Web Page Recommender System based on Folksonomy Mining for ITNG '06 Submissions

Satoshi Niwa
University of Tokyo
niwa@nii.ac.jp

Takuo Doi
University of Tokyo

Shinichi Honiden
University of Tokyo
National Institute of Informatics

Abstract

There have been many attempts to construct web page recommender systems using collaborative filtering. But the domains these systems can cover are very restricted because it is very difficult to assemble user preference data to web pages, and the number of web pages on the Internet is too large. In this paper, we propose the way to construct a new type of web page recommender system covering all over the Internet, by using Folksonomy and Social Bookmark which are getting very popular in these days.

1. Introduction

As the scale of the Internet are getting larger and larger in recent years, we are forced to spend much time to select necessary information from large amount of web pages created every day. To solve this problem, many web page recommender systems are constructed which automatically selects and recommends web pages suitable for user's favor. Though various kinds of Web Pages have been constructed, there are many points to be improved in them.

Most of past web page recommender systems uses collaborative filtering. [1][2][3] Collaborative filtering is often used in general product recommender systems, and consists following two stages. [4]

1. Analyze users' purchase histories and extract user groups which have similar purchase patterns
2. Recommend products which are commonly preferred in the user's group

In general collaborative filtering, similarity between two users is considered '0' if the two users buy no common product. So, to calculate properly, system needs enough amount of purchase histories (N users), compared to the number of products M.

Most of past web page recommender systems replace "purchase histories" by "access logs". In this case, access logs are distributed to each web server, so such systems have to limit their web domains to particular web sites. Even if large amount of user preference data to the entire Internet is obtained, it is yet necessary to cluster web pages by their contents because the number of web pages is huge.

In this paper, we solve the problem of "lack of user preference data to web pages" by using "Social Bookmarks" as data source which are getting popular in these days, and proposes the way to construct a web recommender system which covers the entire Internet. By mining tag data of Folksonomy, we propose a new way to express users' preference to web pages. We also solve the problem of "tag redundancy in Folksonomy".

In what follows, we introduce Folksonomy and Social Bookmark in Chapter 2. In Chapter 3, we describe the algorithm and architecture of our web recommender system. In Chapter 4, we describe our experiments, and in Chapter 5, we examine the result and compare with related works. In the end, we conclude in Chapter 6.

2. Folksonomy and Social Bookmark

Recently Folksonomy and Social Bookmark are getting popular and spreading widely. Folksonomy is one of the components of Web 2.0 which is famous for Semantic Web. Social Bookmark is a web service using Folksonomy.

2.1. Folksonomy

Folksonomy [5] is a new classification technique which may take place of past taxonomy. In case of web page classification using taxonomy, someone constructs the classification tree at first, and pages are classified based on the tree. An example of taxonomy tree is Yahoo Directory. On the other hand, in Folksonomy, end users put keywords called "tags" to each page freely and subjectively, based on their sense of values. Anyone can choose any word as tag, and can put two or more tags to one page.

Figure 1 shows an example of tagging by two users. By analyzing large amount of tags, we can make a vari-

Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)
0-7695-2497-4/06 $20.00 © 2006 IEEE

