Producing Timely Recommendations From Social Networks Through Targeted Search

Anıl Gürsel
Department of Computer Science
University of Tulsa
anil-gursel@utulsa.edu

Sandip Sen
Department of Computer Science
University of Tulsa
sandip@utulsa.edu

ABSTRACT

There has been a significant increase in interest and participation in social networking websites recently. For many users, social networks are indispensable tools for sharing personal information and keeping abreast with updates by their acquaintances. While there has been research on understanding the structure and effects of social networks, research on using social networks for developing targeted referral systems are few even though this can be valuable because of the abundance of information about user preferences, activities and choices. The goal of this research is to develop agent-based referral systems that learn user preferences based on past rating activities and caters to an individual user's interests by selectively searching the contributions posted by other users in close proximity in this user's social network. In particular, we are interested in fast notification of relevant activities in the social network that will enhance user awareness, satisfaction, and currency. In this paper, we propose keeping different trust values for a friend on different topics of interest and emphasize its importance with empirical results. We have developed an online photo referral system that identifies photos of possible interest to a user based on meta-data and comments on the pages of linked users on a popular photo sharing social website (*flickr.com*). We develop a probabilistic category determination mechanism that allows us to identify the possible categories an item belongs to by examining its tags. We use comments as an indirect measure of user preference for a photo. Empirical results show that our Social Network-based Item Recommendation (*SNIR*) system outperforms a content-based approach as well as the current recommendation schemes.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent Agents*

General Terms

Algorithms, Performance, Experimentation, Human Factors

Cite as: Producing Timely Recommendations From Social Networks Through Targeted Search, Anıl Gürsel, Sandip Sen, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 805–812

Keywords

social network, referral systems, recommendation, search, user preference, tags, Flickr, photos

1. INTRODUCTION

Over the last few years, interest in social networking websites such as Facebook [9], Flickr [10], MySpace [24] and Friendster [11] have increased astonishingly. As comScore Media Metrix reported, 1% of all Internet time is spent on Facebook [4]. These networks encourage users to share personal information and enable users to be up-to-date with interesting postings of their contacts. By July 27, 2008, over 24 million photos are being uploaded daily on Facebook [9]. During peak periods, photos viewed on *Flickr* reaches up to 12,000 per second [6]. As interest in these websites explodes, users are confronted with an abundance of information and options, which, in turn, can often lead to users spending more time to find relevant information. This vast volume of information also increases the likelihood of users overlooking information of interest. Recommender systems can be effective for addressing this information overload problem.

Two common recommendation approaches are *Content-based* and *Collaborative filtering* systems. In *Content-based* systems, items that are similar to the ones that the user liked in the past are recommended. This process is independent of the user and as the source of the items are not considered in the recommendation process, lacks personalization in the sense that it can recommend items from arbitrary sources. In reality, however, people may not prefer items that are provided or recommended by unknown people. On the other hand, *Collaborative-filtering (CF)* approaches try to find users with similar interests and incorporate recommendations/opinions of only those people. The recommendation selection process, therefore, is personalized and the user can be assured that the recommenders have similar interests with him/her. In general, users prefer to have more control over the selection of similar users, instead of the calculation of similarity measures by a black box system. Aimeur and Onana [1] modified the traditional Collaborative-filtering approach so that users are allowed to restrict the recommendation collection process to a set of manually selected contacts and assign a level of credit or trust to each selected contact. It was shown that the recommendations from manually selected contacts were better than the ones made by undirected collaborative filtering. Moreover, previous research [18, 20] has shown that people tend to like items that their friends like and are attracted to the activities of others in their social circle compared to people they do not know. User designated friends in social networks, therefore, can be reliable sources of recommendations.

In daily life, when people seek advice from peers, they consider their past interaction history to locate the right peer, or if an advice is received, they utilize these past interactions to judge the advice. Our aim in this research is to develop a recommendation mechanism for social networks that learns the preferences of a user by tracking indirect past ratings of a user. The goal is to recommend recently posted items on the social network that might be of interest to a user. The problem we are working on differs from common recommender systems in that we are not trying to search for and locate items to recommend. Instead, our emphasis is on ranking a large number of available candidate items from pre-existing network of friends to identify those that are likely to be of most interest to the user.

Traditional *CF* approaches are not applicable in the context of this research, i.e., for retrieving relevant information, for a user, that was *recently* uploaded in the social network. Using *CF*, a new item cannot be recommended until a number of peers have rated the item. This means that some time needs to pass after the item is available before it can be recommended. Recency of information is, however, of utmost importance in our problem domain. Moreover, *CF* approaches are good for systems where users have sufficient amount of common ratings. It is inappropriate for social networks because locality effects limit exposure of resources, and people tend to be interested in information posted by their friends and hence will mostly rate only those items.

