Improved recommendation based on collaborative tagging behaviors

CONFERENCE PAPER · JANUARY 2008

DOI: 10.1145/1378773.1378843 · Source: DBLP

6 AUTHORS, INCLUDING:

Shiwan Zhao
IBM
13 PUBLICATIONS **258** CITATIONS
SEE PROFILE

Nan du
Georgia Institute of Technology
32 PUBLICATIONS **411** CITATIONS
SEE PROFILE

Andreas Nauerz
IBM
24 PUBLICATIONS **141** CITATIONS
SEE PROFILE

Improved Recommendation based on Collaborative Tagging Behaviors

Shiwan Zhao[1], **Nan Du**[2], **Andreas Nauerz**[3], **Xiatian Zhang**[1], **Quan Yuan**[1], **Rongyao Fu**[1]
[1]IBM China Research Laboratory, Beijing, 100094, China
{zhaosw,xiatianz,quanyuan,furongy}@cn.ibm.com
[2]Beijing University of Posts and Telecommunications, Beijing, 100876, China
dunan@bupt.edu.cn
[3]IBM Research and Development, Schoenacher Str. 220, Boeblingen, 71032, Germany
andreas.nauerz@de.ibm.com

ABSTRACT
Considering the natural tendency of people to follow direct or indirect cues of other people's activities, collaborative filtering-based recommender systems often predict the utility of an item for a particular user according to previous ratings by other similar users. Consequently, effective searching for the most related neighbors is critical for the success of the recommendations. In recent years, collaborative tagging systems with social bookmarking as their key component from the suite of Web 2.0 technologies allow users to freely bookmark and assign semantic descriptions to various shared resources on the web. While the list of favorite web pages indicates the interests or taste of each user, the assigned tags can further provide useful hints about what a user thinks of the pages.

In this paper, we propose a new collaborative filtering approach *TBCF* (Tag-based Collaborative Filtering) based on the semantic distance among tags assigned by different users to improve the effectiveness of neighbor selection. That is, two users could be considered similar not only if they rated the items similarly, but also if they have similar cognitions over these items. We tested *TBCF* on real-life datasets, and the experimental results show that our approach has significant improvement against the traditional cosine-based recommendation method while leveraging user input not explicitly targeting the recommendation system.

Author Keywords
Web2.0, Tag, Recommendation, Collaborative Filtering

ACM Classification Keywords
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval-Information Filtering

IUI'08, January 13-16, 2008, Maspalomas, Gran Canaria, Spain.

INTRODUCTION
In the common formulation, the recommendation problem is reduced to the problem of estimating the utilization for the items that have not been seen by a user[1]. Generally speaking, collaborative filtering approaches predict the rating of items for a particular user (active user) based on the ratings from other users with similar interests. For example, in a movie recommender system, one user's rating to a movie is a numeric score ranging from zero to five, which indicates how much the user likes or dislikes the movie. People holding similar rating patterns can form a neighborhood, and the ratings from these like-minded neighbors are used to produce predictions for the active user. Although the previous rating scheme can represent the extent to which a particular user may prefer a given item, it can not reveal the user's own opinions and understanding of the item. This rating scheme only demonstrates the strength of preference along a single dimension by scalar quantities. Yet, various users may give the same rating to an item from different points of view. For instance, both *Bob* and *Tom* may rate the movie *Transformers* with five stars, which indicates they all like this movie very much. Nevertheless, as a 3D fan, *Bob* appreciates this movie for its high quality 3D animations, while *Tom* may think that it is a wonderful action movie. Therefore *TBCF* only regards users as close neighbors if those users show a similar strength of interests from similar points of view.

One problem across collaborative recommendation systems is the small number of ratings obtained from each user, which is often referred to as the sparsity problem [3]. Since the success of any collaborative system depends on the availability of a critical mass of users, effective prediction from a small number of examples is important. With the development of modern Web 2.0 applications, tagging technique has become a powerful tool for semantically describing shared resources. Additionally, it can be regarded as another important way to implicitly rate interest, with an additional benefit of also providing semantics relative to that interest. Compared with the traditional ratings, tags can reflect both the user preferences and their opinions. That is, two users should be considered similar only if they rate items similarly and have similar cognitions over the items. By taking the semantic distance between tags assigned by different users into consideration, we propose a new collaborative filtering approach *TBCF* (Tag-

based Collaborative Filtering) to improve the effectiveness of neighbor searching.

The remainder of this paper is organized as follows: In section 2 we review some related work. Section 3 describes our tag-based recommendation approaches. Experimental results and analysis are given in section 4; and we conclude in section 5.

