Using Probabilistic Latent Semantic Analysis for Personalized Web Search

Chenxi Lin[1], Gui-Rong Xue[1], Hua-Jun Zeng[2], and Yong Yu[1]

[1] Apex Data and Knowledge Management Lab,
Department of Computer Science and Engineering,
Shanghai JiaoTong University, Shanghai, 200030, P.R. China.
{linchenxi, grxue, yyu}@apex.sjtu.edu.cn
[2] Microsoft Research Asia,
49 Zhichun Road, Beijing, 100080, P.R. China
hjzeng@microsoft.com

Abstract. Web users use search engine to find useful information on the Internet. However current web search engines return answer to a query independent of specific user information need. Since web users with similar web behaviors tend to acquire similar information when they submit a same query, these unseen factors can be used to improve search result. In this paper we present an approach that mines these unseen factors from web logs to personalized web search. Our approach is based on probabilistic latent semantic analysis, a model based technique that is used to analyze co-occurrence data. Experimental results on real data collected by MSN search engine show the improvements over traditional web search.

1 Introduction

Search engines, such as Google, Yahoo! and MSN, have been the major tools to help users find useful information on the Internet. However current search technologies work in "one size fits all" fashion with results ordered by web site popularity rather than user interests. Since different users may have different information need, it is essential to personalize web search as well as better the service.

Many methods are proposed to study user's interests and build user profiles based on user's search history. These methods focus on analyzing content of the queries and web pages, but in some case there are no suitable descriptors such as keywords, topics, genres, etc. that can be used to accurately describe interests. According to [1], web users usually exhibit different types of behaviors depending on their information needs. Thus, web users with similar web behaviors tend to acquire similar information when they submit a same query to search engine. [2] conducted experiments to verify the effectiveness of several profile construction approaches and the results showed that the user profile constructed based on modified collaborative filtering achieved better retrieval accuracy. The collaborative filtering technique used in [2] is based on nearest neighbor regression or so-called memory-based techniques. These memory-

Y. Zhang et al. (Eds.): APWeb 2005, LNCS 3399, pp. 707 – 717, 2005.

based methods are simple and intuitive on a conceptual level while avoiding the complications of a potentially expensive model-building stage. However, there are a number of severe shortcomings as Hofmann point out in [3]: (i) The accuracy obtained by memory-based methods maybe suboptimal. (ii) Since no explicit statistical model is constructed, nothing is really learned form the available user profiles and very little general insight is gained. (iii) Memory-based methods do not scale well in terms of their resource requirements (memory and computing time). Especially, in web search tasks, the data set are always very large and the online response should be in a very short time. (iv) Actual user profiles have to be kept for prediction, potentially raising privacy issues.

Users' previous web behaviors and other personal information can be used to identify the users' information needs. In this paper, we indicate to analyze clickthrough data to personalize Web search. Clickthrough data is a kind of search log that could be collected by search engine implicitly without any participation of users. It logs for each query the query submitter and the web pages clicked by her. This process is different from those approaches based on user effort, such as providing relevance feedback or registering interest and demographic information. Through analysis of the clickthrough data, we could consider a single user's behavior characteristic and take similar users' interests into account, so as to identify the user's search intention and thus improve the search results.

To address the shortcomings of the memory-based methods mentioned above, we use a model-based technique called *Probabilistic Latent Semantic Analysis* (PLSA) [4]. A three-way learning and prediction model is proposed to deal with the triple relationship between users, queries and web pages on the usage data. The advantages of our method are as follows:

- The algorithm could compress the data into a compact model to automatically identify user search intention.
- The preference predictions could be computed in constant time so as to reduce online response time.
- The huge user profile data does not need to be kept.
- The experimental results also show that our proposed algorithm could achieve higher prediction accuracies on real data set collected by MSN search engine.

This paper is organized as follows. In Section 2, we introduce related work about personalized web search and probability latent semantic analysis. In Section 3, we present our model and show how to perform personalized web search based on the model. Our experiments and interpretation of the result is given in Section 4. Finally, we conclude this paper in Section 5.

