Mining Clickthrough Data for Collaborative Web Search

Jian-Tao Sun[†],Xuanhui Wang[‡],Dou Shen[§],Hua-Jun Zeng[†], Zheng Chen[†]

[†]Microsoft Research Asia, Beijing, P.R.China
{jtsun,hjzeng,zhengc}@microsoft.com

[‡]Department of Computer Science,
University of Illinois at Urbana-Champaign
xwang20@cs.uiuc.edu

[§]Department of Computer Science,
Hong Kong University of Science and Technology
dshen@cs.ust.hk

ABSTRACT

This paper is to investigate the group behavior patterns of search activities based on Web search history data, i.e., clickthrough data, to boost search performance. We propose a *Collaborative Web Search (CWS)* framework based on the probabilistic modeling of the co-occurrence relationship among the heterogeneous web objects: users, queries, and Web pages. The CWS framework consists of two steps: (1) a *cube-clustering* approach is put forward to estimate the semantic cluster structures of the Web objects; (2) Web search activities are conducted by leveraging the probabilistic relations among the estimated cluster structures. Experiments on a real-world clickthrough data set validate the effectiveness of our CWS approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval-Search Process; H.3.5 [**Information Storage and Retrieval**]: Online Information Services-Web based services

General Terms

Algorithms, Experimentation, Performance

Keywords

Clickthrough Data, Cube-Clustering, Collaborative Web Search

1. INTRODUCTION

Typical search engines conduct retrieval without considering the users' preferences. It is not appropriate since users with different interests may expect to get different Web pages even with the same query. Recently, a number of methods have been developed for the "Collaborative Filtering" or "Social Filtering" of information [1]. The idea is to recommend products or services to a person based on the opinions of a group of users who share similar preferences with him/her. In fact, as more and more Web users start using search engines, the search activity can also be regarded as a social behavior. All search activities are recorded in the clickthrough data, which contains a large number of Web objects (users, queries, and pages). Organizing these Web objects into semantic groups and analyzing the relationships among these groups will be potentially helpful for discovering meaningful patterns, such as different users may share similar search behaviors by issuing similar queries and visiting similar Web pages, thus improve the utility of Web search.

However, mining clickthrough data is really challenging: (1) the clickthrough data contains heterogeneous objects, including users, queries, and Web pages, and the relationship among these objects are complicated; (2) since users always only click the first few pages for a query, the data is highly sparse. In order to effectively conduct clickthrough data analysis for Web search, it is a key factor to handle the sparseness problem and discover the hidden relations among the heterogeneous data objects.

In this paper, we propose a framework to overcome the above difficulties. This framework contains the following two steps. (1) The hidden cluster structures contained in the clickthrough data, as well as the probabilistic relations among them, are discovered by an unsupervised method: cube-clustering. (2) The relations among the hidden clusters are used to improve the search performance in a collaborative manner. Since we exploit the group structures of the clickthrough data and our approach is motivated by the collaborative filtering idea, we name it Collaborative Web Search (CWS). In [3], the authors also developed a collaborative search system named I-SPY. Their system is a type of meta-search engine and requires users to explicitly select a community before search activities are conducted. In our CWS framework, the clickthrough data is automatically utilized and Web users' manual efforts are not required.

2. COLLABORATIVE WEB SEARCH

In this section, we describe the two steps of our CWS framework one by one.

2.1 Cube-Clustering Approach

Inspired by Dhillon et al.'s Co-Clustering algorithm for clustering two-dimensional co-occurrence data [2], we put forward the Cube-Clustering approach which clusters users, queries, and Web pages simultaneously based on the three-

WWW 2006, May 22–26, 2006, Edinburgh, Scotland.
ACM 1-59593-332-9/06/0005.

dimensional co-occurrence among them. Our Cube-Clustering algorithm is based on information theory and maximizes the *multi-information* among a set of random variables, which is defined as:

$$I(X_1, \cdots, X_s) = \sum P(X_1, \cdots, X_s) \log \frac{P(X_1, \cdots, X_s)}{P(X_1) \cdots P(X_s)}.$$

We treat a cube data of the three dimensions: user, query, and Web page, as a joint probability distribution among three discrete random variables: $Pr(U, Q, P)$. Variables U, Q and P take values from the user set $\{u_1, \cdots, u_l\}$, query set $\{q_1, \cdots, q_m\}$, and page set $\{p_1, \cdots, p_n\}$. Our goal is to cluster the l users, m queries, and n pages into i, j and k clusters respectively: $\{\hat{u}_1, \cdots, \hat{u}_i\}$, $\{\hat{q}_1, \cdots, \hat{q}_j\}$, $\{\hat{p}_1, \cdots, \hat{p}_k\}$. We use C_U, C_Q, and C_P to denote the mapping functions and refer to the triple (C_U, C_Q, C_P) as a *cube clustering*.

