A User Preference Based Search Engine

[1]Dondeti Swedhan, [2]L.N.B. Srinivas
[1]M-Tech, [2]M-Tech
[1]Department of Information Technology,
[1]SRM University Kattankulathur, Chennai, India

Abstract **- In this paper, we describe the design and initial implementation of a user preference search engine (UPSE) that captures the users' preference in the form of concepts by mining their clickthrough data. UPSE classifies these concepts into content concepts and location concepts. In addition, users' locations (positioned by GPS) are used to supplement the location concepts in UPSE. This search engines provide a flexible interface to the web that allows users to constrain and order search results in an intuitive manner. In this we present an approach to automatically optimizing the retrieval quality of searching using clickthrough data. A good information retrieval system should present relevant document high in the ranking. We develop a new preference mining technique called SpyNB, based on the practical assumption that the search results clicked on by the user reflects the user's preferences, but it does not draw any conclusions about the results that the user did not click on. We also show that the efficiency of SpyNB is comparable to existing simple preference mining algorithms.**

Index Terms **- Clickthrough data, Concept, Location search, Preference, User profiling**

I. INTRODUCTION

The World-Wide Web has reached a size where it is becoming increasingly challenging to satisfy certain information. While search engines are still able to a reasonable subset of the (surface) for may be buried under hundreds of thousands of less interesting results. Thus, search engine users are in danger of drowning in information. A natural work is to add advanced features to search engines that allow users to express preferences in an intuitive manner; a major problem in mobile search is that the interactions between the users and search engines are limited by the small form factors of the mobile devices. Mobile users tend to submit shorter hence, more queries compared to their web search counterparts. To return highly relevant results to the users, search engines must be able to profile the user's interests and preference the search results according to the users' profiles. A practical work to capturing a user interests for preference is to analyze the user's clickthrough data.

We developed a user search engine personalization method based on users' concept preferences and showed that it is more effective than methods that are based on page preferences. Observing the need for different types of concepts, we present in this paper a preference search engine (UPSE) which represents different types of concepts in different anthologies. In particular, recognizing the importance of location information in mobile search, we separate concepts into location and content. For example, a user who is planning to visit Hyderabad may issue the query "hotel," and click on the search results about hotels in Hyderabad. From the clickthroughs of the query "hotel," UPSE can learn the user's content preference (e.g., "room rate" and "facilities") and location preferences ("Hyderabad"). Accordingly, UPSE will favor results that are concerned with hotel information in Hyderabad for future queries on hotel.

The introduction of location preferences offers UPSE an additional dimension for capturing a user's interest and an opportunity to enhance search quality for users. To incorporate context information revealed by user mobility, we also take into account the visited physical locations of users in the UPSE. Since this information can be conveniently obtained by GPS devices, it is hence referred to as GPS locations. GPS locations play an important role in mobile web search. For example, if the user, who is searching for hotel information, is currently located in "Bachupally, Kukatpally," his/her position can be used to preference the search results to favor information about nearby hotels. We can see that the GPS locations (i.e., "Bachupally, Kukutpally") help reinforcing the user's location preferences (i.e., "Hyderabad") derived from a user's search activities to provide the most relevant results. Our proposed framework is capable of combining a user's GPS locations and location preferences into the process. To the best of our knowledge, our paper is the first to propose a preference framework that utilizes a user's content preferences and location preferences as well as the GPS locations in personalizing search results.

Clickthrough for query "Hotels"

Table 1

Doc	Search Results	Ci	Li
D1	Hotels	Room rate	International
D2	Indianhotels.com	Reservation, Room rate	Hyderabad
D3	Americanhotels.com	Room rate	California
D4	Booking.com	Online	India
D5	Park hotel	Room rate	USA

The main contributions of this paper are as follows:

- This paper studies the unique characteristics of content and location concepts.
- The proposed preference mobile search engine is an innovative approach for preference web search results.
- UPSE incorporates a user's physical locations in the preference process.

We propose a new and realistic system design for UPSE. Our design adopts the server-client model in which user queries are forwarded to a UPMSE server for processing the training and re-ranking quickly.

II. SYSTEM DESIGN

With the collection of the data and identifying the aspects from them and then building the ranking framework based on SpyNB (Spy Naive Bayes) algorithm and RSVM (Ranking Support Vector Machine) algorithm for optimizing the ranking function for the user.