2 Related Work

2.1 Personalized Web Search

[5] first proposed personalized PageRank and suggested to modify the global PageRank algorithm, which computes a universal notion of importance of a Web page. [6] used personalized PageRank scores to enable "topic sensitive" web searches.

Because no experiments based on a user's context, this approach actually cannot satisfy different information needs by different users.

Several approaches are proposed to construct user profiles by content of queries and web pages. [7] used ontology to model a user's interests, which are studied from user's browsed web pages. To distinguish long-term and short-term interests, [8] focused on using user's search history rather than browsing history to construct user profiles. Furthermore [9] mapped a query to a set of categories and [10] clustered words into a user interest hierarchy. All these methods are built on the fundamental assumption that users' interests or information needs can be formulated in term of intrinsic features of the information sought. In some case keywords, topics, genres and other descriptors are not able to describe information needs accurately.

[2] considered the unseen factors of the relationship between the web users behaviors and information needs and constructs user profiles through a memory-based collaborative filtering approach. Nevertheless it could not avoid the shortcoming listed in Sec.1.

2.2 Probabilistic Latent Semantic Analysis

Latent semantic analysis (LSA) [11] stems from linear algebra and performs a Singular Value Decomposition. It is mostly used in automatic indexing and information retrieval [12]. The key idea is to map high-dimensional count vectors to a lower dimensional representation in a so-called *latent semantic space*. Although LSA has proven to be a valuable analysis tool with a wide range of applications, its theoretical foundation remains to a large extent unsatisfactory and incomplete.

Hofmann presented a statistical view on LSA which leads a new model, *Probabilistic Latent Semantics Analysis* (PLSA) [4] [13], and provided a probabilistic approach for the discover of latent variables which is more flexible and has a more solid statistical foundation than the standard LSA. The basic of PLSA is a latent class statistical mixture model named aspect model. It is assumed that there exist a set of hidden factors underlying the co-occurrences among two sets of objects. That means the occurrences of two sets of objects are independent when the latent variables are given. PLSA uses *Expectation-Maximization* (EM) algorithm [14] to estimate the probability values which measure the relationship between the hidden factors and the two sets of objects.

Because of its flexibility, PLSA has been used successfully in a variety of application domain, including information retrieval [15], text learning [16] [17], and co-citation analysis [18] [19]. Furthermore, web usage mining can also be based on PLSA. [1] presented a framework to use PLSA for discovery and analysis of web navigational patterns, while it did not refer how to use PLSA to improve personalized web search. In our paper we present the approach and give an experimental evaluation.

3 Using PLSA to Predict

3.1 Prediction Problem Description

As we described in Sec.1, clickthrough data is collected by search engines without any participations of users. When a user submits a query to a search engine, the

search engine returns search results corresponding to the query. Based on the search results, the users may select the web pages which are related to their information need. Search engines could record the behaviors as the clickthrough data. The users, queries and web pages are collected as a co-occurrence triple in the web log. There are two kinds of data we should deal with differently. One is the queries the user has submitted several days ago, the search engine can easily to calculate which page is most frequent and rank it to the top one for the user. We experiment the real data from MSN search engine. If selected pages are not new pages, in other words they occurred in any search tasks in the past 20 days, more than 70% precision are reached of the top ones. The other is the queries that the user never submitted. Then the problem is: Given the clickthrough data, which page should be recommended to the user as the top results. Formally, given a set T that contains all previous (u,q,p) triples, a mapping function $f:U*Q\to P$ should be learned. The input of the function is any pair (u,q) where for any p', the triple $(u,q,p')\notin T$. The output is the page which is predicted the most possible needed page by the user. More generally, the top k pages will be interested as the recommendation problem.

