this report we present a new paradigm of recommender systems which can utilize information in social networks, including user preferences, item's general acceptance, and influence from social friends. We extract data from a real online social network, and our analysis of this dataset reveals that friends have a tendency to select the same items and give similar ratings. An experimental result on this dataset shows slight improvement in the prediction accuracy of recommender systems and also remedies the data sparsity and cold-start issues inherent in collaborative filtering. Such technologies can be deployed by most content providers.

2. INTRODUCTION

In order to overcome information overload, recommender systems have become a key tool for providing users with personalized recommendations on items such as movies, music, books, news, and web pages. Intrigued by many practical applications, researchers have developed algorithms and systems over the last decade. There are typically two types of algorithms for recommender systems -- content-based methods and collaborative filtering. Content-based methods measure the similarity of the recommended item (target item) to the ones that a target user (i.e., user who receives recommendations) likes or dislikes based on item attributes. On the other hand, collaborative filtering finds users with tastes that are similar to the target user's based on their past ratings. Collaborative filtering will then make recommendations to the target user based on the opinions of those similar users. Despite all of these efforts, recommender systems still face many challenging problems. For example, in order to measure item similarity, content-based methods rely on explicit item descriptions. However, such descriptions may be difficult to obtain for items like ideas or opinions. Collaborative filtering has problems of data sparsity and the cold-start problem [1].

3. <u>MOTIVATION</u>

To tackle those problems, two approaches have been proposed [3]. The first approach is to condense the user/item rating matrix through dimensionality reduction techniques such as Singular Value Decomposition (SVD) [3]. By clustering users or items according to their latent structure, unrepresentative users or items can be discarded, and thus the user/item matrix becomes denser. However, these methods do not significantly improve the performance of recommender systems, and sometimes make the performance even worse.

In this paper we try to solve these problems from a different perspective. In particular, we propose a new paradigm of recommender systems by utilizing information in social networks, especially that of social influence.

When friends recommend a product to us, we also tend to accept the recommendation because their inputs are trustworthy.

Additionally, the integration of social networks can theoretically improve the performance of current recommender systems. First, in terms of the **prediction accuracy**, the additional information about users and their friends obtained from social networks improves the understanding of user behaviors and ratings. Second, with friend information in social networks, it is no longer necessary to find similar users by measuring their rating similarity, because the fact that two people are friends already indicates that they have things in common. Thus, the **data sparsity** problem can be alleviated. Finally, for the **cold-start issue**, even if a user has no past reviews, recommender system still can make recommendations to the user based on the preferences of his/her friends if it integrates with social networks.

In our research, we are interested in the role of explicit social relations in recommender systems, such as how user preferences or ratings are correlated with those of friends, and how to use such correlations to design a better recommender system. In particular, we design an algorithm framework which makes recommendations based on user's own preferences, the general acceptance of the target item, and the opinions from social friends.

4. METHODOLOGY

A Socially Augmented Hybrid Recommender Engine (SAHRE)

The decisions that we make in our daily life, such as finding restaurants, buying a house, and looking for jobs, many of them are actually influenced by these three factors. Figure 4.1 further illustrates how these three factors impact customers' final buying decisions. Intuitively, users buying decision (or rating) is decided by both his/her own preference for similar items and his/her knowledge about the characteristics of the target item. Knowledge about the target item can be obtained from public media such as magazines, television, and the Internet. Meanwhile, the feedbacks from friends are another source of knowledge regarding the item, and they are often more trustworthy than advertisements. When a user starts considering the feedbacks from his/her friends, he/she is then influenced by them. Note that this influence is not limited to that from our immediate friends. Distant friends can also cast their influence indirectly to us. The techniques for handling these types of influences are different. We shall find the immediate friend inference, in which we only consider influences from immediate friends. (Distant friend currently beyond the scope of the project).