A deficiency of many recommendation mechanisms is that they consider similarity at the user level, i.e., only one similarity value [15], or trust value [3, 13], is kept for each partner. In these systems the content of the rated items are not taken into account. In reality, however, one may have different similarity/trust values for a friend on different topic of interests. For instance, a user may have similar interests with a particular friend on sports topics but not on movies. We consider item contents while determining user similarities and hence can capture different degrees of relatedness with another user depending on the item topic or category.

To facilitate such determination, we develop a category identification algorithm that utilizes textual content, in particular, tags. In recent years, many websites have introduced tagging, including the ones that support social networking like *del.icio.us* [7] and *flickr.com* [10]. Tags allow users to describe the content of items such as photos and videos. It is usually difficult to strictly categorize digital media into a specific category. Correspondingly, we develop a probabilistic category identification mechanism for items based on associated tags. To accomplish this, we first form dictionaries for different categories and enhance them by utilizing collaborative filtering. Then, given a set of tags for an item, we calculate the probability of the item belonging to each category according to the number of tag matches for that category.

To evaluate the feasibility of our approach, we designed and implemented a photo recommender system for *flickr.com*. The *Flickr* online service allows users to share photos, attach tags to them, and comment on photos of other users. For learning user preferences and evaluating our system, we utilized the history of comments written by users. Comments indicate user interest in photos: we assume that if a comment is written on a photo p by user u, then u may either like or dislike the photo, but it is worthwhile for u to view p. In this work, we do not consider the contents of the comments to derive recommendations as we observed that a large majority of the comments on *Flickr* are positive in nature. Instead of using arbitrary heuristics, our system relies on real, even though indirect, user ratings, in the form of user comments on photos, for training and testing purposes. To the best of our knowledge, this is the first attempt at utilizing user comments for developing recommendation systems for social network applications.

The current trend in social networking websites is to recommend items based on recency. Moreover, if many items have been recently uploaded by users then a set of recommendations are created randomly. We compare the performance of our system with recommendation based on random sampling of recently uploaded items. We empirically show that our recommendation procedure improves the recall and precision values significantly. In addition, we compare the recommendation quality of our system with *content-based* [16, 28] approaches.

Our proposed approach for generating personalized, topic-specific recommendation based on recent activities by related users can be applied to different online application domains that is beset with an overload of recent information. These applications range from photo/video recommendations to smart RSS filters and news aggregators. It can also be utilized for better search results and to quickly locate needed expertise. The rest of the paper is organized as follows: in Section 2, we discuss the role of agents and trust in recommender systems. In Section 3, we formally present our recommendation problem and describe the category identification and recommendation decision mechanisms. Section 4 contains detailed experimental framework and results. We conclude, in Section 5, with a broad vision of our work and explore future directions.

2. RELATED WORK

Trust has several aspects on the web [12, 17, 25]. Much research is focused on authentication of resources, such as digital signatures and certificates. However, in this research, we are not interested in the security aspect of trust. Instead, we study the social notion of trust. In this section, we will discuss research in trust mechanisms that can improve social network based recommender systems by using trust ratings between people.

Some global trust measures are already in use on the web. For instance, *eBay* allows consumers to give positive or negative feedbacks to sellers, from which the average rating for each seller is calculated [30]. *Epinions.com* also allows users to rate transactions with users [21]. On the other hand *Google*, a popular search engine, assigns a global trust value to each webpage. The *PageRank* algorithm, used to calculate the rating of a webpage, takes into account the in-degree values and the trustworthiness of webpages that link to that webpage [5]. Since trust is a subjective concept when it comes to person-to-person relationships, global measures should not be used. To illustrate, some people believe that a politician is trustworthy, whereas other might not think so. Using a global trust value (average rating of that politician's trustworthiness for this example) will not be helpful to either group [12].

Golbeck and Hendler treat trust as a measure of uncertainty in a person or resource [14]. According to Golbeck [13], trust in a person is a commitment to an action based on a belief that the future actions of that person will lead to a preferred outcome. It can be claimed that a person A trusts another person B in terms of books, if A chooses to read a book (commits to an action) that B recommends (based on A's belief that B will not waste A's time). In other words, if A consistently finds the reviews and ratings of B valuable, then A develops trust in B [22].

Research has shown that people prefer recommendations from friends as well as from trusted systems [32, 33]. However, trust can be used in recommender systems only if it reflects user similarity. Ziegler and Lausen [36] verify that a correlation between trust and user similarity exists by providing empirical evidence obtained from a real community.

Utilizing trust in social networks can alleviate some problems of existing recommender systems. A problem with the collaborative filtering approaches is data sparsity. If the number of items is huge compared to the number of users in the system, then it might be difficult to find people who rate the same items. Moreover, a new user with few ratings cannot be matched with similar partners. Previous systems attempted to prevent this problem by limiting the number of ratable items. Another approach was to require users to rate small set of items to generate overlapping user profiles [31]. However, both of these approaches require some centralized mechanism, which is not applicable to distributed recommender systems. If a trust network can be formed, retrieving recommendations from trustful users can alleviate both the sparsity and cold start problems. In addition, collaborative filtering approaches need to use significant computation offline. When the network size increases, these operations become cumbersome. Trust networks help us decrease the amount of computation by pre-filtering users, based on their trust values, or their distance from the recommendee.

