Personalizing Google's Search Results using Naïve Bayesian Probabilistic Model

SRINIVASA K G, SRINIVAS KUMAR M, VINAY T C
Data Mining Laboratory, M S Ramaiah Institute of Technology,
Bangalore *560054*, India
kgsrinivas@msrit.edu, {skumar.msrit, vinaytc}@*gmail.com*

VENUGOPAL K R
Department of Computer Science and Engineering
University Visvesvaraya College of Engineering, Bangalore University
Bangalore *560001*, India
venugopalkr@gmail.com

L M PATNAIK
Microprocessor Applications Laboratory
Indian Institute of Science, Bangalore, India
lalit@micro.iisc.ernet.in

(Paper received on August 24, 2006, accepted on September 27, 2006)

Abstract. Personalization is the process of presenting the right information to the right user at the right moment, by storing browsing history about the user and analysing that information. Personalization methods enables the site to target advertising, promote products, personalize news feeds, recommend documents, make appropriate advice, and target e-mail. In this paper we introduce a Naive Bayesian probabilistic model, which classifies the sites into different categories. The user profile is built dynamically on the recorded interests of the user, which are nothing, categories of the site in which user browses. The algorithms are tested on varying keywords and the results obtained were compared with the Google's page rank system.

1. Introduction

Internet is one development, which fuelled the growth of IT industry to a large extent. It has single handedly changed the way we communicate, collaborate and cooperate with each other. It has been responsible for breaking the barriers of time and geographical limitations and helped set a new business order. From being an information repository it has today grown to be a strategic tool in business. Internet, by its very nature, does not support systematic storage of information. The units of storing information in Internet servers are websites that are uniquely identified by their URL. The problem that everyone had to face aftermath the explosion of information in Internet was to get relevant information on time. Remembering URLs proved cumbersome and difficulty in mining for correct information was proving as hindrance for the further growth of Internet.

Special Issue in Neural Networks and Associative Memories
Research in Computing Science 21, 2006, pp. 87-96

Search Engines proved to the right medicine for this ailing problem of the industry. But search engines faced a typical problem since they relied on keywords for conducting search. The present day search engines have failed to understand the specific requirements of the user [17]. Users experience increasing difficulty finding documents relevant to their interests as search engines throw same result to everyone irrespective of user expectation. Google's Personalized Search drives to make search experience more relevant to the user. Using Personalized Search, user can get the results most relevant to user, based on what user has searched in the past, View and manage user's past searches, including the web pages, images, Froogle results which he has clicked on and create bookmarks. Personalized Search orders your search results based on user's past searches, as well as the search results and news headlines user has clicked on. User can view all these items in your Search History and remove any items you'd like. Early on, user may not notice a huge impact on his search results, but as he builds up your search history, his personalized search results will continue to improve.

2. Related Work

Conceptual search can be done by explicitly providing the meaning of the content in a Web page. So one way to address this problem is by having the authors of the content explicitly specify the meaning associated with a page using a Knowledge Representation Language. One of the Knowledge Representation Languages is *Ontobroker* and is discussed in [1]. Domain-specific Web search engines are effective tools for reducing the difficulty experienced when acquiring information from the Web. Building of a domain-specific search engine simply by adding domain-specific keyword, called "keyword spices," to the user's input query and forwarding it to a general-purpose Web search engine is presented in [2]. A tool that assists an end-user or an application to search and process information from the Web pages automatically by separating the primary content sections from the other content sections is presented in [3]. Ontology is a specification of a conceptualisation. Sophisticated ontologies incorporate logical relationships and membership rules. However, concept hierarchies can also be used as simple ontologies. Use of Yahoo! categories as a concept hierarchy and classifying documents into it using an *n-gram* classifier is discussed in [4].

he user profiles are a representation of the user's interests, such as Wisconsin Adaptive Web Assistant (WAWA). Building profiles non-invasively by observing user's visit to Web pages over a period of time is addressed in [5]. They generally use the profile to suggest related Web pages to the users as they browse. The study of personalized recommendation in a B2C portal to build improved algorithm, EI-B&B-MDL, for learning Bayesian networks effectively and efficiently is proposed in [6]. The algorithm reduces the number of independence tests and database passes while effectively restricting the search space. A personalized recommendation agent, fuzzy cognitive agent, to give personalized suggestions based on the current user's preferences, general user's common preferences, and the expert's knowledge is given in [7]. Fuzzy cognitive agents are able to represent knowledge via extended fuzzy cognitive maps, to learn user's common preferences from most recent cases and to help customers to make in-

ference/decisions through numeric computation instead of symbolic and logic deduction. An algorithm has been presented in [8] which can generate and display helpful links while users navigate a site and hence increasing the Web site's usability and help Web designers and the user achieve their goals.