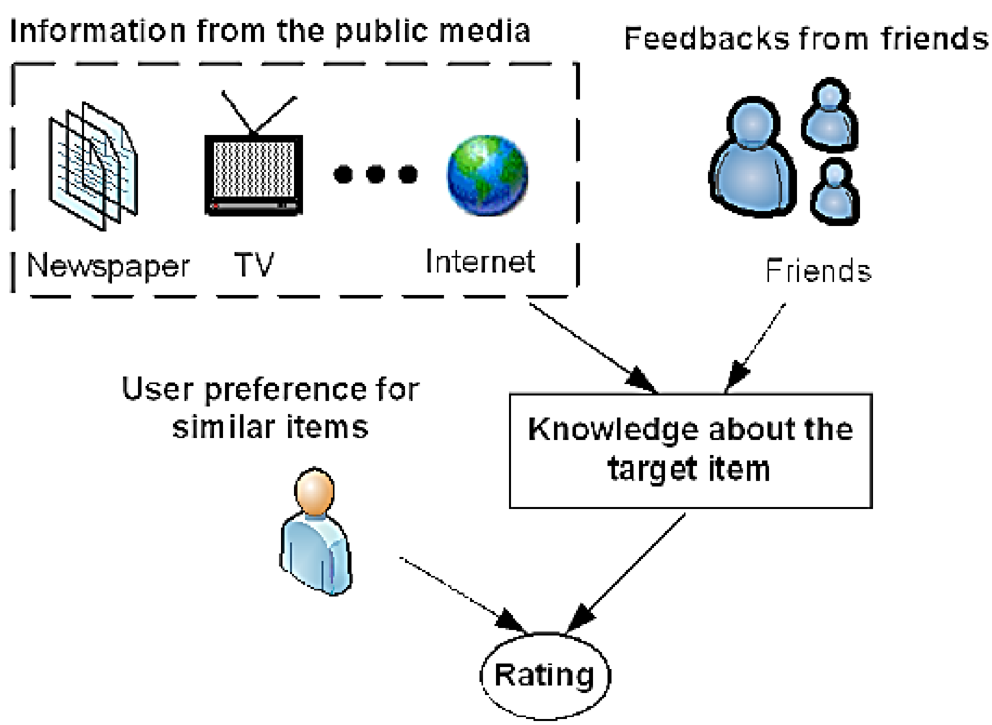

Figure 4.1: The three factors that influence a customer's buying decision: user preference for similar items, information regarding the target item from the public media, and feedbacks from friends.

4.1 Immediate Friend Inference

Formally, let us consider a social network as a graph $G = \boldsymbol{(U, E)}$ in which $\boldsymbol{U}$ represents nodes (users) and $\boldsymbol{E}$ represents links (social relations). Each user U in **U** has a set of attributes A_U as well as immediate neighbors (friends) ***N(U)*** such that if $V \in \boldsymbol{N}(U)$, $(U, V) \in \boldsymbol{E}$**.** The values of attributes A_U are represented as $\boldsymbol{a_U}$. Moreover, a recommender system contains the records of users' ratings, which can be represented by a triple relation of $T = (U, I, R)$ in which U is the users in the social network G; I is the set of items (products or services), and each item I in I has a set of attributes A'_I. $\boldsymbol{R}$ stands for the ratings such that each R_{UI} in R is user U's rating on item I. $\boldsymbol{R_{UI}}$ has a numeric value k *(e.g.* $k \in \{1, 2, \ldots 5\}$*).* Moreover, we define $I(U)$ as the set of items that user U has reviewed, and refer to the set of reviewers of item I as $U(I)$. The goal of this recommender system is to predict $\boldsymbol{Pr(R_{UI} = k \mid A'=a'_I, A=a_U, \{R_{VI} = r_{VI} : \forall V \in U(I) \cap N(U)\})}$; i.e., the probability distribution of the target user U's rating on the target item I given the attribute values of item I, the attribute values of user U, and the ratings on item I rated by U's immediate friends. Once we obtain this distribution, R_{UI} is calculated as the expectation of the distribution. Items with high estimated ratings will be recommended to the target user, and users with high estimated ratings on the target item are the potential buyers.