Inferring trust for unknown people, i.e., people with whom a user has no direct connection, is a key research topic in trust-aware recommender systems. For example, if A highly trusts B and B trusts C and A does not know C *a priori*, how much can A trust C? The question whether trust can be inferred might be debatable, but people do it in their real life. For example, when you ask for a repair shop from one of your friends, you take into account the trust you have for the person you are asking and their trust of the repair shop.

One way to infer trust between two people is to use the degree of connectedness [21, 22]: given a source node, assign a trust value to others based on their minimum distance from the source node. The following formula is used for predicting the trust value of a node j at distance n from the source node i:

$$T_{i,j} = \frac{d - n + 1}{d}$$

where d is the maximum trust propagation distance.

Two different nodes at the same distance are assigned the same trust value by the source node and there is a linear decay in propagating trust. The results and trust values agree with results of Golbeck [13]: shorter paths lead to more accurate information. Furthermore, Massa and Bhattacharjee [22] showed, with experiments based on *epinions.com*, that collaborative filtering approaches give better results when enhanced with trust. In another research, Massa *et al.* [2] apply trust metrics on a real world application, Moleskiing, whose goal is to make mountaineering expeditions safer. In this domain, timing is of critical importance, and hence efficiency of the system is of primary concern. Moleskiing reports the current snow conditions by gathering information only from reliable users.

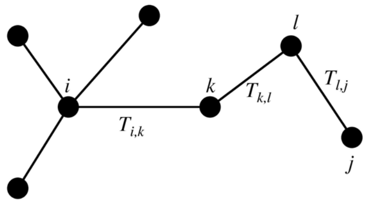

Figure 1: A trust network where each agent assigns a trust value to its neighbors.

The quality of trust values can be improved by considering weighted trust relationships. Walter *et al.* [34] calculates the trust value of an agent j according to the source agent i by multiplying all the trust values along the path from i to j:

$$T_{i,j} = \prod_{(k,l) \in path(i,j)} T_{k,l}.$$

For example, in Figure 1, the trust value of i for j is computed as $T_{i,j} = T_{i,k} * T_{k,l} * T_{l,j}$.

When an agent receives a query, it checks whether it has any related information. The agent sends the recommendation and considers the task completed if it has the information. Otherwise, the query is transmitted to its neighbors. A recommendation can be chosen if *(a)* the recommended item's preference value (according to the agent that recommended the item) is high, and *(b)* the trust along the path is high. There might exist multiple paths from the query source to the provider. Battison *et al.* uses an algorithm that does a breadth-first traversal of the graph. So, in case of multiple paths, they only consider the shortest path found. Golbeck and Hendler [14] averages the trust values in case of multiple paths. Another approach can be taking the minimum trust values.

The above recommendation algorithms compute both the similarity and the trust values between people. Hence, it requires intensive computations when the number of recommended items is high. Bedi and Kaur [3] reduces the number of computations by proposing a smarter aggregation mechanism. After each recommendation, the recommendee gives a feedback and from these experiences the recommender learns the preferences of the recommendee. The recommendee, in its turn, also updates its degree of trust for the recommender. Given past experiences, only recommendations that match the taste of user agent are used, which in turn reduces the number of computations performed by the user agent to find useful recommendations.

Golbeck [13] introduces a personalized movie recommendation system that uses trust. It is different from traditional recommendation systems in that it does not recommend items to users. Rather it provides opinions (how much the user will like it) about an item that the user has already found. To find the raters in the system, a breadth-first search algorithm is used. The system first checks whether there exists any rater that the user knows directly. If there is not a rater among the immediate neighbors, the second-level neighbors are searched and this process continues until a rater is found. A trust value is inferred by using TidalTrust for all the raters at that depth. Then the ratings of the ones with high trust values are used to calculate a weighted (by trust values) average of ratings. The results show that this system produces better performance than a simple averaging mechanism. Moreover, it outperforms a collaborative filtering approach where the similarities between people are calculated by using Pearson's Correlation coefficient.

The inferring algorithms discussed above assume that users assign an explicit trust value for their immediate neighbors. Nonetheless, if the number of contacts is high, this process might be burdensome for the users, and they might skip that step or assign random values. The problem is exacerbated if the system requires different trust values for each topic of interest for every contact. O'Donovan and Smyth [26] use past rating reliabilities to generate trust values. They consider two types of trust: profile-level and item-level trust. At the profile-level, all items that are rated by both the recommendee and recommender are compared and the percentage of correct predictions of the recommender is set as the profile-level trust value. Item-level trust is more fine-grained: only the percentage of correct predictions about a specific item, which can be recommended many times, is considered as the trust value.