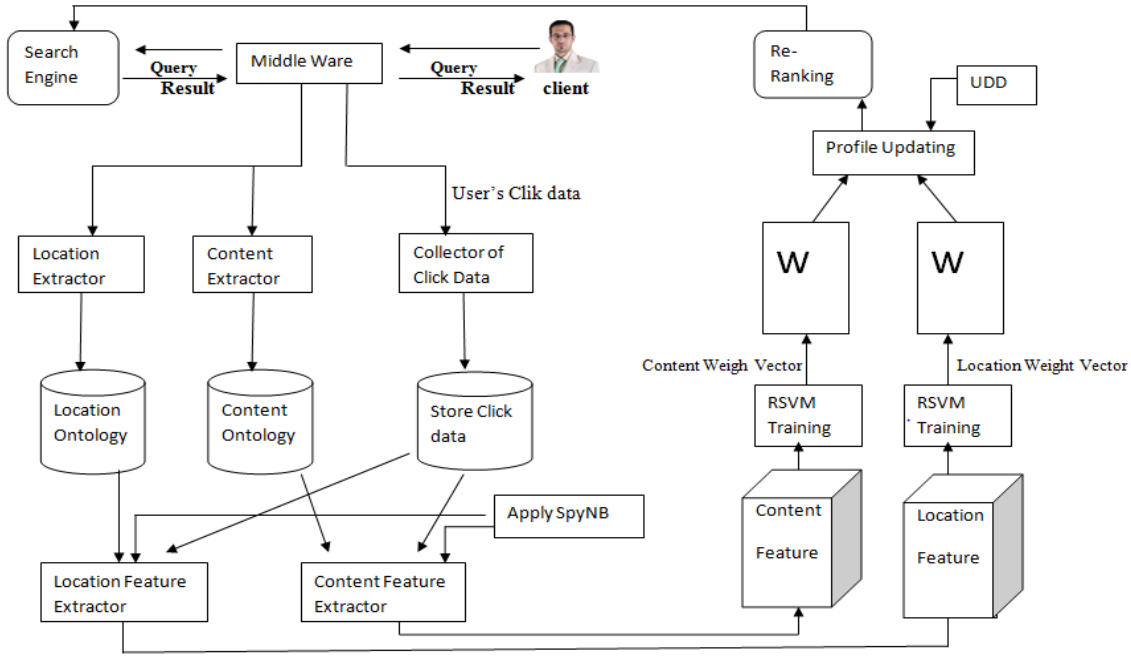


Fig1. UPSE Client Server Architecture

Click through data have been used in determining the users' preferences on their search results introduced an effective approach to predict users' conceptual preferences from clickthrough data for preference query suggestions. Search queries can be classified as content (i.e., non-geo) or location (i.e., geo) queries. A classifier is used to classify geo and non-geo queries. It was found that a significant number of queries were location queries. In order to handle the queries that focus on location information, a number of location-based search systems designed for location queries have been proposed. Location information was extracted from the web documents, which was converted into latitude-longitude pairs. When a user submits a query together with a latitude-longitude pair, the system creates a search circle centered at the specified latitude-longitude pair and retrieves documents containing location information within the search circle.

Fig.1 shows UPSE's client-server architecture, which meets three important requirements. First, computation-intensive tasks, such as RSVM training, should be handled by the UPSE server due to the limited computational power on mobile devices. Second, data transmission between client and server should be minimized to ensure fast and efficient processing of the search. Third, clickthrough data, representing precise user preferences on the search results, should be stored on the UPSE clients in order to preserve user privacy. In the UPSE's client-server architecture, UPSE clients are responsible for storing the user clickthroughs and the ontologies derived from the UPSE server. Simple tasks, such as updating clickthoughs and ontologies, creating feature vectors, and displaying re-ranked search results are handled by the UPSE clients with limited computational power. On the other

hand, heavy tasks, such as RSVM training and re-ranking of search results, are handled by the UPSE server. Moreover, in order to minimize the data transmission between client and server, the PMSE client would only need to submit a query together with the feature vectors to the UPSE server, and the server would automatically return a set of re-ranked search results according to the preferences stated in the feature vectors. The data transmission cost is minimized, because only the essential data (i.e., query, feature vectors, ontologies and search results) are transmitted between client and server during the preferences process.