In order to estimate $Pr(R_{UI} = k \mid A'=a'_I, A=a_U, \{R_{VI} = r_{VI} : \forall V \in U(I) \cap N(U)\})$, we adopt the naive Bayes assumption which assumes that the influences from item attribute values, user attribute values, and immediate friends' ratings are independent.. By making this assumption, the original conditional probability can be factorized as follows,

$$\begin{aligned} &\Pr(R_{UI} = k \mid \boldsymbol{A'} = \boldsymbol{a'}_I, \boldsymbol{A} = \boldsymbol{a}_U, \{R_{VI} = r_{VI} : \forall \boldsymbol{V} \in \boldsymbol{U}(I) \cap \boldsymbol{N}(U)\}) \\ &\quad = \frac{1}{Z} \Pr(R_{UI} = k \mid \boldsymbol{A'} = \boldsymbol{a'}_I) \times \Pr(R_{UI} = k \mid \boldsymbol{A} = \boldsymbol{a}_U) \times \\ &\quad \Pr(R_{UI} = k \mid \{R_{VI} = r_{VI} : \forall \boldsymbol{V} \in \boldsymbol{U}(I) \cap \boldsymbol{N}(U)\}) \end{aligned} \tag{3}$$

First, $Pr(R_U = k \mid A' = a'_I)$ is the conditional probability that the target user U will give a rating k to an item with the same attribute values as item I. This probability represents U's preference for items similar to I.

Second, $Pr(R_I = k \mid A = a_U)$ is the probability that the target item I will receive a rating value k from a reviewer whose attribute values are the same as U. This probability reflects the general acceptance of the target item I by users like U.

Finally, $Pr(R_{UI} = k \mid \{R_{VI} = r_{VI} : \forall V \in U(I) \cap N(U)\})$ is the probability that the target user U gives a rating value k to the target item I given the ratings of U's immediate friends on item I. This is where we actually take social influences into consideration in our system. We shall present the methods to estimate each of the factors in the following subsections.

4.1.1 User Preference

As we pointed out, $\Pr(R_U = k \mid A' = a'_I)$ measures the target user U's preference for the items similar to item I. $\Pr(R_U = k \mid A' = a'_I)$ gives us a hint of how likely it is that user U will give a rating k to a an item I which has attributes a'_i. To estimate this probability, we adopt the naive Bayes assumption again. We assume that the item attributes in A', e.g., category, are independent of each other.

$$\Pr(R_U = k \mid A' = a'_I) = \frac{\Pr(R_U = k) \times \Pr(A'_1, A'_2, \ldots, A'_n \mid R_U = k)}{\Pr(A'_1, A'_2, \ldots, A'_n)} \tag{4}$$

$$= \frac{\Pr(R_U = k) \times \prod_{j=1}^{j=n} \Pr(A'_j \mid R_U = k)}{\Pr(A'_1, A'_2, \ldots, A'_n)}, A' = \{A'_1, A'_2, \ldots, A'_n\}$$

where $\Pr(A'_1, A'_2, \ldots, A'_n)$ can be treated as a normalizing constant, Pr(RU = k) is the prior probability that U gives a rating k, and $Pr(A'_J \mid R_U = k)$ is the conditional probability that each item attribute A'_j in A' has a value a'_j given U rated k; e.g., *Pr(category= american | RU = 4)* is the probability that a restaurant will be of American cuisine, given that U gives a rating 4. The last two probabilities can be estimated from counting the review ratings of the target user U. Specifically,

$$\Pr(R_U = k) = \frac{|I(R_U = k)| + 1}{|I(U)| + n}, \text{ and} \tag{5}$$

$$\Pr(A'_j = a'_j \mid R_U = k) = \frac{|I(A'_j = a'_j, R_U = k)| + 1}{|I(R_U = k)| + m}, \tag{6}$$

where |I(U)| is the number of reviews of user U's in the training set, |I(RU = k)| is the number of reviews that user U gives a rating value k, and *|I (A'j = a'j, RU = k)|* is the number of reviews that U gives a rating value k while attribute A'j of the corresponding target item has a value a'_j. Notice that we insert an extra value 1 to the numerators in both equations, and add n, the range of review ratings to the denominator in Eq. (5), and m, the range of A'j's values, to the denominator in Eq. (6). This method is also known as Laplace estimate, a welknown technique in estimating probabilities [7], especially on a small size of training samples. Because of Laplace estimate, "strong" probabilities, like 0 or 1, from direct probability computation can be avoided.