Montaner *et al.* [23] develop recommender agents with a novel technique for searching similar agents. The agents in the system ask other agents their opinions about the interest of their users on a new, or newly discovered, item, in case of lack of information about the item. This is similar to the opinion based approach of Golbeck [13]. The opinion is calculated according to the opinions of others weighted by their corresponding trust values. Agents learn the initial trust values of other agents by contacting others and using initial exploration. Palau *et al.* [27] provide experimental results from a real restaurant recommender system that uses a similar opinion based approach that uses trust values.

3. TECHNICAL FRAMEWORK

We propose to use the friends on social networking websites, like *Flickr*, as designated peers and use their activities as the basis of recommendations for recently uploaded items. Thus, our recommendation generation approach is different from *collaborative-filtering* where the focus is on identifying peers with similar interests, and therefore, forming an implicit social network based on user browsing and rating activities. Users on social networking websites tend to have a large number of friends and are increasingly sharing more information about themselves. As a result, the number of items available for recommendations from designated friends can still be numerous. This research develops techniques that enable fast selection of items to be recommended to a user from the set of recently shared items by designated friends of this user. In this section, we formally present the Social Network-based Item Recommendation (*SNIR*) system and describe the category identification and recommendation decision mechanisms we have developed for it.

Each recommendation system consists of two main entities: a set of users $U = \{u_1, ..., u_m\}$ and a set of items $I = \{i_1, ..., i_n\}$. In social networking systems, each item can have some content descriptors. In our system, we consider tags $T = \{t_1, ..., t_o\}$ as content descriptors. Comments on items are indirect measures of preferences. We assume that if a user u writes a comment, $comment(i, u)$, for an item i, then, regardless of u liked or disliked the item, u is of interest to i, i.e., it is worthwhile for u to view i.

We use a category-based mechanism that identifies the set of categories to which an item belongs. The system considers a set of categories; $C = \{c_1, ..., c_r\}$. As it is usually difficult to strictly categorize digital media into one category, we present a probabilistic category identification mechanism for items based on its associated tags. Accordingly, we form a dictionary of tags for each category, that contains a set of possible tags for the corresponding category. So the system comprises a set of dictionaries $D = \{d_1, ..., d_r\}$, where each dictionary contains a set of tags: $d_x = \{t_{x_1}, ..., t_{x_v}\}$.

Each item has an associated ordered list of tags. We believe that not all tags attached to an item are of equal importance. The higher the rank of a tag in the list associated with an item, the more effect on the category identification it is expected to have. Hence, we weigh each tag according to its position in the list. Let a tag t^i_j be in the j^{th} position of the list i. Then the weight of t^i_j on the category determination algorithm is calculated with a decreasing function of j as follows:

$$w(t^i_j) = \frac{1 - \alpha * j^\gamma}{\sum_{k=1}^{l_i} 1 - \alpha * k^\gamma}$$

where l_i is the number of tags in the list for item i. α and γ are constants in the range of $(0, 1)$ that can be tuned for each domain.

Given a set of tags for an item i, $tags(i)$, we calculate the probability of an item i belonging to each category according to the number of tags in $tags(i)$ that are also included in the dictionary for that category. Let $lookup(d, t)$ be a predicate that checks whether tag t is included in dictionary d or not:

$$lookup(d, t) = \begin{cases} 0 & \text{if } t \notin d, \\ 1 & \text{if } t \in d. \end{cases}$$

The probability of i belonging to a category c_x is, then, calculated as follows:

$$Pr(cat_i = c_x) = \frac{\sum_{k=1}^{l_i} w(t^i_k) * lookup(d_{c_x}, t^i_k)}{\sum_{c_y \in C} \sum_{k=1}^{l_i} w(t^i_k) * lookup(d_{c_y}, t^i_k)}$$

where cat_i stands for *category of i*.

To choose items to recommend to a user, a pool of candidate items is chosen based on the recently uploaded items by this user's friends. We next calculate a preference measure for each candidate item and recommend the ones that have the highest preference values for the target user. The notation $Pr(\ likes(u_a, i) \mid i \in posted(u_b))$ corresponds to the probability of an item i that is posted by u_b being liked by the target user u_a. This probability calculation plays the key role in our recommendation process. By using Bayes Theorem [29], we calculate this probability, $Pr(\ likes(u_a, i) \mid i \in posted(u_b)\ )$, as follows:

$$\frac{Pr(\ i \in posted(u_b) \mid likes(u_a, i)\ )\ Pr(\ likes(u_a, i)\ )}{Pr(\ i \in posted(u_b)\ )}. \quad (1)$$

Now we derive the probabilities in the expression above except $Pr(\ likes(u_a, i)\ )$ which will be eliminated in the following steps. In our formulation, preference measures for an item depends both on the owner of the item and the content of the item. $Pr(i \in posted(u_b) \mid likes(u_a, i))$, therefore, can be expanded as

$$\sum_{c_x \in C} [Pr(i \in posted(u_b) \mid cat_i = c_x, likes(u_a, i)) \quad Pr(cat_i = c_x \mid likes(u_a, i))]. \tag{2}$$