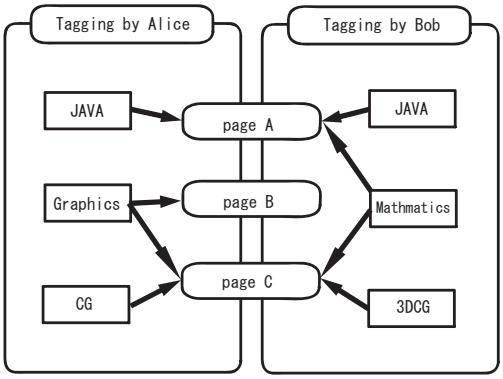

Figure 1. Tagging in Folksonomy

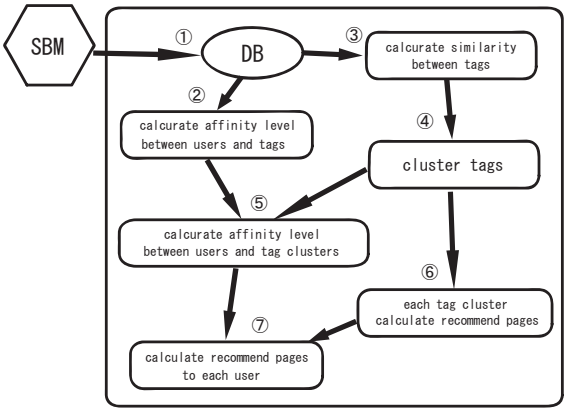

Figure 2. Entire System Architecture

ous kinds of classification. Folksonomy is a bottom up approach where users themselves join the classification, compared to top down taxonomy. By this nature, Folksonomy classification can reflect users' actual interest in real time.

On the other hand, one problem exists in Folksonomy. There is no limit for decision of tags in Folksonomy, so many similar tags are generated as usual. For example, user A puts tag "mathematics" to a mathematical page, and user B puts tag "math" to the same page. This happens very often in Folksonomy tagging, and called "tag redundancy in Folksonomy". We have to consider this when constructing systems which deals with Folksonomy.

2.2. Social Bookmark (SBM)

Social Bookmark (SBM) is a kind of web services on which users can share their bookmarks. Anyone can see anyone's bookmark on SBM. The most popular one is del.icio.us [6], and has been spread rapidly since the latter 2004. Users can put tags to bookmark pages, based on Folksonomy. Users' bookmarks are related dynamically by tags.

3. System Architecture

3.1. System outline

As described in Chapter 1, most past web recommender systems use collaborative filtering. In Basic collaborative filtering, users' preference are expressed as sets of products they purchased. But it doesn't work well if the number of products(pages) is too large, because in such case it's very hard to find similar users. To solve this problem, we express users' web page preference by "affinity level between each user and each tag". By this approach, users' preference are abstracted and it becomes easier to find similar users. In addition, we try to solve the problem of "tag redundancy in Folksonomy" by clustering tags. Clustering also shortens the time to calculate recommendation pages. Figure 2 shows the entire architecture of our system. First we introduce each stage shortly, and then describe in detail in following section.

1. Retrieve public bookmark data on SBM and store it in DB
2. Calculate affinity level between users and tags by using data acquired at stage 1
3. Calculate similarity between tags
4. Cluster tags by using the result of stage 3
5. Calculate affinity level between users and tag clusters by using the result of stage 2 and 4
6. Calculate recommendation pages in each cluster by using the result of stage 4
7. Calculate recommendation pages to each user by using the result of stage 5 and 6

3.2. Detail in each stage

3.2.1. Retrieve data on SBM All users' bookmark pages and tags put to the pages are public on general SBM. Figure 3 shows the model of public data on SBM. There are three kinds of objects (users, web pages, tags) connected with each other. The number put on the edge between tag T and page P shows the frequency that different user T to P. Figure 3 also shows information which user bookmarks which pages.

We express the number between page P and tag T as "w(P, T)" at the following. We also express the bookmark page set of user A as "bookmark(A)".

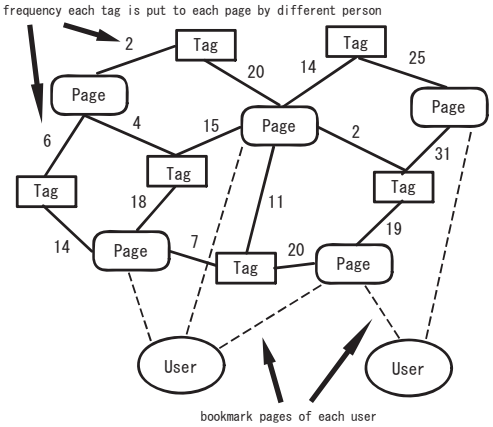

Figure 3. Model of SBM

3.2.2. Calculate affinity level between users and tags In this stage, the system calculates affinity level between each user and each tag. Affinity level is a scholar value which shows the degree of relation. For example, if user Bob bookmarks many pages about car, the affinity level between user Bob and tag " car " will be high.