RELATED WORK

Research within the field of rating-based collaborative recommendations can be classified into two general groups: memory-based and model-based[1]. The rating-based approaches depend on the user explicitly rated items. However, many real-life applications find that people hesitate to give ratings explicitly[7]. Consequently, log-based approaches are often deployed, leveraging implicit interest functions to generate binary-valued preferences. More formally, let U and I be the set of all users and items. The specific value of the unknown rating $r(u_i, c)$ of item c for user $u_i (u_i \in U, c \in I)$ is often estimated from the ratings $r(u_j, c)$ given to item c by other similar users $u_j \in U, j \neq i$. In log-based collaborative recommendation, $r(u_j, c) = 1$ if u_j has once browsed or accessed item c; otherwise $r(u_j, c) = 0$. Suppose U_N denotes the set of N users who are the most similar to u_i. One of the most popular approaches to define $r(u_i, c)$ is to use the weighted sum as follows $r(u_i, c) = k \sum_{u_j \in U_N} sim(u_i, u_j) \times r(u_j, c)$. The weight $sim(u_i, u_j)$ is the similarity measure between users u_i and u_j. k is the normalizing factor such that the absolute values of the weights sum to unity. Since $sim(u_i, u_j)$ is a heuristic function, different recommendation systems, or a system operating in different application domains, may use different similarity measures of their own. Let I_{uv} be the set of all items accessed by both user u and v together. One commonly adopted similarity measure is to treat user u and v as two vectors in m-dimension space where $m = |I|$ accordingly. The similarity value between u and v is thus calculated as the cosine of the angle between the corresponding two vectors.

$$sim(u, v) = \frac{\overrightarrow{u} \cdot \overrightarrow{v}}{\|\overrightarrow{u}\| \cdot \|\overrightarrow{v}\|} = \frac{\sum_{c \in I_{uv}} r(u, c) \times r(v, c)}{\sqrt{\sum_{c \in I_u} r(u, c)^2} \times \sqrt{\sum_{c \in I_v} r(v, c)^2}}$$

Sood and *Owsley*[6] directly apply information retrieval techniques to the content part of the browsed web pages in order to compute $sim(u, v)$. However, it is obvious that not all shared resources can be compared directly by their contents. In particular, our proposed method uses tags as the feature vector of the resources rather than their contents. In addition to recommending pages, *TBCF* can also recommend any uniquely identifiable resource, such as images, videos and even people.

TAG-BASED RECOMMENDATION

Our recommendation approach consists of two parts. First we will adopt a method to calculate the semantic similarity of tags. Then based on this new similarity metric we present our collaborative recommendation approach.

WordNet-based Tag Similarity

WordNet is a public lexical database that provides a large repository of English lexical items. Each word in *WordNet* is stored in a structure called *synset* which is the basic unit of the whole dictionary. Every *synset* includes the word, its meaning and the corresponding synonyms. The meaning of a word is often referred to as *gloss* which actually defines the specific concept. Different meanings of a word correspond to different *synsets*. Terms with the synonymous meanings lie in the same *synset*. For example, the word *love* and *passion* constitute a *synset* with the *gloss*: *any object of warm affection or devotion*. All the *synsets* are organized by some basic semantic relations, such as "the part of" and "is a kind of". Therefore, the whole dictionary can be treated as a large graph with each node being a *synset* and the edges representing the semantic relations.

As discussed above, tagging provides users with means to categorize content autonomously, independent from any central administration. However, since tagging systems do not enforce fixed or controlled vocabularies for tag selection, the free choice of tags can result in two problems. First, multiple tags can have the same meanings, which is referred to as synonymy. Two tags may be morphological variation (apple vs. apples) or semantically similar (love vs. passion). Second, a single tag can have multiple meanings, which is often referred to as polysemy. For instance, a web page tagged with "apple" may be a post about fruits or can be interpreted as introducing *iPod*[6].

To address these problems, we first adopt *Porter*'s stemming algorithm[4] to remove the common morphological and inflexional endings of tags, so the morphological variation can be solved. Then we use *Satanjeev Banerjee*'s algorithm[2] to get rid of the semantic ambiguity of a particular tag in certain contexts. The basic idea of this approach is to count the number of words that are shared between two given glosses. The more common words they share, the more closely they are related. For example, suppose we have two tags *mouse* and *keyboard* together on a page.

