Temporal Top-k Search in Social Tagging Sites Using Multiple Social Networks

Wenyu Huo and Vassilis J. Tsotras

Department of Computer Science and Engineering,
University of California, Riverside, CA, 92521, USA
{whuo,tsotras}@cs.ucr.edu

Abstract. In social tagging sites, users are provided easy ways to create social networks, to post and share items like bookmarks, videos, photos and articles, along with comments and tags. In this paper, we present a study of top-k search in social tagging sites by utilizing multiple social networks and temporal information. In particular, besides the global connection, we consider two main social networks, namely the friendship and the common interest networks in our scoring functions. Based on the degree of participation in various networks, users can be categorized into specific classes that differ in their weights on each scoring component. Temporal information, usually ignored by previous works, can enhance the popularity and freshness of the ranking results. Experiments and evaluations on real social tagging datasets show that our framework works well in practice and give useful and intuitive results.

1 Introduction

The advent of Web 2.0 has facilitated the growth of online communities and applications such as blogs, wikis, online social networks and social tagging sites. In social tagging sites, such as del.icio.us, Flickr, and CiteULike, once a user is logged in, he can easily edit his own personal profile, build social networks with friends, and contribute content by posting bookmarks, videos, photos, or articles. He can also annotate those items with arbitrary tags.

Social tagging sites are free, fun, and functional, attracting more and more people to register as users. Moreover, social tagging sites have formed and stored plenty of valuable information like user-generated items, user social networks, and user tags. How to make good use of this information to improve services such as hot-lists, recommendations and web search is an open and attractive challenge for both academia and industry.

In this paper, we focus on temporal ranking and personal search in social tagging sites. When compared to other related work such as [1,6], our contributions are: First, we apply multiple components to score an item with respect to a particular user's different social networks and assign weights based on the classification of that user's participation in those networks. Then, we take temporal information into account, to enhance popularity and freshness of the top-k

H. Kitagawa et al. (Eds.): DASFAA 2010, Part I, LNCS 5981, pp. 498–504, 2010.

results. We provide a variation of the classic top-k algorithm which works efficiently for our user-dependent temporal scoring functions. Last, experimental evaluations on real social tagging datasets show that our framework works well in practice.

2 Data Model

Previous work in social tagging mostly ignores temporal information, only considering three factors: *User*, *Item*, and *Tags*. We extend the tagging behavior by adding timestamps: <*User, Item, Tags, Timestamp*>, which indicates that a user annotated one item with arbitrary tags at some time. In the following, we first demonstrate the model of social networks and static scoring functions without timestamps, and then explore a method to incorporate temporal information into ranking.

In social tagging sites, users are generally participating in multiple social networks. Aside from the global connection (*Global*), meaning everyone is connecting with anyone else on the whole web, we consider two other main kinds of social networks, namely, friendship (*Friends*) and common interest networks (*Links*).

Friendship is a kind of *explicit* social network. One user can choose to add any other users as friends. Most of them could be acquaintances in real life—friends, schoolmates, business contacts, etc; some may be known through the internet. We use $\mathit{Friends}(u)$ to represent all users in a friendship with user u. Social tagging sites enable users to create and join special groups. This is also an explicit social network, since group members have direct connections with each other. We categorize groups into *Friends* as well.

We also consider another kind of social network called common interest network [1]. Different from the traditional explicit social networks built up by adding friends or joining groups, the common interest network is *implicit* in nature, formed based on similar tagging behaviors. The items posted and the tags used by a user can be considered indicators of that person's interests. Linking people together whose tagging behaviors overlap significantly can implicitly form common interest networks.

For example, Let $\mathit{Items}(u)$ be the set of items tagged by the user u with any tag. Using $\mathit{Links}(u)$ to represent the common interest network for the user u, we could define that another user v is in $\mathit{Links}(u)$ iff a large fraction of the items tagged by u are also tagged by v, as $|\mathit{Items}(u) \cap \mathit{Items}(v)| > \theta$, where θ is a given threshold.