III. MODULE DESCRIPTION

User Interest profiling

UPSE uses "concepts" to model the interests and preferences of a user. Since location information is important in mobile search, the concepts are further classified into two different types, namely, content concepts and location concepts. The concepts are modeled as ontologies, in order to capture the relationships between the concepts. We observe that the characteristics of the content concepts and location concepts are different. Thus, we propose two different techniques for building the content ontology and location ontology.

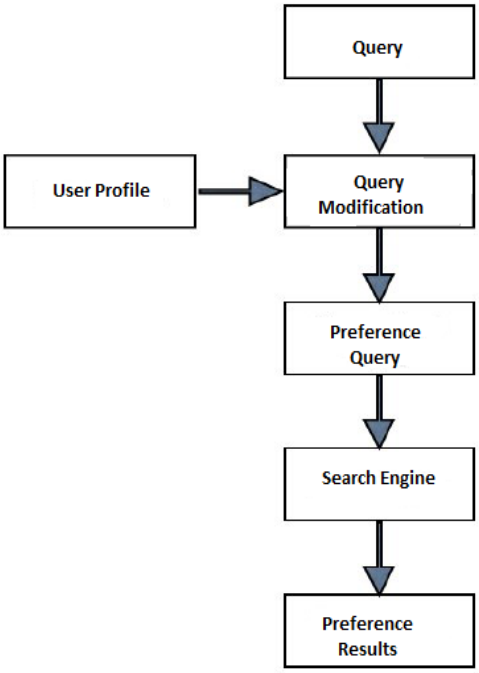


Fig 1 A

Diversity and concept entropy

UPSE consists of a content facet and a location facet. In order to seamlessly integrate the preferences in these two facets into one coherent preference framework, an important issue we have to address is how to weigh the content preference and location preference in the integration step. To address this issue, we propose to adjust the weights of content preference and location preference based on their effectiveness in the personalization process. For a given query issued by a particular user, if the preference based on the content facet is more effective than based on the location facets, more weight should be put on the content-based preferences; and vice versa.

User preference extraction and privacy preservation

Given that the concepts and clickthrough data are collected from past search activities, user's preference can be learned. These search preferences, inform of a set of feature vectors, are to be submitted along with future queries to the UPSE server for search result re-ranking. Instead of transmitting all the detailed personal preference information to the server, UPSE allows the users to control the amount of personal information exposed. In this section, we first review a preference mining algorithms, namely SpyNB Method that we adopt in UPSE, and then discuss how UPSE preserves user privacy. SpyNB learns user behavior models from preferences extracted from clickthrough data. Assuming that users only click on documents that are of interest to them, SpyNB treats the clicked documents as positive samples, and predict reliable negative documents from the unlabeled (i.e. unclicked) documents. To do the prediction, the "spy" technique incorporates a novel voting procedure into Naïve Bayes classifier to predict a negative set of documents from the unlabeled document set. The details of the SpyNB method can be found in. Let P be the positive set, U the unlabeled set and PN the predicted negative set (PN $\subset$ U) obtained from the SpyNB method. SpyNB assumes that the user would always prefer the positive set over the predicted negative set.

Preference ranking function

Upon reception of the user's preferences, Ranking SVM (RSVM) is employed to learn a personalized ranking function for rank adaptation of the search results according to the user content and location preferences. For a given query, a set of content concepts and a set of location concepts are extracted from the search results as the document features. Since each document can be

represented by a feature vector, it can be treated as a point in the feature space. Using the preference pairs as the input, RSVM aims at finding a linear ranking function, which holds for as many document preference pairs as possible. An adaptive implementation, SVM light available at, is used in our experiments.

In the following, we discuss two issues in the RSVM training process:
1) How to extract the feature vectors for a document;
2) How to combine the content and location weight vectors into one integrated weight vector.

