SocialNews: Enhancing Online News Recommendations By Leveraging Social Network Information

Mike Gartrell, Richard Han, Qin Lv, Shivakant Mishra

University of Colorado at Boulder

Contact: mike.gartrell@colorado.edu

Department of Computer Science

University of Colorado at Boulder

Technical Report CU-CS-1084-11

August 2011

SocialNews: Enhancing Online News Recommendations By Leveraging Social Network Information

Mike Gartrell, Richard Han, Qin Lv, Shivakant Mishra
University of Colorado at Boulder
mike.gartrell@colorado.edu

ABSTRACT

SocialNews seeks to enhance the news results returned by online news services such as Google News or Bing News by leveraging social networking information to provide more relevant results to the user. For example, Facebook profiles contain a wealth of information about the user's interests, hobbies, personal history, and friends. We describe a system that analyzes the content in a user's social networking profile, and provides more targeted news recommendations based on Facebook information such as hometown, strength of social ties, user similarity, etc. Preliminary assessment of the system is presented.

1. INTRODUCTION

Recommender systems, which automatically identify and recommend objects of interest to specific users, have become increasingly important as we move quickly into the digital era. A number of recommendation techniques have been developed, targeting different types of objects [5]. Among these applications, news article recommendation, due to high demand as well as the diversity and dynamics of new articles, is of particular importance and poses unique challenges on recommender systems. Previous approaches rely on content similarity (e.g., via topic analysis) or collaborative filtering [8]. Although these techniques perform reasonably well, they do not consider the social interests or social relationships of users, which also play an important role in news article recommendation. For example, a user may be interested in a particular news article because its liked by her close friends or because the event occurred in a place where her family lives.

Recently, online social networks or social media websites (e.g., Facebook, Twitter, LinkedIn) have grown rapidly, attracting users from all over the world. A wealth of social related information has become available online, such as user profiles, posts, social relationships and interactions, etc. A number of techniques have been proposed for social network analysis, such as estimating social tie strength [10, 14] and mining the power of likes [12]). Researchers have also started investigating social-based recommender systems [11, 7, 6, 13], and different techniques have been proposed, such as social map, matrix factorization with trust propagation, social regularization, friendship-interest propagation, etc.

In this work, our goal is to develop a social-based news article recommender system, called SocialNews. Building upon previous research on general recommender system design and recent social-based recommendation techniques, we propose an integrated system that fuses together social-based and content-based information, thus allowing for the identification and recommendation of news articles that are of interest to particular users due to similarity or closeness in either content or social connections. This problem is challenging and differs from existing work as we need to determine what social and content information is important for news article recommendation, how to extract this information, and how to fuse it together, as different factors may play different roles depending on the specific user, his/her social environments, and the specific news article.

We claim that SocialNews makes the following contributions:

- To the best of our knowledge, SocialNews is the first news recommendation system that fuses together multiple dimensions of social indicators, including social strength and user similarity, to generate news recommendations (Sections 3.1 and 3.5).

- SocialNews incorporates a novel approach that dynamically adjusts the weight given to social strength and user similarity based on user clickthrough feedback (Section 3.6).

- We believe our method of computing user similarity by applying Explicit Semantic Analysis (ESA) to users' Facebook "Likes" is unique (Section 3.4).

- The approach of generating news recommendations from Facebook Likes is seen as novel (Section 3.2).

The remainder of this paper is organized as follows. Section 2 surveys research works that are most related to our work. Section 3 gives an overview of the system

framework, and discusses in detail the individual design components, including recommendation based on Facebook Likes, social strength, user similarity, ensemble, and system feedback via clickthrough processing. Section 4 presents our preliminary evaluation results. Finally, Section 5 concludes and discusses future directions.

2. RELATED WORK

Our system of social-based news article recommendation builds upon research in recommender systems in general, social network analysis, and social-based recommendation in particular. In this section, we survey research most related to ours.