Affinity level tends to be high if tag T is rare. We define " rel(A, T) " as affinity level between user A and tag T. To calculate this value, we first calculate " rel(P, T) ", which shows affinity level between page P and tag T.

$$\begin{aligned} rel(P,T) &= TF(P,T) \times IDF(T) \\ TF(P,T) &= \frac{w(P,T)}{\sum_{T_i \in TAGS} w(P,T_i)} \\ IDF(T) &= \log \frac{\sum_{P_j \in PAGES} \sum_{T_i \in TAGS} w(P_j,T_i)}{\sum_{P_j \in PAGES} w(P_j,T)} \end{aligned}$$

This formula is based on TF-IDF. In this case, TF(P, T) shows the ratio of tag T in all related tags to page P, and IDF(T) shows the rareness of tag T.

By using this result, we then calculate rel(A, T) as follows.

$$rel(A,T) = \sum_{P_i \in bookmark(A)} rel(P_i,T)$$

3.2.3. Calculate similarity between tags Using the similar algorithm as stage 2, the system calculates the similarity between each tag. "Similarity" also shows the degree of relation. For example, the similarity between tag " mac " and tag " apple " will be high.

We define " $rel(T_1, T_2)$ " as the similarity of tag T_2 from the view point of tag T_1, and the system calculates the value as follows. This value is not necessary the same as $rel(T_2, T_1)$.

$$rel(T_1,T_2) = \sum_{P_i \in PAGES} w(P_i,T_1) \times rel(P_i,T_2)$$

3.2.4. Cluster tags The system clusters high related tags based on the tag similarity values acquired at stage 3. By clustering tags, the system can treat tags with an appropriate size of topic. Clustering can also solve the problem of " tag redundancy in Folksonomy ". We cannot describe the clustering algorithm in detail here, so we briefly show the outline.

1. System calculates the " parent tag " of each tag, which is the most important and highest related tag to each tag
2. System generates tag clusters where each cluster has only one tag, which is also the " cluster leader " of each cluster.
3. Merge two clusters, if one cluster's leader's parent is included in the other cluster and similarity between the clusters exceeds certain threshold. The leader of the merged cluster becomes the leader of the latter cluster. This phase is repeated until no clusters meet the requirement to merge.

We can adjust the average size of clusters by changing condition parameters used in phase 3.

3.2.5. Calculate affinity level between users and tag clusters In this stage, the system calculates the affinity level between each user and each tag cluster, based on the affinity data of stage 2 and tag clusters of stage 4. By calculating this value, the user preference can be expressed by the units of topics. We define $rel(A, C)$ as the affinity level between user A and tag cluster C, calculated as follows.

$$rel(A,C) = \sum_{T_i \in C} rel(A,T_i)$$

3.2.6. Calculate recommendation pages in each tag cluster In this stage, the system calculates recommendation pages of each tag cluster, which is considered to be corresponds with each topic. First, we define $point(C, P)$ as "recommendation point" of page P in tag cluster C, and calculate as follows.

$$point(C,P) = \sum_{T_i \in C} w(P,T_i)$$

Pages with high points are chosen sequentially as recommendation pages.

	A	B	C	D	E
Number of clusters	2175	1430	877	512	412
Average cluster size	1.00	1.52	2.48	4.25	5.28
Average calculation time of stage 7 in each user (sec)	5.4	2.8	2.2	1.8	1.6
Size of cluster "English"	1	3	5	7	10
Size of cluster "Java"	1	11	15	30	34
Size of cluster "blog"	1	28	51	64	70

Figure 4. Each clustering case

3.2.7. Calculate recommendation pages to each user By using the affinity level of stage 5 and recommendation points of stage 6, the system then calculates recommendation pages to each user. First, we define $point(A, P)$ as "recommendation point" of user A in tag cluster C, and calculate as follows.

$$point(A, P) = \sum_{C_i \in CLASTERS} rel(A, C_i) \times point(C_i, P)$$

Pages with high points are chosen sequentially as recommendation pages.