According to *WordNet*, *mouse* has four meanings:
1. *any of numerous small rodents typically resembling diminutive rats having pointed snouts and small ears on elongated bodies with slender usually hairless tails.*
2. *a swollen bruise caused by a blow to the eye.*
3. *person who is quiet or timid.*
4. *a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad; on the bottom of the device is a ball that rolls on the surface of the pad.*

The tag *keyboard* has two different meanings:
1. *device consisting of a set of keys on a piano or organ or typewriter or typesetting machine or computer or the like.*
2. *holder consisting of an arrangement of hooks on which keys or locks can be hung.*

Table 1. Tag-based User-Item Matrix

	Item 1	Item 2	Item 3	Item 4
Alice	Art, photo	Home, Products	Writing , Design	Learning, Education
Daniel	Photo, Album, Image	$\emptyset$	Typewriter	Tutorial, Training
Sherry	$\emptyset$	Cleaning	$\emptyset$	Language, Study
Maggie	Photography	$\emptyset$	Ovens	$\emptyset$

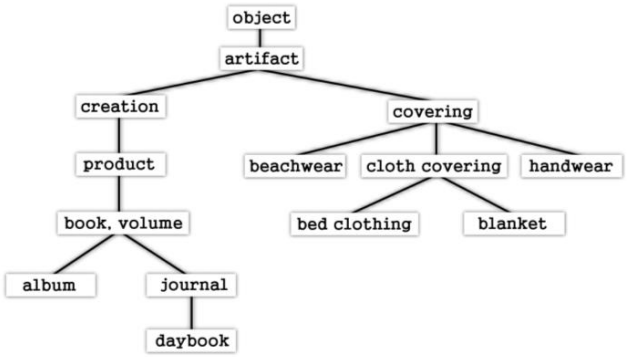

Figure 1. *WordNet* Concept Tree

Applying *Satanjeev Banerjee*'s algorithm to these tags, we can find that the fourth meaning of "mouse" and the first meaning of "keyboard" shares the words "computer" and "device". As a consequence, these two *glosses* are selected to define a semantic relation between "mouse" and "keyboard". Since the organization of *WordNet* can be regarded as an undirected graph, a single tag potentially having multiple meanings or glosses can therefore appear in multiple vertices. However, once a specific gloss of a tag is identified, the corresponding position or vertex of the tag in the graph is thus fixed. Therefore, the most straightforward method to calculate the semantic similarity between two given tags is to find the shortest path connecting them in the graph. In Figure 1, we see that "book" and "volume" are within the same *synset* indicating that they have the same meaning exactly. Thus the distance is zero. While, the distance between "journal" and "book" is 1, "bed clothing" and "blanket" is 2, and "album" and "blanket" is 7. For the tags that are not included in *WordNet*, we use *Levenshtein* algorithm[5] to calculate the edit-distance between them. Therefore, the total similarity measure is defined as $sim(x, y) = 1/(dis(x, y) + 1)$ if x and y are contained in *WordNet*. $dis(x, y)$ is the shortest path length between x and y. Otherwise, $sim(x, y) = 1 - (Lev(x, y)/maxlength(x, y))$, where either x or y is not contained, and $Lev(x, y)$ is the value of Levenshtein distance.

Tag-based Recommendation Approach

The basic idea of our recommendation approach is to find the top-N nearest neighbors by using the semantic similarity among tags. Formally, let U be the set of all users, I be the set of all items, and T be the set of all tags assigned on all the items by users in U. The new user-item rating matrix with tags is shown in Table 1.

We can observe that each element in the matrix is now the set of rated tags rather than the binary-value 1 or 0 compared with traditional log-based method. Since that the feature vector of each user is no longer in the m-dimension space $m = |I|$ of real numbers, we cannot directly calculate the cosine of the angle between these vectors. Alternately, we turn to compute the user similarity based on the similarity value of tag sets. In the first step, given two users $u, v \in U$, we obtain the set of common items I_{uv} by intersecting I_u with I_v. For each element $c \in I_{uv}$ the tag sets of user u and v on item c are defined as $T(u, c)$ and $T(v, c)$ respectively, where $T(u, c), T(v, c) \subseteq T$.

In the second step, the similarity calculation of two tag sets is formulated as computing a maximum total matching weight of a bipartite graph G which can be partitioned into two disjoint node sets $T(u, c)$ and $T(v, c)$ such that every edge connects a tag in $T(u, c)$ with a tag in $T(v, c)$ carrying the similarity value $sim(x, y)$ $x \in T(u, c), y \in T(v, c)$ as the weight. The *Hungarian* method is thus adopted to obtain the set of matching pairs defined as M. We then define the similarity value of user u and v as follows: $sim(u, v) = \sum_{c_i \in I_{uv}} \sum_{(x,y) \in M_{c_i}} sim(x, y)$, where $x \in T(u, c_i), y \in T(v, c_i)$. Based on this similarity measure, we find the top-N nearest neighbors U_N of the particular user u. For each element $c_k \in \overline{I} = I - I_u$, the predicted rating of c_k is defined as $R(u, c_k) = \sum_{v_i \in U_N} w(v_i) \times sim(u, v_i)$, where $w(v_i) = 1$ if v_i rates c_k; otherwise, $w(v_i) = 0$. In the end, we only return the top M predicted items to user u.