3.2 Model Specification

The starting point for PLSA is a latent class statistical mixture model which has been called aspect model. This model is a latent semantic class model for co-occurrence data which associates an unobserved class variable $z\in Z\{z_1,z_2,\cdots z_k\}$ with each observation. These unobserved classes stand by the hidden factors underlying the co-occurrence among the observed sets of objects. Therefore this model well capture unseen factor that lead to the fact that web users exhibit different types of behavior depending on their information needs. At the same time it well characterizes the hidden semantic relationship among users, queries as well as users, queries and web pages. Therefore, in our web search scenario, an observation is a triple (u,q,p) corresponding an event that a user u submits a query q to a search engine, and selects a page p from the results. In the context of web search, users $u\in U\{u_1,u_2,\cdots u_n\}$, queries $q\in Q\{q_1,q_2,\cdots q_m\}$, together with web pages $p\in P\{p_1,p_2,\cdots p_l\}$, form triple relationship (u,q,p). The relationships are associated with the latent variables $z\in Z\{z_1,z_2,\cdots z_k\}$. The mixture model depends on a conditional independence assumption, namely each set of observed objects are independent conditioned on the state of the associated latent variable. Conceptually, the latent variables are search intentions. According to the assumption, users, queries and web pages are independent when given search intentions. It means that a user u and a query q determine a latent search intention z, and latent variables in turn "generated" web page p. Fig.1 depicts the model as a Bayesian network.

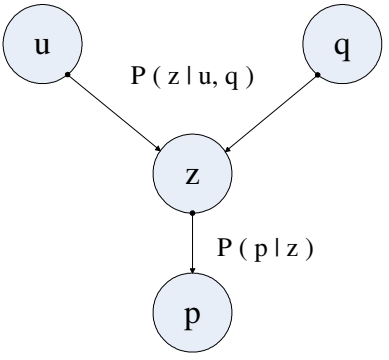

Fig. 1. Graphical representation of the three-way aspect model

Therefore, the joint our model is defined as follows:

$$P(u,q,p) = P(u,q)P(p \mid u,q)\,, \tag{1}$$

$$P(p \mid u,q) = \sum_{z \in Z} P(p \mid z)P(z \mid u,q)\,. \tag{2}$$

Reversing the arc from users and queries to latent search intentions, we can get an equivalent symmetric specification:

$$P(u,q,p) = \sum_{z \in Z} P(z)P(u \mid z)P(q \mid z)P(p \mid z)\,. \tag{3}$$

3.3 Model Fitting with the EM Algorithm

PLSA uses the *Expectation-Maximization* (EM) [14] algorithm to estimate the probability value which measure the relationships between the hidden factors and the sets of objects. According to Formula (3), in order to explain a set of observation (u,q,p), we need to estimate the parameters $P(z)$, $P(u \mid z)$, $P(q \mid z)$ and $P(p \mid z)$, while maximizing following likelihood. The algorithm alternates two steps:

- An expectation (E) step, where posterior probabilities are computed for the latent variable, based on the current estimates of the parameters;
- A maximization (M) step, where parameters are re-estimated to maximize the expectation of the complete data likelihood.

Let $n(u,q,p)$ be the number of times user u selects page p of query q. Given training data, the log likelihood L of the data is:

$$L = \sum_{u,q,p} n(u,q,p) \log P(u,q,p)\,. \tag{4}$$

In the E-step, we compute:

$$P(z \mid u,q,p) = \frac{P(z)P(u \mid z)P(q \mid z)P(p \mid z)}{\sum_{z' \in Z} P(z')P(u \mid z')P(q \mid z')P(p \mid z')}, \tag{5}$$

In the M-step, the formulae are:

$$\begin{aligned} P(z) &= \frac{\sum_{u,q,p} n(u,q,p)P(z \mid u,q,p)}{\sum_{z' \in Z}\sum_{u,q,p} n(u,q,p)P(z' \mid u,q,p)} \\ &= \frac{\sum_{u,q,p} n(u,q,p)P(z \mid u,q,p)}{\sum_{u,q,p} n(u,q,p)} \end{aligned} \tag{6}$$

$$P(u \mid z) = \frac{\sum_{q,p} n(u,q,p)P(z \mid u,q,p)}{\sum_{u',q,p} n(u',q,p)P(z \mid u',q,p)} \tag{7}$$

$$P(q \mid z) = \frac{\sum_{u,p} n(u,q,p)P(z \mid u,q,p)}{\sum_{u,q',p} n(u,q',p)P(z \mid u,q',p)} \tag{8}$$

$$P(p \mid z) = \frac{\sum_{u,q} n(u,q,p)P(z \mid u,q,p)}{\sum_{u,q,p'} n(u,q,p')P(z \mid u,q,p')} \tag{9}$$

Iterating these two steps monotonically increases the log-likelihood of the observed data until a local maximum optimal solution is reached.