Moreover, in some cases when item attributes are not available, we can approximate $Pr(R_U = k \mid A' = a'_I)$ by the prior probability $Pr(R_U = k)$. Even though $Pr(R_U = k)$ does not contain information specific to certain item attributes, it does take into account U's general rating preference; e.g., if U is a generous person, he/she gives high ratings regardless of the items.

4.1.2 <u>Item Acceptance</u>

$Pr(R_I = k \mid A = a_u)$ captures the general acceptance of item I from users like user U. Similar to the estimation in user preference, we use the naive Bayes assumption and assume user attributes are independent. Thus, we have

$$\Pr(R_I = k \mid A = a_U) = \frac{\Pr(R_I = k) \times \Pr(A_1, A_2, \ldots, A_m \mid R_I = k)}{\Pr(A_1, A_2, \ldots, A_m)} \tag{7}$$

$$= \frac{\Pr(R_I = k) \times \prod_{j=1}^{j=m} \Pr(A_j \mid R_I = k)}{\Pr(A_1, A_2, \ldots, A_m)}, A = \{A_1, A_2, \ldots, A_m\}$$

which $Pr(R_I = k)$ is the prior probability that the target item I receives a rating value k, and $Pr(A_j \mid R_I = k)$ is the conditional probability that user attribute Aj of a reviewer has a value of a_j given item I receives a rating k from this reviewer. These two probabilities can be learned by counting the review ratings on the target item I in a manner similar to what we did in learning user preferences. When user attributes are not available, we use $Pr(R_I = k)$, i.e., item I's general acceptance regardless of users, to approximate $Pr(R_I = k \mid A = a_u)$. In addition, $Pr(A1, A2, ..., Am)$ in Eq. (7) is a normalizing constant.

4.1.3 <u>Influence from Immediate Friends</u>

Finally, $Pr(R_{UI} = k \mid \{R_{VI} = r_{VI} : \forall V \in U(I) \cap N(U)\})$ is where SAHRE utilizes the influences from immediate friends. To estimate this probability, SAHRE learns the correlations between the target user U and each of his/her immediate friends V from the items that they both have rated previously, and then assume each pair of friends will behave consistently on reviewing the target item I too. Thus, U's rating can be estimated from r_{VI} according to the correlations. A common practice for learning such correlations is through estimating user similarities or coefficients, either based on user profiles or user ratings. However, user correlations are often so delicate that they cannot be fully captured by a single similarity or coefficient value. It is even worse that most of those measures seem ad hoc. Different measures return different results, and have different conclusions

on whether or not a pair of users is really correlated . To another extreme, user correlations can be also represented in a joint distribution table of U's and V's ratings on the same items that they have rated; i.e., $Pr(R_{UI}, R_{VI})\ \forall I \in I(U) \cap I(V)$.
This table fully preserves the correlations between U's and V's ratings. However, in order to build such a distribution with accurate statistics, it requires a large number of training samples. This is especially a problem for recommender systems, because in most of these systems, users only review a few items compared to the large amount of items available in the system, and the co-rated items between users are even less. Therefore, in this study, we use another approach to remedy the problems in both cases.

Thus, $H(R_{UI} – R_{VI})$ serves as the correlation measure between U and V. For rating ranges from one to five, $H(R_{UI} – R_{VI})$ is a distribution of nine values, i.e. from -4 to 4. Compared to similarity measures, it preserves more details in friends' review ratings. Compared to a joint distribution approach, it has fewer degrees of freedom.

Assuming U's and V's rating difference on the target item I is consistent with $H(R_{UI} – R_{VI})$. Therefore, when RVI has a rating r_{VI} on the target item, the probability that R_{UI} has a value k is proportional to $H(k - r_{VI})$.

$$\Pr(R_{UI} = k \mid R_{VI} = r_{VI}) \propto H(k - r_{VI}). \tag{9}$$

For example, assume that both U and V rated the items as shown in Table 3.1. Given their ratings in the table, we want to predict U’s possible ratings on item I_6 according to the correlation with V. From the previous ratings of U and V, we find out that two out of five times U’s rating is the same as V’s, and three out of five times U’s rating is lower than V’s by one. According to such a correlation, we predict that there is a 40% chance that RUI6 is 4 and 60% chance that R_{UI6} is 3.