Recommender system has been an area of active research since the mid-1990s (see [5] for a survey). Various recommendation techniques have been proposed, including content-based similarity analysis, user or item based collaborative filtering, and hybrid recommendation techniques.

More recently, with the increased popularity of online social networks and social media websites (e.g., Facebook, Twitter, LinkedIn), social network analysis and social-based recommendation have attracted much attention in the research community. One line of research focuses on using the interaction information between users to prune the spurious relationships and highlight the stronger relationships. Gilbert and Karahalios proposed a predictive model that maps social media data to tie strength [10]. They have considered seven factors in distinguishing strong and weak ties: intensity, intimacy, duration, reciprocal services, structural, emotional support, and social distance. Xiang et al. proposed a link-based latent variable model and a coordinate ascent optimization procedure to infer relationship strengths based on user profile similarity and interaction activity, thus automatically distinguishing strong relationships from weak ones [14]. Jin et al. has recently developed a prototype system called LikeMiner, which mines the power of 'like' in social media networks using a heterogeneous network model for social media with 'likes' [12].

A number of social-based recommendation techniques have been proposed that utilize social-related information to improve the prediction accuracy of traditional recommender systems. Zhao et al. have developed a social map based recommender system, called Pharos [16]. A social map summarizes users' content-related social behavior over time (e.g., reading, writing, and commenting) as a set of latent communities. Such a visual social map allows (new) users to quickly identify interested content or people. Chen et al. investigated the use of thread length, topic, and tie-strength for recommending Twitter conversations [6]. They considered users' purposes of using Twitter and found that tie-strength

based algorithms performed significantly better for people who use Twitter for social purposes than for people who use Twitter for informational purpose only. In an earlier work, the factors of content sources, topic interest models of users, and social voting have also been explored for recommending Twitter content [7]. A matrix factorization technique with trust propagation along social links has been proposed by Jamali and Ester for recommendation in social networks [11]. Their experimental results demonstrate increased performance with trust propagation, in particular for cold start users. Ma et al. studied the differences between social-based recommender systems and traditional recommender systems, and proposed a general matrix factorization framework with *social regularization*, which represents social constraints on recommender systems [13]. A friendship-interest propagation (FIP) framework was recently proposed by Yang et al. [15]. This framework integrates both friendship connections, via a factor-based random walk model, and interest interactions, via a coupled latent factor model. As a result, this framework performs well on both interest targeting and friendship prediction.

While our system uses similar information for measuring the strength of social ties, it fuses a number of aspects of social and content based information (e.g., Facebook user profiles) for the purpose of recommending news articles that are of high interest to users, and dynamically shifts the weights associated with each aspect based on user feedback.

3. RECOMMENDATION FRAMEWORK

SocialNews uses an ensemble of recommender systems based on social relationship strength and user similarity. In the cold-start scenario, where a new user has begun using SocialNews, we assume that this user will be interested in a mixture of news articles related to three streams of information: the preferences of close friends, friends that are most similar to the new user, and the user's own interests. As the user clicks on recommended articles, SocialNews learns about the user's preferences and dynamically adjusts the mixture of recommended articles from each of these three streams.