After applying Bayes rule to the second conditional probability, the right hand side of Equation 2 becomes

$$\sum_{c_x \in C} [Pr(i \in posted(u_b) \mid cat_i = c_x, likes(u_a, i)) \quad \frac{Pr(likes(u_a, i) \mid cat_i = c_x) \; Pr(cat_i = c_x)}{Pr(likes(u_a, i))}]. \tag{3}$$

The probability of an item i posted by a user also depends on the content, since a user's posting habits might be biased towards some categories. So, the denominator of Expression 1, $Pr(i \in posted(u_b))$, can be written as follows:

$$\sum_{c_x \in C} [Pr(i \in posted(u_b) \mid cat_i = c_x) \; Pr(cat_i = c_x)]. \tag{4}$$

After substituting 2, 3, and 4, and simplifying, $Pr(i \in u_b \mid likes(u_a, i))$ can be written as:

$$\frac{\sum_{c_x \in C} [Pr(i \in posted(u_b) \mid cat_i = c_x, likes(u_a, i)) \quad Pr(likes(u_a, i) \mid cat_i = c_x) \; Pr(cat_i = c_x)]}{\sum_{c_x \in C} Pr(i \in posted(u_b) \mid cat_i = c_x) \; Pr(cat_i = c_x)}. \tag{5}$$

The unknown probabilities in 5 are computed as follows:

- $Pr(i \in posted(u_b) \mid cat_i = c_x, likes(u_a, i))$ = (Number of photos posted by u_b that belong to c_x and are commented on by u_a) / (Number of all photos that are in c_x and are commented on by u_a).
- $Pr(likes(u_a, i) \mid cat_i = c_x)$ = (Number of all photos that are in c_x and are commented on by u_a) / (Number of all photos that are in c_x).
- $Pr(i \in posted(u_b) \mid cat_i = c_x)$ = (Number of photos posted by u_b that are in c_x) / (Number of all photos that are in c_x).

4. EXPERIMENTAL STUDY

We have evaluated the quality of our *SNIR* system on a real domain *Flickr* [10], the popular photo sharing website. In *Flickr*, users can upload photos, tag photos with descriptive words, and write comments on them to express their opinion. Moreover, it also allows users to designate others as friends, which enables them to easily track activities of friends in the social network.

For our experiments, we collected data for 15 root users. These users are selected randomly from the posters of photos that are listed on *Flickr*'s *interestingness* page. Then, we selected 10 out of those users that have posted a relatively higher number of comments. For each root user, we visit his/her friends' accounts and gather information about their photos, e.g., tags and comments, that are uploaded or posted between *January 1, 2008* and *April 30, 2008*. Our corpus, therefore, includes photo information of 4025 users, who have together posted 121953 photos. The total number of comments that are written by root users is 30040. The average number of unique tags for a root user's friends' photos is 20017.6. In our experiments, the data from the first 3 months, from *January 1, 2008* to *March 31, 2008*, is used for training the prediction system (learning the probabilities in Equation 5), and the last month's data, from *April 1, 2008* to *April 30, 2008*, is used for testing the quality of recommendations generated by SNIR.

After researching social networking websites and some online services, e.g., *digg* [8], *youtube* [35], we decided to use 10 different categories for photos. Because of categorization uncertainty a photo in our system can belong to one or more of the following categories: *Animal, Art, City, Entertainment, Nature, News and Politics, People, Science and Technology, Sports, Travel and Places.* A more fine-grained set of categories can also be used since our system does not depend on a pre-determined, fixed, set categories. For each category, we built a dictionary that contains a set of category-related tags.

4.1 Analysis of User Behavior

Typical users of social networking applications are mostly interested in their friends' activities [19, 20]. However, users tend to have a large number of friends (peers) in their contact lists and their interest in different peers is not the same. Moreover, a user might ascribe different preference values for the same peer based on the topic or category of photos. These preference values are not correlated with the number of photos posted by a peer on a topic.

We now state and verify the key hypothesis about user behavior in *Flickr* that motivates our approach.

HYPOTHESIS 4.1. *A user has different preferences for different contacts and these preferences are not correlated with the number of photos posted by a contact.*

In Figure 2, the number of photos by friends and the total number of comments written on photos of that friend by a *Flickr* user is plotted. These plots are representative of the behavior of other randomly selected users in *Flickr* that we have analyzed. On the *x-axis* the friends are placed in descending order based on the number of comments posted by the *Flickr* user. As is seen from the figure, the number of comments written on a friend's photos is not correlated with the number of photos. It is clear that more active friends, i.e., friends posting more photos, does not necessarily receive more attention.