Given the mapping functions (C_U, C_Q, C_P), $C_U(U)$, $C_Q(Q)$, and $C_P(P)$ are the cluster random variables and we denote them as $\hat{U}$, $\hat{Q}$, $\hat{P}$ respectively. Apparently, a fixed cube clustering mapping determines the joint probability of the cluster random variables. We judge the quality of cube clustering by the loss in multi-information. The multi-information is a measurement to capture the amount of information that a set of variables contain about each other. Naturally, a good clustering should preserve the information of the original data as much as possible, thus minimize the loss in multi-information: $I(U, Q, P) - I(\hat{U}, \hat{Q}, \hat{P})$. This is also the objective function that an optimal cube clustering minimizes, subject to the constraints on the cluster numbers in each dimension. For a cube clustering, we find the loss in multi-information is equal to the KL divergence between $Pr(U, Q, P)$ and $\tilde{Pr}(U, Q, P)$, where $\tilde{Pr}(U, Q, P)$ is a distribution of the form

$$\tilde{Pr}(u, q, p) = Pr(\hat{u}, \hat{q}, \hat{p}) Pr(u|\hat{u}) Pr(q|\hat{q}) Pr(p|\hat{p}) \quad (1)$$

where $u \in \hat{u}, q \in \hat{q}, p \in \hat{p}$. That is, the cube clustering can be approached by minimizing the KL divergence between $Pr(U, Q, P)$ and $\tilde{Pr}(U, Q, P)$. We can also prove that the loss in multi-information can be expressed as a weighted sum of the KL-divergence between two distributions associated with a fixed dimension. Thus the calculation of the objective function can be solely expressed in terms of user clustering, query clustering, or page clustering. Based on this conclusion, we have the Cube-Clustering algorithm. We omit all the proofs for the space reason.

The input of the Cube-Clustering algorithm is the joint probability $Pr(U, Q, P)$. Assume i, j, k are the desired number of clusters for each dimension. The output is the partition function C_U, C_Q, C_P. The Cube-Clustering algorithm is described as follows:

Step 1: Start with some initial partition functions, thus for each u, q, p, we have its corresponding cluster. Compute

$$\tilde{Pr}(\hat{U}, \hat{Q}, \hat{P}), \tilde{Pr}(U|\hat{U}), \tilde{Pr}(Q|\hat{Q}), \tilde{Pr}(P|\hat{P}) \quad (2)$$

Step 2: (2.1) Calculate the distributions $\tilde{Pr}(Q, P|\hat{u})$, $1 \leq \hat{u} \leq i$ *using*

$$\tilde{Pr}(q, p|\hat{u}) = \tilde{Pr}(p|\hat{p}) \tilde{Pr}(q|\hat{q}) \tilde{Pr}(\hat{q}, \hat{p}|\hat{u}) \quad (3)$$

(2.2) Update user clusters: for each u, find its new cluster index as

$$C_U = argmin_{\hat{u}} KL(Pr(Q, P|u) || \tilde{Pr}(Q, P|\hat{u}))$$

(2.3) Update the distributions listed in Eq 2.

Table 1: Search Results of CF and CWS

N	$P@N$ (CF)	$P@N$ (CWS)	Relative Improvement
1	0.581887	0.632321	8.66%
2	0.315347	0.339479	7.75%
3	0.214571	0.232106	8.17%
4	0.162961	0.176383	8.23%
5	0.131669	0.141973	7.82%

Step 3: Process queries and pages symmetrically as in step 2.

Step 4: Iterate Step 2 and Step 3 until the change in objective function is less than a small value, e.g., 10^{-6}*; else go to step 2.*

2.2 Search Based on Cube-Clustering

After the cube-clustering step, the Web search problem is converted to recommendation of a ranked page list according to their relevance with the $\langle u, q \rangle$ pair, instead of depending only on q. In our CWS framework, we rank Web pages by estimating $\tilde{Pr}(p|u, q)$. Given u and q,

$$\begin{aligned} \tilde{Pr}(p|u, q) &= \frac{\tilde{Pr}(u, q, p)}{\tilde{Pr}(u, q)} \\ &\propto \tilde{Pr}(u, q, p) \\ &= Pr(\hat{u}, \hat{q}, \hat{p}) Pr(u|\hat{u}) Pr(q|\hat{q}) Pr(p|\hat{p}) \\ &\propto Pr(\hat{u}, \hat{q}, \hat{p}) Pr(p|\hat{p}) \quad (4) \end{aligned}$$

Thus according to Eq (4), the Web pages are ranked by $Pr(\hat{u}, \hat{q}, \hat{p}) Pr(p|\hat{p})$.

3. EXPERIMENTS AND CONCLUSIONS

We use 10 days' clickthrough data for experiments, the first 5 days' data is used for estimating the cluster structures and the rest 5 days' for testing. The search accuracy is evaluated using the $P@N$ measure, that is, the percentage of correct pages among the top N pages. We compared the result of CWS and Pearson collaborative filtering algorithm (CF) [1]. The results are given in Table 1. N is varied from 1 to 5. We can see the CWS approach leads to better search result compared with CF (around 8% improvement).

This shows the CF algorithm can not effectively exploit the clickthrough data as the data is three-way and highly sparse. However, the cluster structures can be discovered by the Cube-Clustering algorithm and the probabilistic relations among them can be utilized for improving Web search. This validates the effectiveness of our CWS approach: leveraging the clickthrough data for collaborative Web search.

4. REFERENCES

[1] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52. Morgan Kaufman, 1998.

[2] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *SIGKDD*, pages 89–98, 2003.

[3] B. Smyth, E. Balfe, O. Boydell, K. Bradley, P. Briggs, M. Coyle, and J. Freyne. A live-user evaluation of collaborative web search. In *IJCAI*, pages 1419–1424, 2005.