Given a query $Q = t_1, \ldots, t_n$ with n tags, issued by user u, and a number k, we want to efficiently return the top-k items with the highest overall scores. Our search strategy is user-focused, giving different results to different users. Our scoring functions consider the user's multiple social networks. Moreover, the top-k results returned take into account tagging behaviors' temporal information.

3 Scoring Functions

The static scoring functions for each social network component and overall combined scores are initially described. A method of weight assignments based on user classification is then discussed. Finally, temporal information of tagging behaviors is added and temporal scoring functions are examined.

3.1 Multiple Social Network Components

The overall static scoring function needs to aggregate three social network components: friendship, common interest network, and global connection.

Given a user u, the friendship component score of an item i for a tag t is defined as the number of users u's friends who tagged i with tag t:

$$Sc_F(i, u, t) = |Friends(u) \cap \{v|Tagging(v, i, t)\}| \tag{1}$$

Similarly, the score from common interest network is defined as the number of users in u's *Links* who tagged i with tag t:

$$Sc_L(i, u, t) = |Links(u) \cap \{v|Tagging(v, i, t)\}| \tag{2}$$

Besides the above two scoring component from a user's social networks, we also consider the global effect on scoring. Not everyone is an active participant; if we only use the local social network scoring, the search effectiveness may decrease. The *Global* score, which is user-independant, is defined as the total number of users in the whole website tagged item i with tag t:

$$Sc_G(i, t) = |\{v|Tagging(v, i, t)\}| \tag{3}$$

As a result, the static overall score of item i for user u with one tag t is an aggregate function of the weighted scores from the three components:

$$Sc_O(i, u, t) = w_1 * Sc_G(i, t) + w_2 * Sc_F(i, u, t) + w_3 * Sc_L(i, u, t) \tag{4}$$

where w_i is the weight of each component and $\sum_{i=1}^{3} w_i = 1$

Since a query contains multiple tags, we also define the static overall $SCORE$ of item i for user u with the whole query $Q = t_1, \ldots, t_n$ as the sum of the scores from individual tags, which is a monotone aggregation function:

$$SCORE(i, u) = \sum_{j=1}^{n} Sc_O(i, u, t_j) \tag{5}$$

3.2 User Classification

Different weight assignments of components can generate different overall scores. There are several ways to assign component weights using machine learning or statistics methods. However, those need a large amount of data such as user feedbacks and log records, which are not easy to access. For simplicity, we use a user classification method based on the social networks size and recommend weight assignments for each class.

Users in social tagging sites have different usage patterns and degrees of participation in their social networks. Some users have many friends, while some may only have few. Also, for tagging, some users do frequent tagging and thus have a lot of tagged items; while others may not tag as much.

In our general framework, we use three categories for *Friends* and *Links* social network component, described as: *many*, *some* and *few*; so there are nine classes totally. Within this classification, we assume that users in the same class have similar degree of trust on each social network scoring component. Then we can give a recommendation of weight assignments for users in each class.

3.3 Temporal Scoring Functions

We believe that ranking results will be more attractive to users not only based on their relevance, but also on popularity and freshness. For example, one item may be more interesting if it is recently added. In this case, a simple interpretation of freshness is the first date the item was posted. However, a more subtle way may consider how many recent tagging behaviors have targeted an item.

Our basic approach is to divide the tagging behaviors into multiple time slices, based on their time stamps. We use m to denote the number of time slices and adjust the weights of different time slices based on their recency. A decay factor a $(0 < a < 1)$ is used to penalize the count score from old time slices. Thus, the temporal score of Global component of item i with tag t can be defined as:

$$TSc_G(i,t) = \sum_{s=1}^{m} Sc_G(i,t,s) * a^{m-s} \tag{6}$$

where $Sc_G(i,t,s)$ is the global score of item i with tag t at time slice s, with $s = m$ being the current time slice.