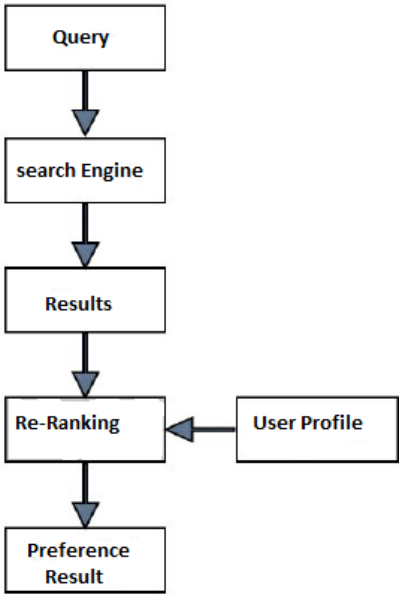


Fig 2. Re-ranking Process

IV. ALGORITHM

Algorithm 1 Training the Naive Bayes Algorithm

Input:

$L = \{l_1, l_2, \ldots, l_N\}$ /* a set of links */

Output:

Prior probabilities: $Pr(+)$ and $Pr(-)$;

Likelihoods: $Pr(w_j|+)$ and $Pr(w_j|-)$ $\forall j \in \{1, \ldots, M\}$

Procedure:

1: $Pr(+) = \frac{\sum_{i=1}^{N} \delta(+|l_i)}{N}$;

2: $Pr(-) = \frac{\sum_{i=1}^{N} \delta(-|l_i)}{N}$;

3: for each attribute $w_j \in W$ do

4: $Pr(w_j|+) = \frac{\lambda + \sum_{i=1}^{N} Num(w_j, l_i)\delta(+|l_i)}{\lambda M + \sum_{k=1}^{M}\sum_{i=1}^{N} Num(w_k, l_i)\delta(+|l_i)}$;

5: $Pr(w_j|-) = \frac{\lambda + \sum_{i=1}^{N} Num(w_j, l_i)\delta(-|l_i)}{\lambda M + \sum_{k=1}^{M}\sum_{i=1}^{N} Num(w_k, l_i)\delta(-|l_i)}$;

6: end for

When predicting unlabeled links, Naive Bayes calculates the posterior *probability* of a link, *l*, using the Bayes rule:

$$Pr(+|l) = \frac{Pr(l|+)Pr(+)}{Pr(l)}$$

When the training data contains only positive and unlabeled examples, the spying technique can be introduced to learn the Naive Bayes classifier. First, a set of positive S, are randomly selected from *P* and put in *U* to act as spies". Then, the un-labeled in *U* together with *S* are regarded as negative examples to train the Naïve Bayes classifier. The trained classifier is then used to assign *posterior probability*, *Pr* (+*jl*), to each in (UuS). After that, a threshold, *Ts*, is determined based on the *posterior probabilities* assigned to *S*. An unlabeled example in *U* is selected as a predicted negative example if its probability is less than *Ts*. As *S* act as spies", since they are positive and put into *U* pretending to be negative. During the process of prediction, the unknown positive in *U* is assumed to have similar behavior as the spies (i.e., assigned comparative probabilities). Therefore, the predicted negatives, *PNi* can be identified, which is separated from *U*. As a result, the original *U* is split into two parts after the training. One is *PNi* which may still contain some positive items (white region) due to error in the classification arising from *pi*. Another is the remaining items in *U* which may still contain some negative items (black region), also due to error in the classification. Note that *pi* returns to *P*, since it is known to be (sure) positive.

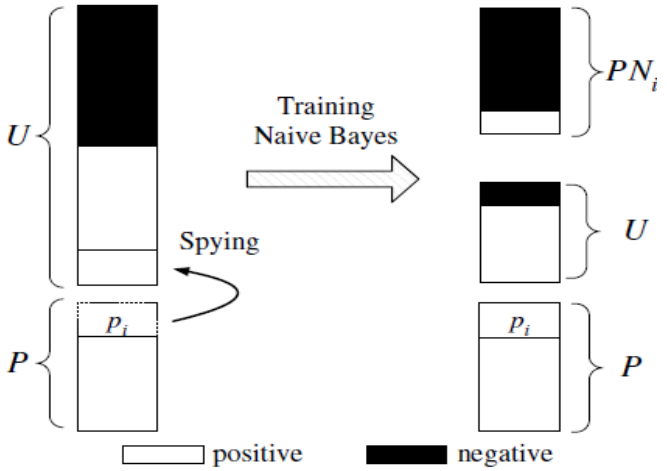
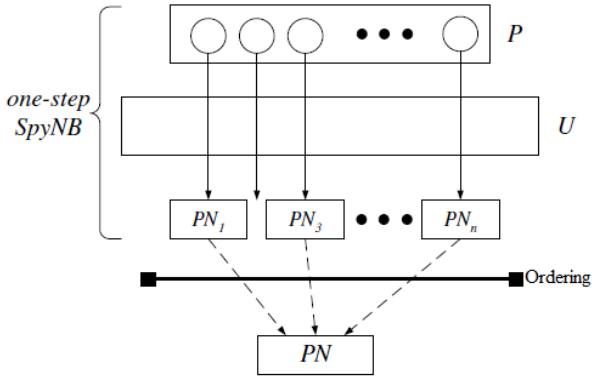