In the literature data mining methods are very much exploited to build the customer profiles [9]. The 1:1 Pro system constructs profiles based on customers' transactional histories. The system uses data mining techniques to discover a set of rules describing customers' behaviour and supports human experts in validating the rules. The vision of ontology learning including a number of complementary disciplines that feed on different types of unstructured, semi structured and fully structured data to support semiautomatic, cooperative ontology engineering is presented in [10]. A new method for tracking the dynamics of user interests from a minimal number of relevance judgments is given in [11].

3. Proposed System

Problem Formulation: Search engines are affected by problems such as ambiguity [16] and results ordered by Web site's popularity rather than user interests. Natural language queries are inherently ambiguous. For example, consider a user query "Pluto". Due to ambiguity in the query terms, the results obtained are either related to astronomy or cartoon. Most users enter quires in just a word or two without providing enough information. These short queries are often ambiguous, providing little information to the search engine. A user profile that represents the interests of a specific user can be used to supplement queries, narrowing down the number of topics considered when retrieving the results. If system had a prior knowledge that user has a strong interest in Astronomy and little in others like cartoon, the Astronomy related results of Pluto could be presented to the user first and then cartoons preferentially. Therefore, user profile creation is important for personalization [18].

Our approach of building user profiles is based on the user's interactions with a particular search engine. For this purpose, GoogleWrapper: a wrapper around the Google search engine [13] is implemented, it logs the queries, search results, and clicks on a per user basis. The snippets, which are obtained using wrapper serves as the input for the algorithm to identify the category of that site. This information is then used to create user profiles and these profiles are used in a controlled study to determine their effectiveness for providing personalized search results. Information about the user can be collected either explicitly or implicitly. Explicit construction is the way to fill up the forms at the time of first login giving preferences or ratings. Implicit construction is observing user behaviours such as the categories of the URLs visited, time spent and number of inner clicks made in a particular site. Explicit construction of user profiles has several drawbacks. The user may provide inconsistent or incorrect information and the profile built is static whereas the user's interests may change over time, and the construction of the profile places a burden on the user. User browsing histories i.e., the categories the user has browsed so far are the most frequently used source of information about user interests. This information is used to create user profiles. Classifying the collected Web pages with respect to a category in which the user is interested cre-

ates the user profile. The fact that a user has visited a page and spent some time is an indication of user interest in that page's content i.e., in that category.

Problem Definition: Consider a Search Engine E, which accepts queries in the form of keywords and returns a list of near relevant web pages. The queries are of the form k_1 [**(op)** k_i] for i= 2 to n

Where k_i is the i^{th} keyword supplied to E

(op) is a Boolean operator like OR, AND...

n is the number of keywords supplied to E

[...] indicated the parameters are optional.

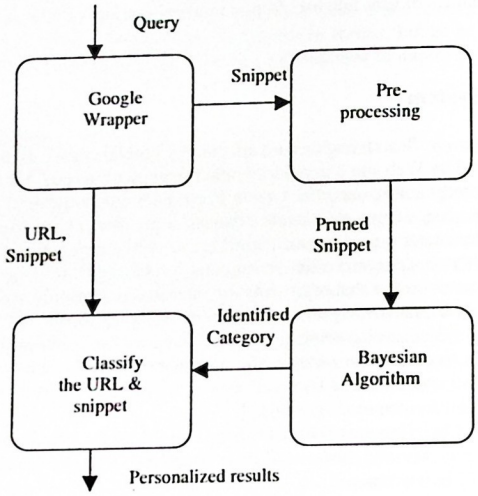

Fig. 1. . System Architecture.

When search engine E returns its results, user clicks on some site and then the corresponding snippet S is used as an input to the algorithm to identify the category of the site. This category represents the user's interest and is added to the user profile. Each category has an associated weight and the weight decides the choice of the category. Each time a new category is created, it is assigned an initial weight $C_{Weight} = 0$. Hence all categories start with the same weight or same opportunity in the user profile. The weights for a category are decided dynamically to reflect the user's interest in certain domain. The weight of the category also depends on the amount of time the user spends in a particular site and therefore in a particular category. A higher weight indicates a greater interest in that domain. When the weight for a category crosses a pre-defined

lower threshold, it is deleted from the user profile. This reflects the fact that the user has lost interest in that domain. The overall system architecture is given in Figure 1.