3.1 System Architecture

SocialNews is composed of several components that are used in the process of computing recommendations, as shown in Figure 1. First, we rank the user's Facebook friends using the social strength and user similarity components. Then, for each of a user's top friends, we rank the friend's Facebook Likes and compute recommended news articles from these top-ranked Likes. We also rank the user's own Likes and compute recommendations from these Likes using this mechanism. Facebook Likes can include the names of a user's in-
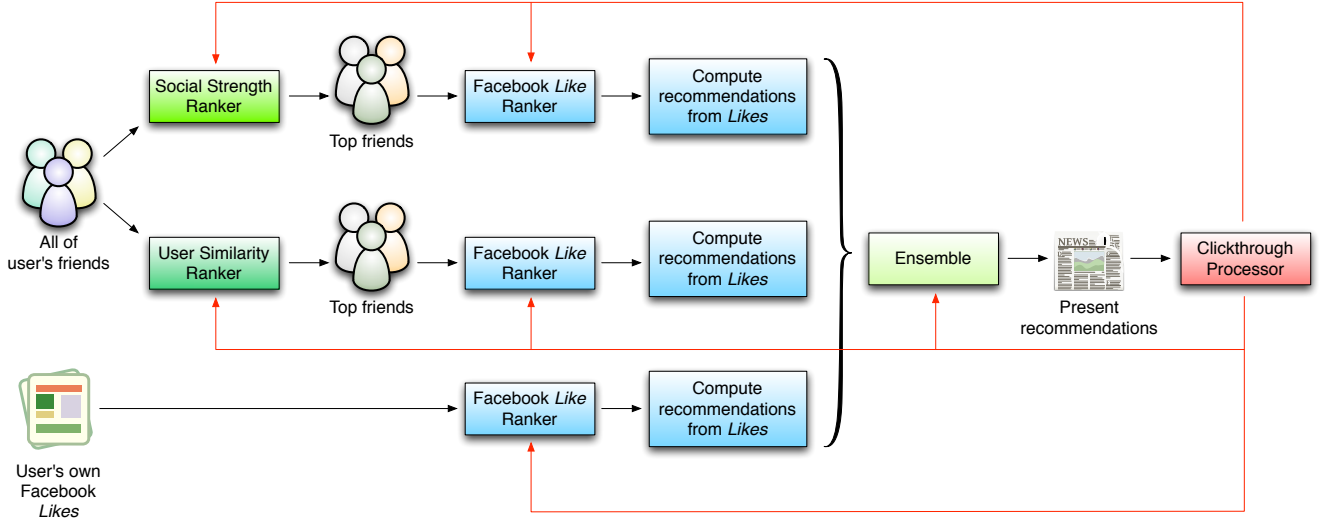
Figure 1: SocialNews system architecture

terests and favorite movies, books, music, Web pages, etc.

SocialNews recommendations are generated from three streams: social strength ranking of friends, user similarity ranking of friends, and a ranking of the user's own Facebook Likes. The ensemble component is responsible for merging the recommendations from each stream into a final list of recommendations presented to the user. Finally, the SocialNews clickthrough processing component records the user's clicks on recommended news articles and provides feedback used to tune parameters in the components used to rank Likes and friends. Clickthrough data is also used to alter parameters in the ensemble component. The following sections describe the operation of each of these components.

3.2 Computing News Recommendations from Facebook Likes

All news recommendations generated by SocialNews are computed from the Facebook Likes of a user or his friends. To determine which Likes to use for news recommendations, we rank the Likes for all users and users' friends in SocialNews by building a Like profile for each user and friend. Our approach is conceptually similar to the Self-Profile approach described in [7], although we use Facebook Likes instead of words that users have included in their Twitter messages (tweets). For each user/friend u we create a profile that consists of a vector $V_u = (v_u(l_1), \ldots, v_u(l_n))$, where n is the number of Likes for u, and each $v_u(l_i)$ indicates the strength of u's interest in Like l_i. The value of $v_u(l_i)$ is computed using a term-frequency inverse-user-frequency scheme (TF-IDF) defined as

$$TF_u(l_i) = clickCount(u, l_i) \qquad (1)$$

$$IDF_u(l_i) = log(userCount(all)/userCount(l_i)) \quad (2)$$

$$v_u(l_i) = TF_u(l_i) \cdot IDF_u(l_i) \qquad (3)$$

where $clickCount(u, l_i)$ is the number of clicks from u on SocialNews recommendations generated from l_i, $userCount(all)$ is the number of users/friends in Social-News, and $userCount(l_i)$ is the number of users/friends with Like l_i.