The temporal scoring functions for *Friends* and *Links* components are defined similarly with the same temporal factors in *Global*:

$$TSc_F(i,u,t) = \sum_{s=1}^{m} Sc_F(i,u,t,s) * a^{m-s} \tag{7}$$

$$TSc_L(i,t) = \sum_{s=1}^{m} Sc_L(i,u,t,s) * a^{m-s} \tag{8}$$

The temporal overall scoring function of item i for user u with tag t is:

$$TSc_O(i,u,t) = w_1 * TSc_G(i,t) + w_2 * TSc_F(i,u,t) + w_3 * TSc_L(i,u,t) \tag{9}$$

Therefore, the temporal scoring for whole query is:

$$TSCORE(i,u) = \sum_{j=1}^{n} TSc_O(i,u,t_j) \tag{10}$$

4 Temporal Ranking Algorithm

Typically, one inverted list is created for each keyword and each entry contains the identifier of a document along with its score for that keyword [2]. For our framework, when the query is composed of multiple tags, we need to access multiple lists and apply the top-k processing algorithms.

One straightforward method is to have one inverted list for each (tag, user) pair and sort items in each list according to the temporal overall score (TSc_O) for the tag t and user u. However, there are too many users registered (del.icio.us has over 5 million users). If we create inverted lists per keyword for each user, there will be a huge amount of inverted lists and thus large space is required.

Another solution is to factor out the user from each inverted list by using upper-bound scores [1]. Since we use the number of users as the static score without normalization and set all three social network component with the same temporal factors for a query, for the same item i with the same tag t, no matter which user, we have $TSc_F <= TSc_G$ and $TSc_L <= TSc_G$. As a result, temporal global score is an upper-bound of temporal overall score for all users. Since the global component scoring is user-independent, we can create only one list for each keyword along with the temporal global scores (TSc_G) as an upper-bound of the user-based temporal overall scores (TSc_O).

The temporal factor can be designed as adjustable for users, so the temporal factors may also need to be factored out from the inverted lists. The static global scores (Sc_G) is an upper-bound for the temporal global scores (TSc_G), since the static scores correspond to the temporal ones with $a = 1$. Therefore, the final upper-bound scores used in the inverted lists are the static global scores (Sc_G).

We can thus extend Fagin's classic top-k TA algorithm [3] to rank the items listed in the order of static global scores (Sc_G) as the upper bound. When a new item is seen for the first time, we compute its exact temporal overall score (TSc_G) with a "local" aggregation function of three component temporal scores. The Algorithm stops whenever the score of the kth item in the heap is no less than the sum of bottom bounds of all lists. More details are covered in [4].

5 Experimental Evaluation

To evaluate the effectiveness of our scoring functions and algorithms, we collected real datasets from CiteULike (http://www.citeulike.org), an academic article social tagging site. In CiteULike, articles are stored with their metadata, abstracts, and links, and users can add tags and personal comments. CiteULike provides some datasets from their core database. However, to get more recent data, we further crawled datasets before 2009.7.1. An extended collection of our experimental evaluations appears in [4].

Here we use the NDCG (normalized discounted cumulated gain) measurement [5] to evaluate the performance of our experiments. Every item in top-k lists is given a corresponding human judgment scoring from 0 to 3 (0=Bad, 1=Fair, 2=Good, 3=Excellent) based on relevance and attractiveness (popularity and freshness) for particular query tags.

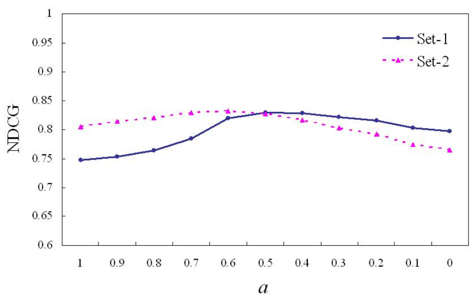

Fig. 1. Average NDCG results for different decay factor a

User		*Friends*		
		many	*some*	*few*
Links	*many*	Class 1	Class 2	Class 3
	some	Class 4	Class 5	Class 6
	few	Class 7	Class 8	Class 9