Table 1: Naïve Bayesian Probabilistic Model

Algorithm: Naïve Bayesian Probabilistic Model

STEP 1: The data sample is Snippet $S = (w_1, w_2, \ldots, w_n)$, where, w_i for $i=1 \ldots n$ are the words in the snippet after removing stop words.

STEP 2: Suppose that there are m categories $C_1, C_2, \ldots, C_m$. Given an unknown Snippet S (i.e., with no label), the classifier will predict that S belongs to the category having highest probability.

By Bayes Theorem $P(C_i/S) = P(S/C_i) * P(C_i) / P(S)$

STEP 3: $P(S)$ is constant for all categories, therefore only $P(S/C_i) * P(C_i)$ need to be maximized.

If the category prior probabilities are not known, then it is assumed that categories are equally likely i.e., $P(C_1) = P(C_2) = \ldots = P(C_m)$. Therefore maximize only $P(S/C_i)$.

Otherwise, maximize $P(S/C_i) * P(C_i)$. This even identifies the user's interest in a particular category.

Category prior probability is estimated as

$P(C_i)$ = (Number of times user visited category C_i / Total number of times visits)

STEP 4: Now, Naïve assumption of *class conditional independence* is made.

i.e., $P(S/C_i) = \prod_{k=1}^{n} P(S_k/C_i)$

Thus $P(w_1/C_i), P(w_2/C_i), \ldots, P(w_n/C_i)$ can be estimated from training samples, where $P(S_k/C_i)$ = (Number of words in C_i having value S_k / number of words in C_i).

STEP 5: In order to classify an unknown snippet S, $P(S/C_i) * P(C_i)$ is evaluated for each category C_i. Sample snippet S is then assigned to the category C_i if and only if $P(S/C_i) * P(C_i) > P(S/C_j) * P(C_j)$ for $1 \le j \le m, j \ne i$.

In other words, it is assigned to the category C_i for which $P(S/C_i) * P(C_i)$ is maximum.

Observation: Our version of Naïve Bayesian Probabilistic model (Table 1) differs in computation of category prior probability $P(C_i)$ (Step 3). Traditional model esti-

mates category prior probability by $P(C_i)$ = (number of training samples in category C_i / total number of training samples), instead we make use of user profile for computation of $P(C_i)$. Also we classify the sites into the categories found in the *user profile* (Table 1) and not rest, since user would be most interested in those categories found in his profile. This reduces the cost of the model to a large extent because we are not classifying the sites into all the available categories. It only tries to classify into other categories when user is no more interested in these categories, which seldom happens. In that case $P(C_i)$ is not considered, i.e., only $P(S/C_i)$ is maximised. Three cases exist:

Case 1: YES, the results retrieved are as needed by the user.
- Increment the weight of that particular category, $C_{Weight} = C_{Weight} + 1$.

Case 2: NO, the results retrieved are not as per user's need.
- Classify the remaining results and display.
- Append the category of the site on which user clicks and assign weight 1.

Case 3: YES, the retrieved results are correct but user wants other results also.
- Increment the weight of that particular category, $C_{Weight} = C_{Weight} + 1$.
- Classify the remaining results and display.
- Append the category of the site on which user clicks and assign weight 1.

Table 2: User Profile.

Category (C_i)	Weight (No. of visits)	$P(C_i)$
Science > Astronomy	2	2/10 = 0.2
Science > Computer	5	5/10 = 0.5
Science > Biology	3	3/10 = 0.3

Example: Lets assume that a user has logged in and browsed 10 sites, 5 of which fall under Science > Computer, 3 of which come under Science > Biology and remaining 2 Science > Astronomy as shown in his profile (Table 2). Now if he submits a query say 'Genetics' (Ambiguous because genetics can be related to both genetic algorithms and genetics of biology), all the possible results of genetics will be retrieved initially. Now we apply Naïve Bayesian algorithm to classify these results into categories found in the profile.

Let's say there is an unknown snippet, which is to be classified. It has words w_i = {genetics, powerful} after getting pruned. Number of training samples belonging to category Science > Astronomy (SA) = 5, neither of the words w_i are found in training set of SA so not this category. Number of training samples belonging to category Science > Biology (SB) = 10. This has only one word, genetics. P(genetics/SB) * P(SB) = (1/10) * 0.3 = 0.03. Number of training samples belonging to category Science > Com-

puter (SC) = 8. This has only two words. genetics and algorithm. P(genetics/SC) * P(SC) = (1/8) * 0.5 = 0.0625. P(genetics/SC) * P(SC) > P(genetics/SB) * P(SB) therefore the snippet S belongs to SC i.e., (Science>Computer) category.