A high TF value for a Like indicates that a user reads recommendations generated from that Like frequently. A high IDF score for a Like indicates that it is rare across the SocialNews population of users and friends. We consider rare Likes to be more representative of a user's unique interests than common Likes. After computing TF-IDF scores for all user and friend Likes, we sort by TF-IDF and select the top k Likes for each user/friend. To compute recommendations from a lLke, the name of each top Like is passed to the Bing News search engine as a query, which returns the most relevant news articles for this Like. We then select the top m news articles for each Like. Thus, we have a list of up to $k \cdot m$ articles associated with each user/friend. After the top friends for a user are computed by the Social-News social strength and user similarity ranking components, we present the news articles associated with each top friend as news recommendations to the target user.

3.3 Social Strength Ranking

The social-strength-based ranking component in SocialNews sorts a user's Facebook friends according to social strength. We model the social strength between a (u, f) pair using a number of Facebook interaction and profile features, where u is a user and f is a friend of that user. A score is assigned to certain values or
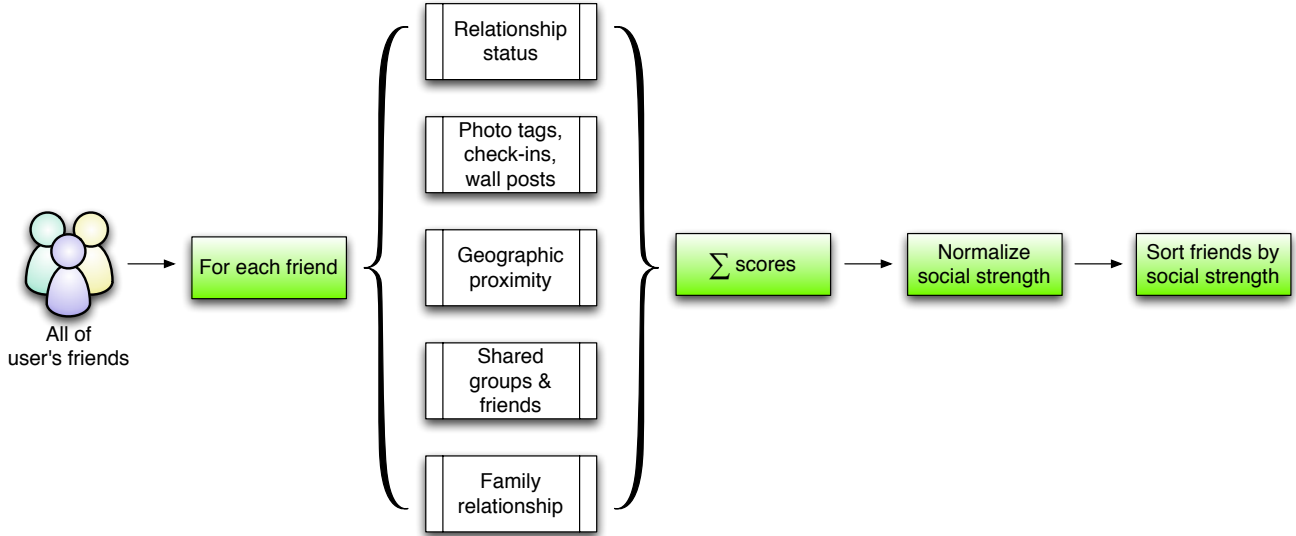
Figure 2: Social strength ranking component

thresholds for each feature, which reflects the relative importance of each feature in terms of social strength. We use the following features:

1. *Relationship status.* We consider this feature to be one of the most important indicators of social relationship strength. If the status is married, then we assign a score of 12 to this feature. If the status is set to engaged or "in a relationship with", we assign a score of 10 and 8, respectively.