4. Experiments

We constructed the system with Java 5.0, and did two types of experiments, (A) and (B). (A) is an experiment to evaluate recommendation accuracy by using data only on SBM. The purpose of experiment (A) is to find the optimal cluster size with best accuracy. (B) is an experiment in which human users evaluate the recommendation accuracy. In (B), the optimal cluster size acquired in (A) is used, and the absolute performance of the system is evaluated.

4.1. Experiments outline (common to (A) and (B))

This time, we choose "hatena bookmark"[7] as input SBM. It is the most popular SBM in Japan, with over 500,000 members and 1,000,000 bookmark pages.

As mentioned in 3.2.4, we can adjust the average size of tag clusters by changing parameters. This time, we did experiments with 5 different cluster sizes, shown in figure 4. In figure 4, A is the case in which no clusters are merged. The size of all the clusters in case A is 1, and the number of clusters is same as the number of all tags in case A. In contrast, the average cluster size of case E is the largest among 5 cases. We also show the average calculation time of stage 7 in each case. As you see, the number of clusters and the calculation time are roughly proportional. In our system, the stage 1 to 6 are previously executable before the test users input their bookmarks, so stage 7 is the only stage which needs quick calculation.

Figure 5 shows the members of "java" cluster in each case. Tags which are high related to "java" are seen in case

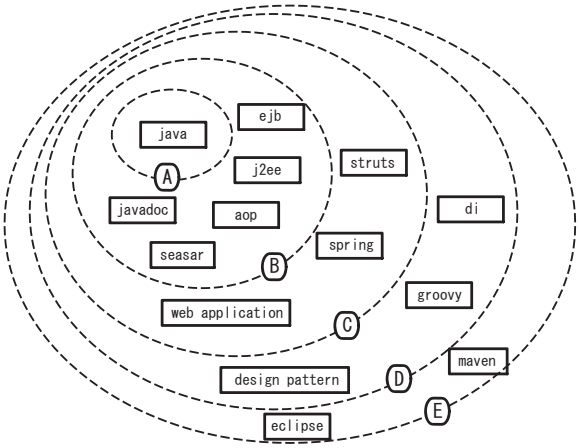

Figure 5. Members of "java" cluster in each case

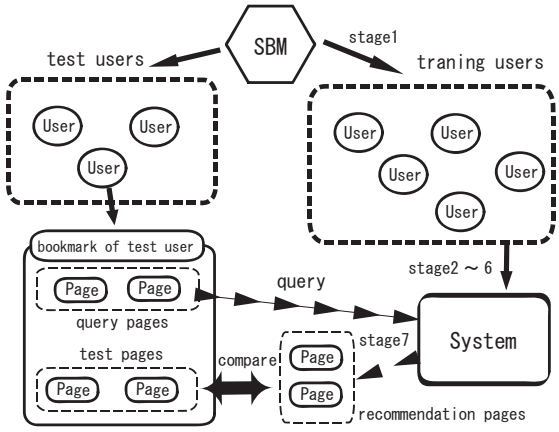

Figure 6. Experiment (A)

B and C, and tags which are related to "java" to some degree are seen in case D and E. Though somewhat subjective, clustering seems to be succeed to some degree.

We also found many "synonym clusters" in case B and C. Synonym cluster is a cluster composed of synonym tags. For example one cluster is composed of tag "mac", "mackintosh", "mac os". This means that the problem of "tag redundancy in Folksonomy" is solved to some degree by tag clustering.

4.2. Experiment (A)

4.2.1. Evaluation in experiment (A) In experiment (A), we evaluates the performance of the system by using only the data of SBM, changing the cluster size. Figure 6 shows the outline of experiment (A).