4. Performance Analysis

Experiment 1: Average rank of the Google search results remain the same irrespective of user interest, where as in our proposed system the ranking differs as the users and their interest changes. When user interest is 0 i.e., the system doesn't know anything about the user, the rank is same as the Google rank say 5. As user interest approaches 1, i.e., the system is learning about the user gradually, then its ranking improves to 3 as shown in Figure 2. Google pagerank works on the basis that if a website abc.com has been linked from a website xyz.com, abc.com must have some good content and therefore Google will count the link from xyz.com as a vote for abc.com.

Experiment 2: Precision is the percentage of retrieved documents that are in fact relevant to the query (i.e., "correct" responses). Precision of the system depends on the number of times the users uses the system and the sequence in which he proceeds. The precision is calculated as follows.

$$\text{Precision} = |\{\text{Number of Relevant documents}\} \cap \{\text{Number of Retrieved documents}\}| / |\{\text{Number of Retrieved documents}\}| \quad (1)$$

Fig. 2. the Precision of Google's search results is compared with our results with respect to varying logins. Precision of Google search doesn't vary much, where as the precision of our search result grows as the number times the user logins.

An illustrative example is given below to explain the variation in precision. Here we assume that user submits just one query per login.

Let's say the system retrieves 100 results every time on the query Pluto. For the first time 20 relevant documents (user interested in cartoon) are found and user clicks on cartoon's site and hence the precision is 0.2. For the second time when the user logs in, he gives a query Pluto, only cartoon sites are retrieved as user showed interest in cartoons last time. But now user doesn't want cartoons site, he wants planets. Therefore no relevant sites are found for planets sites and hence the precision is 0. For the third time when the user logs in, he again gives a query Pluto, the system retrieves both cartoons and planets because both categories have equal weight, and user is interested in any one of them, say user shows interest in cartoons, so cartoon site gets more weight and hence precision is 0.5. For the forth time when the user logs in and query is Pluto again, the system retrieves 75 cartoon results and 25 planet results based on weight. Users wants cartoon and therefore precision is 0.75. In the next login with the same

query Pluto the system retrieves 80 results of cartoon and 20 of planets and user wants cartoon sites and hence precision is 0.8.

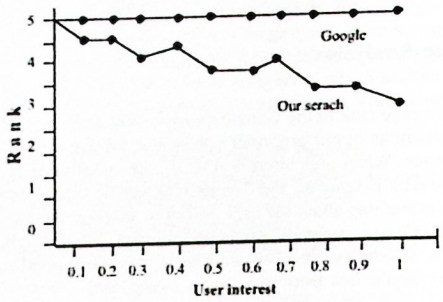

Fig. 3. Variation in ranking of Google results and Our results as the user interest is varied.

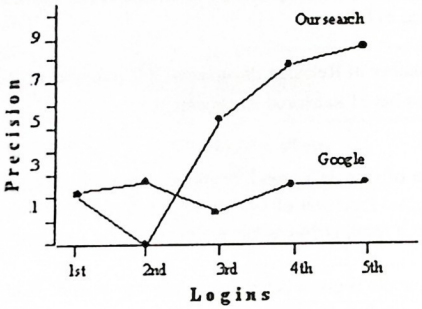

Fig. 4. : Variation in precision of Google results and our search results as the number of times user logs in.

Experiment 3: Recall is the percentage of documents that are relevant to the query and are, in fact, retrieved. It is formally defined as,

Recall = | {Number of Relevant documents} ∩ {Number of Retrieved Documents} | / | {Number of Relevant documents} |. (2)

Figure 5 shows a variation in recall of Google results and our search results as the user interest is varied. Recall of Google search results doesn't change much as it doesn't

depend on user's interests whereas recall in our case varies as user interest change and it approaches one as user interest approaches one.

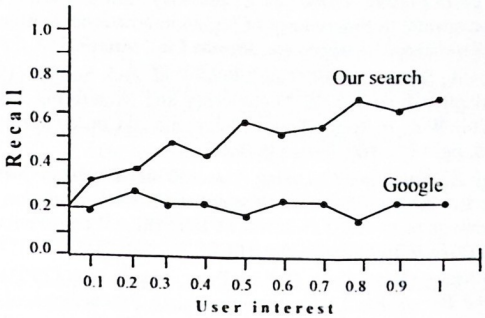

Fig. 5. Variation in recall of as the user interest is varied.