2. *Photos tags, check-ins, and wall posts.* These are measures of the interaction between a (u, f) pair. For the photo tags feature, we count the number of instances where the (u, f) pair are tagged in the same photo. Regarding check-ins, we count the number of check-ins where (u, f) are found in the same check-in tag, which indicates that they checked in at the same location together. Finally, for the wall posts feature, we count the number of times that the friend has posted on the user's wall. The maximum scores for the photo tags and check-in features are 6, while the maximum score for the wall posts feature is 2. The maximum scores for these features reflect their relative importance; we observe that photo tags and check-ins are measures of real-world interaction, and thus better indicators of relationship strength than wall posts. We use the Gompertz function

$$y(c_{feat}) = ae^{-5e^{-c_{feat}}} \qquad (4)$$

to compute the scores for these features, where c_{feat} is the count for the specified feature, and a (the upper asymptote of the function) is set to be

the maximum score for the feature. We selected this function to avoid favoring the size of c_{feat} too much and to keep the feature score in the specified range.

3. *Geographic proximity.* The geographic distance between two individuals is a vital factor in social relationship strength. We consider the user's current city of residence (the "lives in" field on Facebook) when computing geographic distance. For a (u, f) pair, if both individuals live in the same city, we assign the maximum score for this feature: 5. If the geographic distance is less than 60 miles, we assign a score of 3. Finally, if the geographic distance greater than 60 and less than 100 miles, we assign a score 2. We perform similar computations for the "hometown" field on Facebook. If the hometowns of (u, f) are the same, when we assign a score of 3 for the hometown geographic proximity feature. If the hometowns are less than 60 miles apart, we assign a score of 2. For hometowns greater than 60 and less than 100 miles apart, we assign a score of 1 for this feature.

4. *Count of shared friends and shared groups.* The maximum score for the count of shared friends feature is 4, while the maximum score for the count of shared groups is 3. To compute the scores for each of these features, we normalize across the count of common friends/groups for each of (u, f) pair for a specified user u.

5. *Family relationship.* We look for sibling (score of 4), parent (score of 3), and other (score of 2) family relationships as indicators of social strength.

4

We obtain data for each of the features described above using the Facebook Graph API. As shown in Figure 2, after summing the scores for each feature, we normalize the net social strength to the range [0, 1.0] and then rank each of a user's friends according to social strength.

3.4 User Similarity Ranking

The SocialNews user-similarity-based ranking component sorts a user's friends based on semantic similarity between Facebook user profiles. We use Explicit Semantic Analysis (ESA) to compute the semantic similarity between profiles. Briefly, ESA computes the semantic similarity between two blocks of text by projecting each text block into a high-dimensional vector space composed of Wikipedia articles and computing the cosine similarity between the vectors for each text block; see [9] for details.

To compute the semantic similarity between user profiles, we construct a text block composed of the names of each Facebook Like for the user. After constructing the Likes text block for a user u and each of his friends f, we compute the semantic similarity for each (u, f) pair by passing each corresponding pair of text blocks to ESA. ESA returns a similarity score in the range [0, 1.0]. Finally, we rank each of a user's friends according to the computed similarity scores.

3.5 Ensemble

As shown in Algorithm 1, SocialNews uses a combination of the recommendations generated by the social strength and user similarity components to generate the final list of recommendations provided to the user. Additionally, we incorporate recommendations generated from top-ranked Likes in the user's own Facebook profile. For a new user, the top recommendations from each of these three sources is interleaved into the recommendation list, and the percentile weight for each source is the same - up to 33%. As a user clicks on recommended news articles, we use this feedback to dynamically adjust the weight of each source to reflect the user's preferences.