	U	*V*
I_1	5	5
I_2	3	4
I_3	4	4
I_4	2	3
I_5	4	5
I_6	?	4

Table 4.1: An example of predicting user rating from an immediate friend

The previous example illustrates how we utilize the correlation between the target user and one of his/her immediate friends. When the target user has more than one immediate friend who co-rates the target item, the influences from all of those friends can be incorporated in a product of normalized histograms of individual friend pairs.

$$\Pr(R_{UI} = k \mid \{R_{VI} = r_{VI} : \forall V \in U(I) \cap N(U)\} = \frac{1}{Z} \prod_{V} \frac{1}{Z_V} H(k - r_{VI}) \qquad (10)$$

where Z_V is the normalizing constant for the histogram of each immediate friend pair, and Z is the normalizing constant for the overall product.

Once we obtain $Pr(R_U = k \mid A' = a'_I,)$, $Pr(R_I = k \mid A = a_u)$, and $Pr(R_{UI} = k \mid \{R_{VI} = r_{VI} : \forall V \in U(I) \cap N(U)\})$, the ultimate rating distribution of R_{UI}, under the factors of user preference, item's general acceptance, and the correlations with immediate friends, can be estimated from Eq. (3). R'_{UI}, the estimated value of R_{UI}, is the expectation of the distribution as shown in Eq. (11).

$$R'_{UI} = \sum_{k} k \times \Pr(R_{UI} = k \mid A' = a'_I, A = a_U, \{R_{VI} = r_{VI} : \forall V \in U(I) \cap N(U)\} \qquad (11)$$

5.DATASET

In this section, we introduce the dataset that we use for this research, and present some characteristics of this dataset. Our dataset is obtained from a real online social network Yelp.com, a website that provides local search for restaurants, spas, auto services etc along with their attributes.
Users that come to this site can either look for information from Yelp or make their own voices by writing reviews for some local commercial entities that they have experienced. The homepage of each business contains a list of reviews of users who have visited this restaurant before and a numerical rating ranging from one to five stars. Five starts means the highest rating to this restaurant, and one star is the lowest rating.
Besides maintaining traditional features of recommender systems, Yelp provides social network features so that it can attract more users.. The friendship at Yelp is mutual relationship and non-weighted relationship, which means that when a user adds another user as a friend, the first user will be automatically added as a friend of the second user.. Each user homepage contains basic personal information, all the reviews written by this user, and *links to the friends* that are explicitly identified by this user.

For our experiment we picked the restaurant problem domain. We crawled and parsed the hompages of top **80** restaurants in the New York City area. By following the reviewers' links in the Yelp restaurant homepages, we also crawled the homepages of the **40** latest reviewers in each of the restaurants, which resulted in **3200 users**.

Based on the friend links in each user's homepage, we are able to identify friends from the crawled users, and thus reconstruct a social network. Note that the friends we collected for each user may only be a subset of the actual friends listed on his/her homepage. That is because we require every user in our dataset to have a least one review in the crawled restaurants

In terms of friends, the average number of immediate friends of every user is 8.18. Because most users on Yelp review only a few restaurants, we expect the dataset to be extremely sparse. In fact, the sparsity of this dataset, i.e., the percentage of user/item pairs whose ratings are unknown, is 99.86%.

Furthermore, we perform the following analysis on this dataset, particularly focusing on immediate friends' review correlation and rating correlation. Basically, we want to answer two questions: 1) whether or not friends tend to review the same restaurant; and 2) whether or not friends tend to give ratings that are more similar than those from non-friends. Clearly, these two questions are essential to SAHRE.