Fig 3 Principle of the spying technique

In this spying technique, the identified *PN* can be influenced by the selection of spies. As for clickthrough data, there are typically very few positive links we can make full use of all the potential spies to reduce the influence. Thus, we introduce a Ordering procedure to strengthen the spying technique further.

The idea of an ordering procedure is depicted in Figure 4 and is explained as follows. First of all, the algorithm runs the spying technique *n* times, where *n* = *jPj* is the number of positive links. Each time, a positive link, *pi*, in *P* is selected to act as a spy and put into *U* to train the Naive Bayes classifier, *NBi*. The probability, *Pr* (+*jpi*), assigned to the spy, *pi*, can be used as the threshold, *Ts*, to select a candidate predicted negative set (*PNi*). That is, any unlabeled uj , with a smaller probability of being a positive link than the spy (*Pr*(+*juj*) < *Ts*) is selected into *PNi*. As a result, *n* candidate predicted negative sets, *PN*1, *PN*2…... *PNn*, are identified. Finally, a voting procedure is used to combine all *PNi* into the ¯nal *PN*. An unlabeled example is included in the final *PN*, if and only if it appears in at least a certain number (*Tv*) of *PNi*. *Tv* is called the *order threshold*. The ordering procedure selects *PN*s based on the opinions of all spies and thus minimizes the bias of the spy selection.



V. CONCLUSION

To adapt to the user mobility, we incorporated the user's GPS locations in the preference process. We observed that GPS locations help to improve retrieval effectiveness, especially for location queries. The privacy parameters facilitate smooth control of privacy exposure while maintaining good ranking quality. Our paper is the first, to our knowledge, to interpret post search user behavior to estimate user preferences in a real web search setting. We showed that our robust models result in higher prediction accuracy than previously published techniques. We introduced new, robust, techniques for interpreting clickthrough evidence by aggregating across users and queries. Our methods result in clickthrough interpretation substantially more accurate than previously published results not specifically designed for web search scenarios. Our methods' predictions of relevance preferences are substantially more accurate than the current state-of-the-art search result ranking that does not consider user

interactions. We also presented a general model for interpreting post-search user behavior that incorporates clickthrough, browsing, and query features. By considering the complete search experience *after* the initial query and click, we demonstrated prediction accuracy far exceeding that of interpreting only the limited clickthrough information. For future work, we will investigate methods to exploit regular travel patterns and query patterns from the GPS and clickthrough data to further enhance the personalization effectiveness of UPSE

VI. ACKNOWLEDGEMENT

SRM University Scientific Research Projects Unit supports this study with the project number as 1701310012

REFERENCES

[1] X. Long and T. Suel. Optimized query execution in large search engines with global page ordering. In Proc. of the Int. Conf. on Very Large Data Bases, 2003.
[2] A. Markowetz, Y. Chen, T. Suel, X. Long, and B. Seeger. Design and Implementation of a Geographic Web Search Engine. Technical Report TR-CIS-2005-03, CIS Department, Polytechnic University, February 2005.
[3] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2000.
[4] B. Bartell, G. Cottrell, and R. Belew. Automati combination of multiple ranked retrieval systems. In Annual ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR), 1994.
[5] J. Boyan, D. Freitag, and T. Joachims. A machine learning architecture for optimizing web search engines. In AAAI Workshop on Internet Based Information Systems, August 1996.
[6] N. Fuhr. Optimum polynomial retrieval functions based on the probability ranking principle. ACM Transactions on Information Systems, 7(3):183–204, 1989.
[7] E. Agichtein, E. Brill, and S. Dumais, Improving Web Search Ranking by Incorporating User Behavior, in Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR), 2006
[8] C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, Learning to Rank using Gradient Descent, in Proceedings of the International Conference on Machine Learning (ICML), 2005.
[9] T. Joachims, Making Large-Scale SVM Learning Practical. Advances in Kernel Methods, in *Support Vector Learning*, MIT Press, 1999.