3.6 Clickthrough Processing

We track the clicks recorded when a SocialNews user clicks on recommended articles and use this data to compute future recommendations. As explained in the previous subsections, clickthrough data alters the ranking of Facebook Likes for a user or friend, and also alters the ranking of friends in the social strength and user similarity components. For ranking Likes for a user/friend, we consider the number of clicks for articles associated with a particular Like as the TF term in the TF-IDF computation for this Like. For computing the weights for the social strength, user similarity,
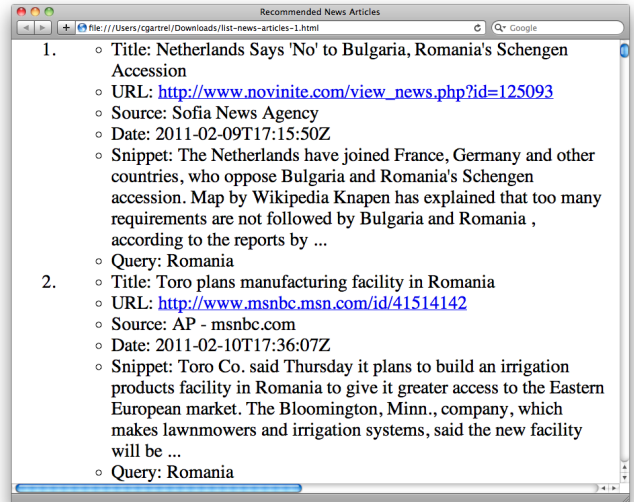

Figure 3: Screenshot of preliminary SocialNews recommendations

and Like-profile components in the ensemble list of recommendations, we use the number of clicks associated with each source.

Clickthrough data alters the ranking of friends in the social strength and user similarity components by setting a click multiplier for a friend based on the number of clicks associated with recommendations from that friend. The value of the click multiplier is scaled linearly according to this click count as follows

$$w_f = k \cdot c_f \cdot s_f \qquad (5)$$

where w_f is the scaled social strength or user similarity score for friend f, k is a constant, c_f is the click count, and s_f is the initial social strength or user similarity score for a friend computed without regard to click data.

4. PRELIMINARY RESULTS

We are currently implementing the SocialNews backend and user interface. We have used the Spring Roo framework [4] to implement the backend data persistence and recommendation compute logic. SocialNews provides a Web-based user interface implemented using the Google Web Toolkit [2]. We use the Facebook Platform OAuth 2.0 protocol [1] for SocialNews user authorization and authentication, and the Facebook Graph API [3] to retrieve data from the Facebook social graph.

To test the viability of SocialNews, one of the authors of this paper constructed a manually-ranked list of his top 20 Facebook friends, submitted the first five Facebook Likes for each friend to Bing News search, and generated a list of recommended news articles from the Bing News search results. The recommendation results include articles on several topics related to close friends
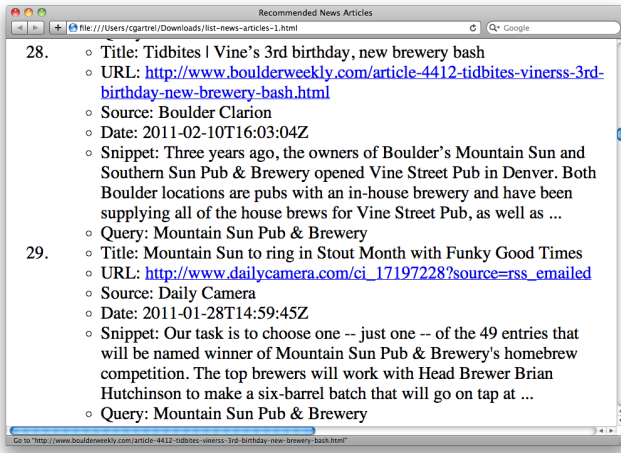
Figure 4: Screenshot of preliminary SocialNews recommendations

and family that are of interest to the author, as seen in the screenshots in Figure 3 and Figure 4. For example, Figure 3 shows recommendations results generated from the author's Romanian spouse, while Figure 4 shows recommendation results from a friend who is interested in a local pub and brewery in Boulder, Colorado. These results provide some initial indication of the feasibility and promise of SocialNews. We are currently in the process of finalizing the implementation and internal testing of SocialNews, and plan to soon conduct user studies with a number of Facebook users.