HYPOTHESIS 4.2. *A user has different preferences for the same peer based on item topics (photos in the case of* Flickr*).*

In Figure 3 percentage of photos of friends that are commented on by the *Flickr* users are plotted for two different topics. On the *x-axis* the friends are placed in descending order based on percentage of commented photos belonging to Topic 1 by the root user. The figure tells us that interest in a peer's posts is different for different topics. Just because a user u prefers a friend f_1 over a friend f_2 for topic t_1, it does not mean the user will also prefer f_1 over f_2 for another topic t_2. Let $P^t_{u,f}$ be the preference level of user u for friend f on topic t. Hence, $P^{t_1}_{u,f_1} > P^{t_1}_{u,f_2}$ does not imply $P^{t_2}_{u,f_1} > P^{t_2}_{u,f_2}$ where $t_2 \neq t1$.

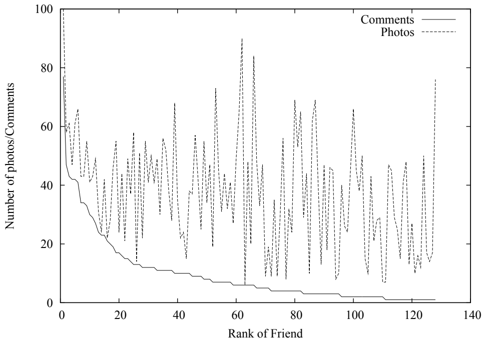

Figure 2: The number of photos posted by a friend and the number of comments written on photos of that friend.

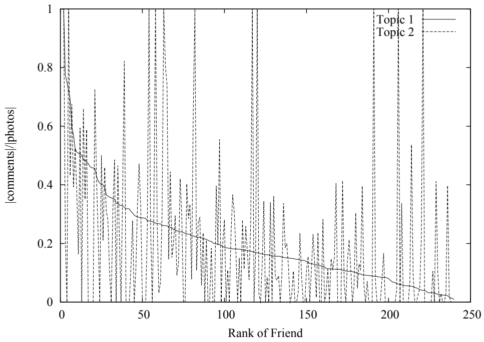

Figure 3: Number of comment per photo by a user on photos posted by his/her friends on two different topics.

4.2 Evaluation

As contact lists of users grow and users become more active in social networks, the number of possible recommendations for a particular user increases. The current trend in social networking websites is to recommend most recently posted items or items that are randomly sampled from recently posted photos. We cannot retrieve the login times of users from *Flickr*, and hence cannot use the recency factor. We compare the performance of *SNIR* with recommendations randomly sampled from the photos posted the previous day. Instead of using a randomizing function to select photos to recommend by the *Flickr* recommendation mechanism and observe the corresponding precision of prediction (fraction of recommended photos commented on by the user), we calculate *expected precision* from random sampling.

Overall Success Rates: We use *precision* and *recall* as our performance measures. Figure 4 compares *random sampling* and our *SNIR* system in terms of *precision* and *recall* values, respectively. The figures show the performance of the two approaches during a one-month testing period in *April, 2008*. The results are averaged over 10 target root users. In Figure 4 the precision values are calculated for 20 recommendations per day. The expected precision of random recommendation is in the *0.2-0.3* range and *SNIR* can approximately double this on some days. We note that half of the users in the system write a maximum of 10 comments per day (one of them has a maximum of 5). Thus, their precision values cannot cross 0.5 as the number of recommendations per day is *20*. Note that we measure the success of the system based only on comments. The actual success rate, which can include photos viewed and liked but not commented on, can only be higher. Also, there might have been a set S of photos that were overlooked because of the information overload problem, and that may be the reason why the user did not comment on it. If our system did recommend some of those photos in s, their probable success will not be reflected in the results. We, therefore, claim that the results shown on the graphs are the lower bound of the precision of *SNIR*.

Recall values in Figure 4 also show performance improvement using *SNIR*. Users in our system (mostly the ones that have less number of comments) have less comments in the first half of the month as well as more dynamic behavior. The deviation is high therefore in the first 15 days. If the number of recommendations per day is increased, the graph will stabilize and will exhibit higher recall value, but at the cost of a decrease in precision values.

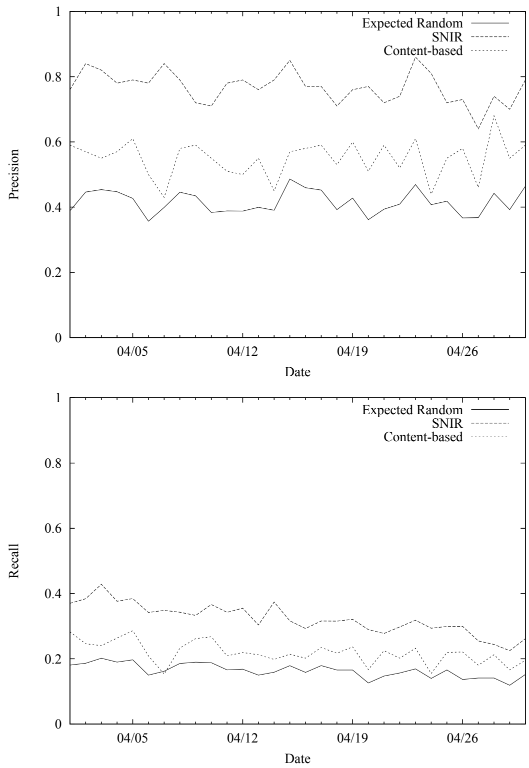

Figure 4: Comparison of precision (upper figure) and recall (bottom figure) values of random sampling method, content-based recommendation and SNIR.