EXPERIMENTAL EVALUATION

We demonstrate the working of our approach on the datasets extracted from the web logs of *Dogear* with IBM *Lotus Connections* [8], which is an enterprise collaborative bookmarking system within IBM comparable with *del.icio.us*. We use a triple relation $< userID, pageID, tag >$ to record which user puts which tag on which page. Based on the above triple relation, we have extracted total 8000 users, 5315 pages and 7670 tags. We performed two experiments to compare the performance of *TBCF* with that of the classic rating-based algorithm which depends on a cosine-based similarity measure.

In the first experiment, the pages which were previously tagged by the user are withheld. We then generate a top-M list of recommendations and record whether each withheld page should appear or not. For each active user $u_i \in U$, I_{u_i} represents the set of pages that u_i has tagged. Let R_{u_i} be the set of pages that appear in the corresponding recommendation

Table 2. Statistical all but one results

Algorithm	Average Precision	Average Ranking
$TBCF$	0.27	2.8
$cosine$	0.13	1.5

list. The average precision is thus defined as

$$SABONE = (\sum_{u_i \in U} \frac{|R_{ui}|}{|I_{u_i}|})/|U|$$

We go through the whole user space and do the same experiment on each user. Then the statistical average is calculated for all the users, which is termed as "statistical all but one". This experiment measures the algorithms' performance when given as much data as possible from each active user. Higher *SABONE* corresponds to greater accuracy in the algorithm's recommendations. In our experiments, for each active user, we search for the top-5 most similar neighbors, and generate a top-10 recommendation list each time. The results of our *TBCF* approach and the cosine-based method are given in table 2, where the average ranking value reflects the average position that the withheld pages hold in the corresponding top-10 recommendation list.

In the second experiment, a random user is selected and a search performed for similar neighbors in a randomly generated subset of the user space instead of the whole one, which is termed as "random all but one" and focuses on the performance of the neighborhood selection. Here, we have randomly generated 4 subsets with the total user population being as 500, 2000, 4000, and 6000 accordingly. Table 3 shows the experimental results. The crucial step in collaborative filtering recommendation systems is the selection of the neighborhood. Traditionally, the similarity among users is only determined by the ratings given to co-rated items; items that have not been rated by both users are ignored. In *TBCF*, the similarity is based not only on the rating patterns of the users, but also based on their cognitions on the same items. We claim that this feature of *TBCF* makes it possible to select a better, more representative and more accurate neighborhood. For example, consider two users u and v with four common pages about *Java* technology. User u often browses these pages searching for the GUI programming techniques, but the other one often focuses on the aspects of *JSP* and *B/S* architecture. Traditional collaborative filtering would consider them similar. By contrast, *TBCF* would not consider them similar and may further find user w holding three common pages with v and having similar tags as *JSP*, *Tomcat* and *Struts*. Consequently, although the number of common pages between w and v is less than that between u and v, user w could be more similar to user v than user u and therefore they would be considered neighbors.

CONCLUSION AND FUTURE WORK

Incorporating the semantic information of the tags associated with the shared resources into collaborative filtering can significantly improve predictions of a recommender system. In this paper, we have provided an effective way to achieve this goal. We have shown how tag-based collaborative filtering outperforms the traditional cosine-based collaborative filtering method. *TBCF* uses the semantic similarity among the tags to significantly improve the neighborhood searching process which is a critical step for the collaborative recommendation system and directly determines the accuracy of the final predictions. We have tested *TBCF* on the dataset from the real-life applications. It can also help overcome the sparsity issue to some extent. For the future work, we seek to further improve the calculation of the semantic distance among new tags that are not stored in *WordNet*, and we could use the community wisdom reflected in social network analysis to further improve the neighborhood selection.

Table 3. Random all but one results

Random generated subset	Average Precision $TBCF$	Average Precision cosine
500	0.208	0.121
2000	0.182	0.118
4000	0.202	0.173
6000	0.209	0.180

ACKNOWLEDGMENTS

We thank Rich Thompson for his generous help and valuable comments, and thank Jonathan Feinberg and David R Millen for providing us the Dogear data.

REFERENCES

1. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.
2. S. Banerjee and T. Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *CICLing '02*, pages 136–145, London, UK, 2002. Springer-Verlag.
3. D. Billsus and M. J. Pazzani. Learning collaborative information filters. In *Proc. 15th International Conf. on Machine Learning*, pages 46–54. Morgan Kaufmann, San Francisco, CA, 1998.
4. W. Kraaij and R. Pohlmann. Porter's stemming algorithm for dutch, 1994.
5. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8, 1966.
6. S. Sood, S. Owsley, K. Hammond, and L. Birnbaum. Tagassist: Automatic tag suggestion for blog posts. March 2007.
7. J. Wang, A. P. de Vries, and M. J. Reinders. A user-item relevance model for log-based collaborative filtering. In *Proc. of European Conference on Information Retrieval (ECIR 2006), London, UK*, 2006.
8. "http://www-306.ibm.com/software/lotus/products/connections/dogear.html"