5. CONCLUSIONS

In this paper, we have described the design and implementation of SocialNews, a system that leverages social network information to provide relevant news recommendations. Facebook profiles provide significant insight into a user's personal interests and the nature of social connections. SocialNews mines this information and provides relevant news recommendation according to the user's similarity to others, strength of social connections, and the user's own personal interests. We process the user's clicks on recommended news articles to dynamically alter the computation of recommendations based on changes in the user's interests over time. Preliminary results point to the promise of SocialNews. We plan to conduct user studies with SocialNews to evaluate the performance of the system across a number of users.

6. REFERENCES

[1] Authentication - facebook developers. `https://developers.facebook.com/docs/authentication/`.
[2] Google web toolkit. `http://code.google.com/webtoolkit/`.
[3] Graph api - facebook developers. `https://developers.facebook.com/docs/reference/api/`.
[4] Spring roo. `http://www.springsource.org/roo`.
[5] 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*, 17:734–749, 2005.
[6] J. Chen, R. Nairn, and E. Chi. Speak little and well: recommending conversations in online social streams. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 217–226. ACM, 2011.
[7] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: Experiments on recommending content from information streams. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1185–1194. ACM, 2010.
[8] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *Proc. of WWW '07*, pages 271–280, 2007.
[9] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, 2007.
[10] E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 211–220. ACM, 2009.
[11] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.
[12] X. Jin, C. Wang, J. Luo, X. Yu, and J. Han. Likeminer: A system for mining the power of 'like' in social media networks. In *KDD'11: Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2011. Demo.
[13] H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King. Recommender systems with social regularization. In *WSDM'11: Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.
[14] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 981–990. ACM, 2010.
[15] S. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 537–546. ACM, 2011.
[16] S. Zhao, M. Zhou, Q. Yuan, X. Zhang, W. Zheng, and R. Fu. Who is talking about what: Social map based recommendation for content centric social websites. In *RecSys'10: Proceedings of the fourth ACM conference on Recommender systems*, 2010.

Input: Clickthrough data and lists of top k friends
Output: Ensemble list of recommendations

```
 1 begin
 2 |   begin
        /* Get recommendation lists        */
 3 |   |   socialStrenRecs =
        getRecommendations(socialStrenTopKFriends);

 4 |   |   userSimRecs =
        getRecommendations(userSimTopKFriends);

 5 |   |   ownProfileRecs =
        getRecommendations(ownTopKLikes);
 6 |   end
 7 |   begin
        /* Get clicks counts for
           recommendation sources          */
 8 |   |   socialStrenClickCnt =
        getClickCountForSource(SOCIAL);
 9 |   |   userSimClickCnt =
        getClickCountForSource(SIMILARITY);
10 |   |   ownProfileClickCnt =
        getClickCountForSource(SELF);
11 |   |   totalClickCnt =
        getClickCountForSource(ALL);
12 |   end
13 |   begin
        /* Get weights for recommendation
           sources                         */
14 |   |   socialStrenWeight = socialStrenClickCnt
        / totalClickCnt;
15 |   |   userSimWeight = userSimClickCnt /
        totalClickCnt;
16 |   |   ownProfileWeight =
        ownProfileClickCnt / totalClickCnt;
17 |   end
18 |   begin
        /* Get subsets from recommendation
           lists by weight                 */
19 |   |   subsetSocialStrenRecs =
        getSample(socialStrenWeight,
        socialStrenRecs);
20 |   |   subsetUserSimRecs =
        getSample(userSimWeight,
        userSimRecs);
21 |   |   subsetOwnProfileRecs =
        getSample(ownProfileWeight,
        ownProfileRecs);
22 |   end
23 |   ensembleRecs = subsetSocialStrenRecs +
        subsetUserSimRecs +
        subsetOwnProfileRecs;
24 end
```

Algorithm 1: Build ensemble list of recommendations