Precision Versus Item Rank: The *SNIR* recommendation process generates an ordered list of items to be recommended. It is expected that items higher in this ranked list are more likely to be of interest to the user. To verify this expectation, which is a more direct measure of the effectiveness of the recommendation process, we calculate the average precision value for each rank in the ordered list, i.e.,

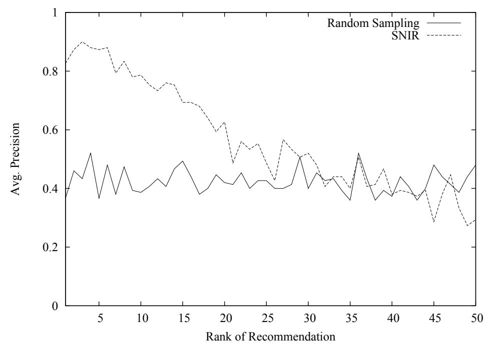

Figure 5: Precision values for different recommended items based on their rank.

for each rank we calculate the fraction of times the corresponding recommended item was liked (commented on) by the target user. In Figure 5, we demonstrate the precision of *SNIR* and random selection for items ranked *1* to *50*. For each ranking position, we tested the system for 30 days during *April 2008* for *5* different users. The figure convincingly demonstrates that items highly ranked by *SNIR* had higher precision. In sharp contrast, the random selection mechanism has similar precision for all items.

Comparison with a Content-Based Recommender: In this set of experiments, we compare the performance of our system with a *content-based* recommender system where items that are similar to the ones previously liked by the user are recommended. So the preferences of the user for each feature (tags in our case) should be determined by observing the past items preferred by the user. We implemented a content-based recommender that determines user's preferences without consideration of preferences for friends. Actually we can view this *content-based* system as a simpler version of *SNIR* where the preference for friends is ignored. Our *content-based* system uses the following equation to calculate the preference level of the user for a photo, $Pr(\ likes(u_a, i)\ )$

$$\sum_{k=1}^{l_i} w(k) \frac{|\text{ commented photos that have } t_{ki} \text{ in their tag list }|}{|\text{ photos that have } t_{ki} \text{ in their tag list }|}. \tag{6}$$

Figure 4 compares the precision and recall values of random sampling method, the content-based system and our system for a test period of one month where the results are averaged over 5 users. Even though the *content-based* system outperforms the random sampling method, its performance falls well short of *SNIR*'s. This figure proves that for a given topic of interest a user has different preferences for different friends and the utilization of these preferences improves the quality of recommendation dramatically.

5. CONCLUSIONS

Social networking websites such as *Facebook* and *Flickr* are attracting more interest everyday and users' participation in these websites is increasing. These websites allow users to designate others as friends or contacts and track their activities as users become more active. As the size of friend lists increase, however, it becomes impossible for users to keep track of all of their contacts and their activities through simple interfaces currently offered [20].

We develop a Social Network-based Item Recommendation (*SNIR*) system to effectively address this information overload by recommending items of interest to the users from items recently posted by their contacts. To evaluate the effectiveness of our *SNIR* approach, we developed a photo recommendation system for the popular photo sharing website *Flickr*. Our system relies on metadata information of items and activities of users that corresponds to tags and comments respectively in *Flickr*. We believe that comments reflect the preferences of users for items. We also believe that an individual's preferences for an item might differ according to both the category of the item as well as the source of that item, e.g., the user who posted that photo in *Flickr*. Therefore, a system that learns user preferences for other friends, based on comments posted by the user for every other friend, is developed. To facilitate the recommendation process, a probabilistic category determination scheme based on item features (tags in *Flickr*) is designed.

We evaluated our system with experiments on data gathered from *Flickr* for 4025 accounts containing 121953 photos. To redress the information overload problem, we increase precision by ordering results based on predicted user preference value for the items and recommending only few highly rated items. The results of experiments demonstrate that our system's recommendation performance outperforms the current methods used both in terms of precision and recall values. We also compare our system with a content-based system in which preferences for each contact are not considered. We find that the *SNIR* system produces considerably better precision and recall values compared to the content-based recommendation system.

One immediate extension to *SNIR* can be to enhance it to provide recommendations from contacts further afar in the social network. By doing so, people might discover friends' friends that have very similar interests for some type of items. However, to do so, information need to be gathered about accounts of many more users. Gathering information about many more users might also give us more reliable results for the existing experiments.

Acknowledgment: A DOD-ARO Grant #W911NF-05-1-0285 partially supported this work.