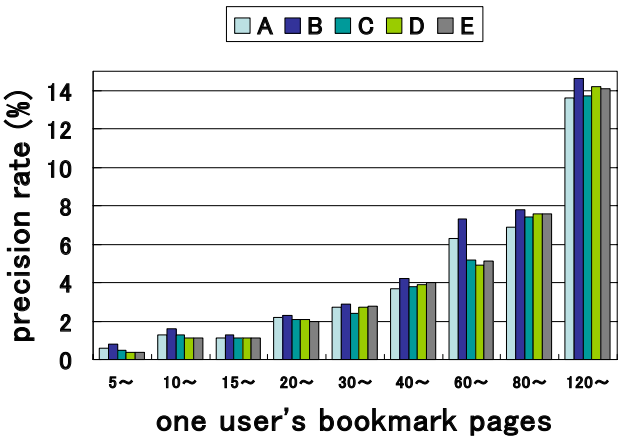

Figure 7. Precision rate in experiment (A)

At first, the system divides the SBM users randomly into "training users" and "test users". Then at stage 1, the system stores only the data of training users into DB. Stage 2 to 6 are also executed with this data.

Next, for each test user, system processes as follows. The system divides bookmark pages of each test user into "query pages" and "test pages". Query pages are used as the input of stage 7. Then the system outputs the recommendation pages to the test user. By comparing these output pages with test pages, we evaluate the precision rate of the recommendation pages.

We define "precision rate" as follows.

$$PrecisionRate = \frac{TestPages \cap RecommendationPages}{RecommendationPages}$$

4.2.2. Result of experiment (A) We experimented with five different clustering cases in figure 4. We divided test users into groups based on the number of bookmark pages they have. This is because the number of test user's bookmark pages influences the performance a lot. We used the half of each test user's bookmark pages as query pages, and the remainder as test pages. We fixed the number of output recommendation pages to 30.

Figure 7 shows the average precision rate in each clustering case. The horizontal axis corresponds to the number of each user's bookmark pages.

We can say from figure 7 that the number of test user's bookmark pages and the system precision rate is proportional. If the number of input pages is larger, the system can analyze user's preference more precisely, so this is a natural result.

Among five clustering cases, the precision rate of case B is totally the best. The precision rate of case B is always higher than case A (in which no clustering has been done). This is probably because a little amount of clustering in case B solved the problem of "tag redundancy in Folksonomy". By clustering synonym tags, the analysis performance of users' preferences was improved to some degree.

On the other hand, the precision rate of case D and E are not necessary higher than case A. This is probably because the degree of clustering is too big, and the users' preferences are abstracted too much.

We can say from this experiment that the level of tag clustering should not be either too big or too small. In following experiment, we use case B as default.

4.3. Experiment (B)

4.3.1. Evaluation in experiment (B) In experiment (A), we evaluated the performance of the system with large amount of test data on SBM. But the precision rate in experiment (A) only means "ratio of pages which are already bookmarked among recommendation pages".

In experiment (B), we evaluate the actual precision rate, that is, "ratio of pages which are acceptable by actual user". Ten users joined the experiment. The process of the experiment is as follows.

First, each user inputs the bookmark data of his usual web browser to the system. Then the system analyzes the bookmark and outputs 30 recommendation pages to the user. The user evaluates each recommendation page subjectively by the following three choices.

(a) I'm very interested in the page, and it is high related to me.

(b) I have a little interest in the page, but it is not especially related to me.

(c) I have no interest in the page, and it is not related to me.

We total these choices and calculate the actual precision rate of the system.

4.3.2. Result of experiment (B) As well as experiment (A), we divided the users into two groups by the number of bookmark pages. User group (1) contains users of less than 50 bookmark pages, and user group (2) contains users of 50 or more bookmark pages.

Figure 8 shows the precision rate of each user group. The horizontal axis describes how many recommendation pages are used in calculation of precision rate. For example, if this value is 10, only ten high rank pages are used in calculation. "(a)ratio" in figure 8 means the ratio of choice (a), and "(a)(b)ratio" means the sum ratio of (a) and (b).