5.1 Review Correlation of Immediate Friends

Let us first study the correlation of immediate friends in reviewing the same restaurants. Specifically, we want to know if a user reviews a restaurant, what is the chance that at least one of his/her immediate friends has also reviewed the same restaurant? To answer this question, we count, for each user, the percentage of restaurants that has also being reviewed by at least one of his/her immediate friends. The average percentage over all users in the dataset is 18.6% (in the considered study).
As a comparison, we calculate the same probability if assuming immediate friends review restaurants uniformly at random and independently. In a social network with ***n*** users, for a user with **q** immediate friends and a restaurant with m reviewers (including the current user), the chance that at least one of **q** immediate friends appears in m reviewers is: $1-({}^{n-q-1}C_{m-1})/({}^{n-1}C_{m-1})$ .We calculate this value for every user and every restaurant he/she reviewed. The average probability over all users, according to the considered study is only 3.7%.

Finally, we compare the average number of co-reviewed restaurants between any two immediate friends and any two users on Yelp. The results are 0.85 and 0.03 respectively, which again illustrates the tendency that immediate friends co-review the same restaurants.

5.2 Rating Correlation of Immediate Friends

To show that whether immediate friends tend to give more similar ratings than non-friends, we compare the average rating differences (in absolute values) on the same restaurant between reviewers who are immediate friends and non-friends. We find that, for every restaurant in our dataset, if two reviewers are immediate friends, their ratings on this restaurant differ by *0.88* on average with a deviation of *0.89*. If they are not, their rating difference is *1.05* and the standard deviation is *0.98*. This result clearly demonstrates that immediate friends, on average, give more similar ratings than non-friends.

In this section we presented some characteristics of our dataset. The results on review correlations as well as rating correlations between immediate friends are critical in validating our assumptions in SAHRE.
In the next section, we are going to present a set of experiments to demonstrate the advantages of considering social network information in a recommender system.

6. EXPERIMENTS

In the experiments we evaluate the performance of SAHRE on the Yelp dataset, focusing on the issues of the prediction accuracy, data sparsity, and cold-start, which are the main issues of current recommender systems. Additionally, we will study the role of distant friends in SAHRE.

The following is the setting for our experiments. We used a restaurant's price range as the item attribute. Since there is no useful user attribute on Yelp, we substituted $Pr(R_I = k \mid A = a_u)$ *with* $Pr(R_I = k)$ when estimating item acceptance. Finally, we set a threshold to require every pair of immediate friends to have at least three co-rated restaurants. If they do not, we ignore their friend relationships.

6.1 Comparision Methods :

Friend Average (FA) To leverage the ratings of friends for inference, the most straightforward approach is to predict the ratings of the target users on the target items with the average ratings of their immediate friends on the same item. We therefore implemented this method as a baseline.

Collaborative Filtering (CF) We implemented the standard collaborative filtering algorithm as we described in the mid-semester report. The K value we used is 20.

6.2 Prediction Accuracy And Coverage

We carried out this experiment in a 10-fold cross-validation. The prediction accuracy was measured by the mean absolute error (MAE), which is defined as the average absolute deviation of predictions to the ground truth data over all the in-stances, i.e., target user/item pairs, in the testing set.

$$MAE = \frac{\sum_{U,I} \left| r_{UI} - r'_{UI} \right|}{L},$$

Where L is the number of testing instances. The smaller the MAE, the better is the inference.

Another metric that we study for each method is the **coverage**, which is defined as the percentage of the testing in-stances for which the method can make predictions.

	MAE
SAHRE	0.947
CF	0.991

Table 5.1: Comparison of the MAEs of the proposed Social Network-Based Recommender System (SAHRE), Collaborative Filtering (CF), in a 10-fold cross-validation.

	COVERAGE
SAHRE	0.482
CF	0.552

Table 5.2 Comparison of Coverage of SAHRE and CF as per the Followed Study

6.3 Sparcity and Coldstart

CF suffers from problems with sparse data. In this study, we want to evaluate the performance of SAHRE at various levels of data sparsity. To do so, we randomly divided the whole user/item pairs in our dataset into ten groups, and then randomly selected n sets as the testing set, and the rest as the training set. The value of n controls the sparsity of the dataset. At each value of n, we repeated the experiment 100 times. The performance was measured by the average MAEs and the cover-age.
We recorded that the MAE of CF increases at a much rapid rate as compared to SAHRE with increase in data sparsity under same conditions.
The coverage of both methods severely drops as the training set becomes sparser. CF performs better with a large training set, allowing it to find more similar users. When the training set becomes sparser, CF finds similar users from fewer candidates for each target user. The similarity obtained from each pair of users is less accurate because that there are fewer co-rated items between these users.