6. REFERENCES

[1] E. Aimeur and F. S. M. Onana. Better control on recommender systems. In *CEC-EEE '06: Proceedings of the The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, page 38, Washington, DC, USA, 2006. IEEE Computer Society.

[2] P. Avesani, P. Massa, and R. Tiella. A trust-enhanced recommender system application: Moleskiing. In H. Haddad, L. M. Liebrock, A. Omicini, and R. L. Wainwright, editors, *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC), Santa Fe, New Mexico, USA, March 13-17, 2005*, pages 1589–1593. ACM, 2005.

[3] P. Bedi and H. Kaur. Trust based personalized recommender system. *INFOCOMP Journal of Computer Science*, 5(1):19–26, 2006.

[4] T. F. Blog. URL http://blog.facebook.com/blog.php?post=2245132130.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[6] H. Champ. Flickr blog: We're going down., 2007. http://blog.flickr.net/2007/05/29/were-going-down/.

[7] Delicious. URL http://delicious.com.

[8] Digg. URL http://www.digg.com.

[9] Facebook. URL http://www.facebook.com/press/info.php?statistics.

[10] Flickr. URL http://www.flickr.com.

[11] Friendster. URL http://www.friendster.com.

[12] J. Golbeck. Combining provenance with trust in social networks for semantic web content filtering. In *International Provenance and Annotation Workshop (IPAW'06*, Chicago, Illinois, USA, May 2006.

[13] J. Golbeck. Generating predictive movie recommendations from trust in social networks. *Trust Management*, pages 93–104, 2006.

[14] J. Golbeck and J. A. Hendler. Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *EKAW*, pages 116–131, 2004.

[15] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250, New York, NY, USA, 2000. ACM.

[16] C. Jian, Y. Jian, and H. Jin. Automatic content-based recommendation in e-commerce. In *EEE '05: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service*, pages 748–753, Washington, DC, USA, 2005. IEEE Computer Society.

[17] G. Lenzini, M. S. Bargh, and B. Hulsebosch. Trust-enhanced security in location-based adaptive authentication. *Electron. Notes Theor. Comput. Sci.*, 197(2):105–119, 2008.

[18] K. Lerman. Social networks and social information filtering on digg. *ArXiv Computer Science e-prints*, December 2006.

[19] K. Lerman. User participation in social media: Digg study. In *Proceedings of the International Workshop on Social Media Analysis (SMA07), Conference on Web Intelligence and Intelligent Agent Technology*, 2007.

[20] K. Lerman and L. Jones. Social browsing on flickr. *ArXiv Computer Science e-prints*, 2006.

[21] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. In R. Meersman and Z. Tari, editors, *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, OTM Confederated International Conferences, Agia Napa, Cyprus, October 25-29, 2004, Proceedings, Part I*, volume 3290 of *Lecture Notes in Computer Science*, pages 492–508. Springer, 2004.

[22] P. Massa and B. Bhattacharjee. Using trust in recommender systems: An experimental analysis. In C. D. Jensen, S. Poslad, and T. Dimitrakos, editors, *Trust Management, Second International Conference, iTrust 2004, Oxford, UK, March 29 - April 1, 2004, Proceedings*, volume 2995 of *Lecture Notes in Computer Science*, pages 221–235. Springer, 2004.

[23] M. Montaner, B. López, and J. L. de la Rosa. Developing trust in recommender agents. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 304–305, New York, NY, USA, 2002. ACM.

[24] MySpace. URL http://www.myspace.com.

[25] M. Nilsson, A. Adams, and S. Herd. Building security and trust in online banking. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1701–1704, New York, NY, USA, 2005. ACM.

[26] J. O'Donovan and B. Smyth. Trust in recommender systems. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174, New York, NY, USA, 2005. ACM.

[27] J. Palau, M. Montaner, B. López, and J. L. de la Rosa. Collaboration analysis in recommender systems using social networks. In *CIA*, pages 137–151, 2004.

[28] Z. Pan, B. Xu, H. Yang, and M. Zhang. Content-'based personalised recommendation in virtual shopping environment. *Int. J. Bus. Intell. Data Min.*, 1(4):430–449, 2006.

[29] S. J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

[30] J. B. Schafer, J. Konstan, and J. Riedi. Recommender systems in e-commerce. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, New York, NY, USA, 1999. ACM.

[31] U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[32] R. R. Sinha and K. Swearingen. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

[33] K. Swearingen and R. Sinha. Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR. Workshop on Recommender Systems*, volume Vol. 13, Numbers 5-6, pages 393–408, 2001.

[34] F. E. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, 2008.

[35] Youtube. URL http://www.youtube.com.

[36] C.-N. Ziegler and G. Lausen. Analyzing correlation between trust and user similarity in online communities. In C. D. Jensen, S. Poslad, and T. Dimitrakos, editors, *Trust Management, Second International Conference, iTrust 2004, Oxford, UK, March 29 - April 1, 2004, Proceedings*, volume 2995 of *Lecture Notes in Computer Science*, pages 251–265. Springer, 2004.