3.4 Prediction in Practice

Theoretically in our model, prediction is provided to users according to:

$$P(p \mid u,q) = \frac{\sum_{z \in Z} n(u,q,p)P(z \mid u,q,p)}{\sum_{p'}\sum_{z \in Z} n(u,q,p')P(z \mid u,q,p')}. \tag{10}$$

In practice, we cluster the web users, queries and web pages before using PLSA in order to: (1) overcoming the overfitting problem with sparse data, (2) reduce the

memory and offline time cost with large data set. We make assumption that each user is belong to exactly one group of users, so as each query and web page. Hence we have mapping functions $c(u) \in C = \{c_1, c_2, \cdots c_h\}$, $d(q) \in D = \{d_1, d_2, \cdots d_i\}$ and $e(p) \in E = \{e_1, e_2, \cdots e_j\}$. Then the cluster algorithm partitions U into h groups, Q into i groups and P into j groups. The algorithm also give the probability values $P(u \mid c(u))$, $P(q \mid d(q))$ and $P(p \mid e(p))$. After the processing, we use PLSA to calculate the probability values which measure the relationships between C, D, E and Z, so in practice we predict the probability for a given (u, q, p) as follows:

$$P(p \mid u, q) = P(e(p) \mid c(u), d(q))P(p \mid e(p)). \tag{11}$$

4 Experimental Evaluation

4.1 Dataset

In our experiments, we use web log data collected by MSN search engine in December 2003. We select those users who use MSN search engine more 25 days in the month in order that the data is not too sparse. Then we randomly select about 100,000 queries submitted to the search engine by the users and 200,000 web pages they selected to browse from the results given by the search engine. In order to evaluate different scenarios, we design two data sets as following:

First, since our approach could show higher performance on the situation that the pairs of user and query do not occur in the training set, we divide the data set by random selecting pairs of user and query. In our experiment, 85% pairs of user and query are selected as training set and the left as testing data, so we get 290,000 data records in training set and 78,000 data records in testing set. This data set is referred as the first data set in our experiment.

Second, we divide the data according to the time series in the log. We use all the data in the first 20 days as training data, while the testing data is from those of later 5 days. If some queries or web pages are in the testing data but not in the training data, without content-based techniques they are impossible to be predicted. So we remove these data in the testing set. Finally we get a training set with 340,000 data records and a testing set with 6,000 data records. We refer to this data set as the second data set in our experiment.

4.2 Evaluation Metric

From the view of the user, the effectiveness of our method is evaluated by the precision of the predictions. Given a triple (u, q, p) in the test set, we first get the predicted list P based on u and q. Then we sort all the $p' \in P$ according to the probability $P(p' \mid u, q)$ in the descending order, and get the rank of p in the sorted

list. For each rank $r > 0$, we calculate the number of triples that exactly rank the r th as $N(r)$. Let $M(r) = \sum_{1 \le i \le r} N(i)$, and $S(r) = M(r)/G$ where G stands by the number of triple in the whole test set. Thus $S(r)$ stands for the precision of the method when predicting the top r web pages.

4.3 Baseline Method

We have implemented a baseline method to calibrate the achieved results. The method is based on cluster technique. We use the relationship between the users, queries and web pages to cluster the users and the queries. Then we calculate the times of each page is occurred with every user group and query group in the training set and give the prediction. Let $n(c(u), d(q), p) = \sum_{u,q} n(u, q, p)$ for any $(u, q, p) \in T$. So for a given (u, q, p), the probability is predicted as follows:

$$P(p \mid u, q) = \frac{n(c(u), d(q), p)}{\sum_{p'} n(c(u), d(q), p')} \tag{12}$$

4.4 Experimental Results

As we know, the number of cluster is difficult to decide. In our experiment, we tried several times to tune the parameters in order to get higher performance of clustering. We finally cluster the users into 1000 groups, the queries into 1500 groups and the web pages 2000 groups. Meanwhile, that the EM algorithm will get a local optimization after 30-60 iterations.

Fig. 2.

Fig. 3.