Thus, both the prediction accuracy and the coverage of CF are adversely affected by the data sparsity. Meanwhile, the coverage of SAHRE also decreases because there are fewer friends who have ratings on the target items as the dataset becomes sparser. But the coverage of SAHRE decreases with a slower pace compared to that of CF.

The prediction accuracy of SAHRE is consistent at all levels of data sparsity. This is because friends are provided explicitly by social networks, and there is no need for SAHRE to find similar users from the training set. Therefore, as long as there are friends who have reviewed the target item, SAHRE can make accurate predictions.

Cold-start is an extreme case of data sparsity where a new user has no reviews. In such a case, CF cannot make a recommendation to this new user since CF is not able to find similar users for him/her. SAHRE cannot either if this new user has also no friends.

However, in some cases of cold-start when a new user is invited by some existing users in the system, the initial friend relationships of this new user can still make the inference of SAHRE possible. Even though there is no prior knowledge of the new user's own preference, SAHRE can make recommendations to this new user based on the preferences of his/her friends.

7. CONCLUSION

Social networks provide an important source of information regarding users and their interactions. This is especially valuable to recommender systems. In this paper we presented a Socially Augmented Recommender System (SAHRE) which makes recommendations by considering a user's own preference, an item's general acceptance and influence from friends. In particular, we proposed to model the correlations between immediate friends with the histogram of friend's rating differences. The influences from distant friends are also considered in an iterative classification. In addition, we have collected data from a real online social net-work. The analysis on this dataset reveals that friends have a tendency to review the same restaurants and give similar ratings.

We compared the performance of SAHRE with other methods, such as collaborative filtering (CF), friend average (FA). In terms of the prediction accuracy, SAHRE achieves the best result. It yields a improvement compared to that of CF. In the sparsity test, SAHRE returns consistently accurate predictions at different values of data sparsity.

The coverage of SAHRE decreases when the data is sparse but at a slower speed than CF. In the cold-start test, SAHRE still performs well. We also studied the role of distant friends in SAHRE, and found that by considering the influences from distant friends, the coverage of SAHRE can be significantly improved with only a minor reduction in the prediction accuracy.

The performance of SAHRE can be further improved by selecting relevant friends for inference, which can be achieved by collecting the semantics of the friend relationships or fine-grained user ratings. Such an approach can be adopted by current content providers.

8. <u>REFERENCES</u>

1. G. Adomavicius, and A. 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.
2. C. Basu, H. Hirsh, and W. Cohen. Recommendation as Classification: Using Social and Content-based Information in Recommendation. *In Recommender System Workshop* '98. pp. 11-15, 1998.
3. D. Billsus and M. Pazzani. Learning Collaborative Information Filters. In *Proceedings of International Conference on Machine Learning*, 1998.
4. P. Bonhard and M. A. Sasse. "Knowing me, knowing you" - Using Profiles and Social Networking to Improve Recommender Systems. *BT Technology Journal*, Vol 24 No 3, July 2006.
5. J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 43-52, 1998.
6. V. Carvalho, and W. W. Cohen. On the Collective Classification of Email Speech Acts. *Special Interest Group on Information Retrieval*. 2005
7. B. Chandra, M. Gupta, and M. P. Gupta. Robust Approach for Estimating Probabilities in Naive-Bayes Classifier. *In Pattern Recognition and Machine Intelligence*, Volume 4815/2007, pp. 11-16, 2007
8. J. He and W. W. Chu Social Impact in Modern Recommender Systems. University of California, Los Angeles, June 2007.
9. J. He, W. W. Chu and Z. Liu. Inferring Privacy Information from Social Networks. In Proceedings of *IEEE Intelligence and Security Informatics Conference (ISI 2006)*, San Diego, California, May 2006.