5. Conclusions

In this paper, we have proposed a new version of Naive Bayesian probabilistic model, which classifies the sites into different categories. The algorithm is tested on Google's results and Google PageRank to obtain the top 10 results and the results obtained are satisfactory in nature. Table 4 shows the Comparison of Google search results with our search results for the query Pluto. Table 5 Comparison of Google search results with our search results for the query Genetic.

Acknowledgements

This work is partially supported by AICTE, New Delhi, under AICTE Career Award for Young Teachers (AICTE File No. F.No.1-51/FD/CA/(9)/2005-06) to Mr. Srinivasa K G, Lecturer, Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, MSR Nagar, Bangalore – 560054, Karnataka, India.

References

[1] Stefan Decker, Michael Erdmann, Dieter Fensel, Rudi Studer, (1998). Ontobroker: Ontology based Access to Distributed and Semi-Structured Information, In Proceedings of W3C Query Language Workshop QL'98.

[2] Satochi Oyama, Takashi Kokubo, and Toru Ishida, (2004). Domain-Specific Web Search with Keyword Spices, 1041-4347/04/ 2004 IEEE.

[3] Sandip Debnath, Prasenjit Mitra, Nirmal Pal, and C. Lee Giles (2005). Automatic Identification of Informative Section of Web Pages, 1041-4347/05 IEEE.

[4] Labrou, Finin (1999). Yahoo! as an ontology: using Yahoo! categories to describe documents. In Proceedings of Eighth international conference on Information and Knowledge Management, Kansas City, Missouri.

[5] Jude Shavlik, Susan Calcari, Tina Eliassi-Rad, Jack Solock (1999). An Instructable, Adaptive Interface for Discovering and Monitoring Information on the World Wide Web. In Proceedings of International Conference on Intelligent User Interfaces, pp. 157 - 160, Redondo Beach, CA.

[6] Junzhong Ji, Chunnian Liu, Jing Yan (2004). Bayesian Networks Structure Learning and Its Application to Personalized Recommendation in a B2C Portal, In Proceedings of the IEEE/WIC/ACM International conference on Web Intelligence (WI'04), 0-7695-2100-2/04 IEEE.

[7] Chunyan Miao, Qiang Yana, Haijing Fang, Angela Goh (2002). Fuzy Cognitive Agents for Personalized Recommendation, In Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE.02) 0-7695-1766-8/02 IEEE.

[8] Mamata Jenamani, Pratap K.J. Mojapatra, and Sujoy Ghose (2002). Online Customized Index Synthesis in Commercial Web Sites, 1094-7167/02 IEEE Intelligent Systems, pp 20-26.

[9] Gediminas, Adomavicius, Alexander, Tuzhilin (2001). Using Data Mining Methods to Build Customer Profiles, 0018-9162/01 IEEE Computer, 74-82.

[10] Alexander Maedche and Steffen Staab (2001). Ontology Learning for the Semantic Web, 1094-7167/01 IEEE Intelligent Systems.

[11] Dwi H. Widyantoro & Thomas R. Ioerger, John Yen (2003). Tracking Changes in User Interests with a Few Relevance Judgments, CIKM'03, November 3-8, 2003, New Orleans, Louisiana, USA, ACM 1-58113-723-0/03/0011.

[12] Open Directory Project http:/dmoz.org

[13] http:/www.google.com/api

[14] Yannis Labrou, Tim Finin (1999). Yahoo! As An Ontology – Using Yahoo! Categories To Describe Documents, In Proceedings of the 8th International Conference On Information Knowledge Management (CIKM), 1999, pp. 180-187.

[15] Yahoo! http://www.yahoo.com

[16] Robert Krovetz and Bruce W. Croft (1992). Lexical Ambiguity and Information Retrieval, In Proceedings of ACM Transactions on Information Systems, 10(2), April 1992, pp. 115-141.

[17] P Deepa Shenoy, K G Srinivasa, A O Thomas, Venugopal K R & L M Patnaik (2004). Mining Top-k Ranked Webpages using Simulated Annealing & Genetic Algorithm, AACC 2004, pp. 137-144.

[18] Srinivasa K G, P Deepa Shenoy, Venugopal K R & L M Patnaik (2005). A Hybrid System for Web Search Personalization using Query Refinement, Recommender Systems and User Profiles", In Proceedings of 13th International conference on Advanced Computing (ADCOM 2005), Coimbatore, Dec. 14-17, 2005.