Fig.2 shows the results on the first data set. As shown in the figure, our method gets a better performance on the queries that the user never submitted. The top 5^{th} precision has been increased to 22.23% over the cluster-based method. The result shows that our model has strong ability of prediction. Without any direct information about the user and the submitted query, the results of the prediction based on the statistics are not very good. As illustrated in Fig.3, our method also outperforms the baseline method on the second data set.

4.5 Computational Complexity

In the web search scenario, because of the amount of data is always large, the computational complexity is one of the crucial factors for a successful algorithm. One has to distinguish between the offline and online computational complexity. The former accounts for computations that can be performed before hand, that is, before actual predictions for specific users have to be made. In contrast, the latter deals with those computations that can only be performed in real-time during the interaction with a specific user. Therefore the online computational complexity is more important here. PLSA algorithm has an online computational complexity of $O(|Z|)$. The detail complexity analysis of PLSA could be found in [20].

5 Conclusions

In this paper, we present an approach to perform better personalized web search based on PLSA. We consider the latent semantic relationship between users, queries and web pages by a three-way aspect model and use the proposed algorithm to deal with the sparsity problem. Meanwhile, the model could character the users' search

intention. An effective algorithm is proposed to learn the model and compute the preference prediction. The results on the real clickthrough data show that our proposed algorithm could achieve higher prediction accuracies than the baseline work. In the future, we consider integrating the content and the link information into the algorithm and doing the better prediction.

References

1. X. Jin and Y. Zhou and B. Mobasher: Web Usage Mining based on Probabilistic Latent Semantic Analysis In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'04), Seattle (2004)
2. Sugiyama, K., Hatano, K., Yoshikawa, M.: Adaptive web search based on user profile constructed without any effort from users. In: Proceedings of the 13th international conference on World Wide Web, ACM Press (2004) 675-684
3. Hofmann, T.: Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press (2003) 259-266
4. Hofmann, T.: Probabilistic Latent Semantic Analysis. In: Proceedings of Uncertainty in Artificial Intelligence, UAI'99, Stockholm (1999)
5. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
6. T. H. Haveliwali.: Topic-Sensitive PangeRank. In: Proceedings of the 11th International World Wide Web Conference (WWW2002), pages 517-526, 2002.
7. Pretschner A., Gauch S.: Ontology based personalized search. In: ICTAI. (1999) 391-398
8. Micro Speretta, Susan Gauch: Personalizing Search Based on User Search Histories. In: Thirteenth International Conference on Information and Knowledge Management (CIKM 2004).
9. Liu, F., Yu, C., Meng, W.: Personalized web search by mapping user queries to categories. In: Proceedings of the eleventh international conference on Information and knowledge management, ACM Press (2002) 558-565
10. Kim H. R., Chan P. K.: Learning implicit user interest hierarchy for context in personalization. In: Proceedings of the 8th international conference on Intelligent User Interfaces, ACM Press (2003) 101-108
11. Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R.: (indexing by latent semantic analysis)
12. Berry, M., Dumais, S., G.OBien: Using linear algebra for intelligent information retrieval. (1995)
13. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning 42 (2001) 177-196
14. Dempster, A.P., Laird, N.M., Rubin, D.B.: (Maximum likelihood from incomplete data via the EM algorithm)
15. Hofmann, T.: Probabilistic Latent Semantic Indexing. In: Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval, Berkeley, California (1999) 50-57
16. T. Brants, F. Chen, I.T.: topic-based document segmentation with probabilistic latent semantic analysis. In: Proceedings of Eleventh International Conference on Information and Knowledge Management. (2002)

17. E. Gaussier, C. Goutte, K.P.F.C.: a hierarchical model for clustering and categorising documents. In: 24th BCS-IRSG European Colloquium on IR Research.(2002)
18. Cohn, D., Chang, H.: Learning to probabilistically identify authoritative documents. In: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc. (2000) 167-174
19. Cohn, D., Hofmann, T.: The missing link - A probabilistic model of document content and hypertext connectivity. In Leen, T.K., Dietterich, T.G., Tresp, V., eds.: Advances in Neural Information Processing Systems 13, MIT Press (2001) 430-436
20. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. 22 (2004) 89-115