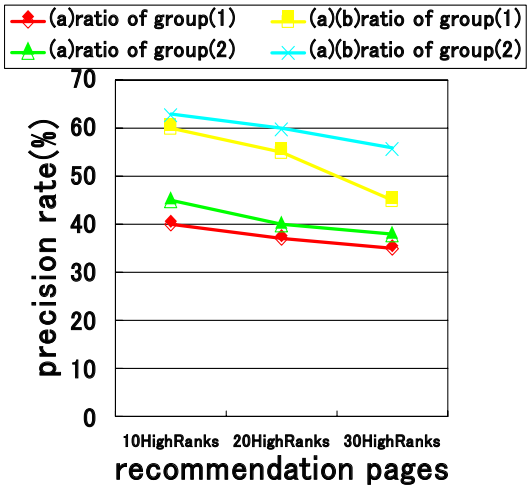

Figure 8. Precision rate in experiment (B)

According to the expectation, the precision rate of user group (2) is higher than (1). The precision rate is also higher when calculating with only high rank pages.

From this experiment it can be said that the precision rate of our system is about 40% to 60%.

5. Related work

The precision rate of our recommender system is about 40% to 60%. This is not especially a high value, but this is not lower than other web recommenders systems, either. Our system doesn't need any extra user operation other than inputting usual bookmark data, while most of other web recommender systems need some operation.

As mentioned before, most past web recommender systems use collaborate filtering with access logs. J Li et al.[1] constructed a web recommender system by analyzing site navigation patterns of users. They did content mining and structure mining of web pages in addition to mining access logs. But their system is limited to particular web sites like many other systems.

T Zhu et al. [8] constructed "Web ICLite", a famous web recommender system which is not limited to particular web sites. Web ICLite analyzes web navigation patterns by logging actions from user side. Each user's favorite pages and favorite keywords are analyzed in each user machine. The system gathers and merges the data, and calculates recommendation pages to each user. Web ICLite and our system are common in the point that both calculate recommendation pages in each unit of topic.

User tests which are very similar to ours were done with Web ICLite. The result is as follows. (a)ratio of Web ICLite is about 30 to 50%, and (a)(b)ratio is about 60 to 70%. Though comparison is not easy because the result of user tests depends greatly on each user's subjectivity, this value is almost as high as our system. But to use Web ICLite, users have to do web navigation with Web ICLite operating background for a certain period, while our system needs no extra operation other than preparing usual bookmark data.

C Nicolas et al. [9] proposed "Taxonomy - Driven similarity Metrics", the way to abstract users' preference by mining taxonomy tree. Our approach is based on this method, so it can be said "Folksonomy - Driven similarity Metrics".

6. Conclusion

We constructed a new web recommender system which is not limited to particular web sites, based on large amount of public bookmark data on SBM. We also utilize Folksonomy tags to classify web pages and to express users' preferences. By clustering Folksonomy tags, we can adjust the abstraction level of users' preferences to the appropriate level. We also solved the problem of "tag redundancy in Folksonomy" by clustering tags.

References

[1] J Li, O Zaiane : Combining Usage, Content, and Structure Data to Improve Web Site Recommendation, *Proceedings of "WebKDD-2004 workshop on Web Mining and Web Usage*,(2004).

[2] P Kazienko, M Kiewra : Integration of relational databases and Web site content for product and page recommendation, *Database Engineering and Applications Symposium, 2004. IDEAS*,(2004).

[3] N Golovin, E Rahm : Reinforcement Learning Architecture for Web Recommendations, *Proceedings of the International Conference on Information*,(2004).

[4] A Mathes - Retrieved : Evaluating collaborative filtering recommender systems *ACM Transactions on Information Systems*,(2004).

[5] J Golbeck, B Parsia, J Hendler : Folksonomies-Cooperative Classification and Communication Through Shared Metadata, (2004).

[6] del.icio.us : http://del.icio.us/

[7] hatena bookmark : http://b.hatena.ne.jp/

[8] Tingshao Zhu, Russ Greiner, Gerald Haeubl, Bob Price, Kevin Jewell : A Trustable Recommender System for Web Content, *International Conference on Intelligent User Interfaces*,(2005).

[9] Cai-Nicolas Ziegler : Semantic Web Recommender Systems, *Proceedings of the Joint ICDE/EDBT, Ph. D. Workshop*,(2004).