Class #	Recommendation
Class 1	r1: $w_1 = 0.1$, $w_2 = 0.45$, $w_3 = 0.45$;
Class 2	r2: $w_1 = 0.1$, $w_2 = 0.3$, $w_3 = 0.6$;
Class 3	r3: $w_1 = 0.1$, $w_2 = 0.1$, $w_3 = 0.8$;
Class 5	r5: $w_1 = 0.2$, $w_2 = 0.4$, $w_3 = 0.4$;
Class 6	r6: $w_1 = 0.2$, $w_2 = 0.3$, $w_3 = 0.5$;
Class 9	r9: $w_1 = 0.4$, $w_2 = 0.3$, $w_3 = 0.3$;

Fig. 2. User classification and weights recommendation for representative classes

Different queries may prefer different temporal factor settings, thus we use two different sets of popular query tags. For set-1, the queries are "social-network" and "tagging". These are popular and very hot recently. For set-2, we use "algorithm" and "database" separately as popular and classic queries.

We divide the time range of our datasets into six-month periods, starting from the most recent 2009.1.1 - 2009.6.30 to earlier time slices, which will remain the same throughout this paper. We change the decay factor a from 1 to 0, which means setting recency priority from low to high. and only evaluate the global temporal scoring (TSc_G) to factor out user diversity.

From the results in Fig. 1, we observe that different kinds of queries have different preferences. Hot queries may prefer recent tagging behaviors much more than classic queries. But for both sets, the average NDCG peaks when a is set 0.5 or 0.6, neither too high to miss the temporal information nor too low to lose classic items.

Then we evaluate the NDCG of different user classes with different weight assignments for each social network and we set *few* as 0-5, *some* as 6-15, and *many* as 15+ for both Friends and Links.

Based on the user classification, we provide an example recommendation of weight assignments for six representative classes in Fig. 2. The decay factor is set as $a = 0.5$ and the time slices are six-months. We tested two queries— "tagging" and "algorithm", picked up two users from our dataset for each class, and extracted the average NDCG. As shown in Fig. 3, in all six representative classes, our multiple-component method produced better NDCG than any other methods considering only one type of social network.

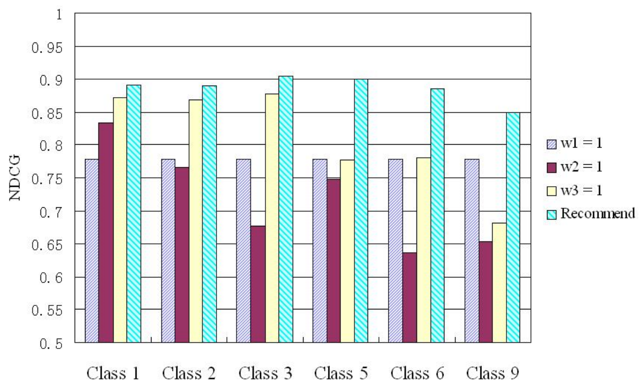

Fig. 3. Average NDCG for weight assignments across six representative classes

6 Conclusions

In this paper, we presented a study of temporal top-k search in social tagging sites using three main types of social networks, friendship, common interest networks, and global connections. To set the weights of each scoring component for different users, a classification method is proposed based on the size of users' social networks. To improve the popularity and freshness of ranking results, the timestamps of tagging behaviors are recorded and divided into multiple time slices and temporal scoring functions are formed by giving higher weights to more recent time slices. In addition, an efficient temporal top-k algorithm for ranking is proposed with upper-bound scores. Experimental evaluation on real datasets shows that our framework and methodology work well in practice.

Acknowledgments. This work is partly supported by NSF grants IIS-0705916 and IIS-0910859.

References

1. Amer-Yahia, S., Benedikt, M., Lakshmanan, L., Stoyanovich, J.: Efficient Network-Aware search in Collaborative Tagging Sites. In: VLDB (2008)
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (1999)
3. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. In: PODS (2001)
4. Huo, W.: Temporal Top-k Search in Social Tagging Sites Using Multiple Social Networks. Master's thesis, University of California-Riverside (2009)
5. Jarvelin, K., Kekalainen, J.: IR evaluation methods for retrieving highly relevant documents. In: SIGIR (2000)
6. Schenkel, R., Crecelius, T., Kacimi, M., Michel, Neumann, T., Parreira, J. X., Weikum, G.: Efficient top-k querying over social-tagging networks. In: SIGIR (2008)