PMSE: A Personalized Mobile Search Engine

Kenneth Wai-Ting Leung, Dik Lun Lee, and Wang-Chien Lee

Abstract—We propose a personalized mobile search engine (PMSE) that captures the users' preferences in the form of *concepts* by mining their clickthrough data. Due to the importance of location information in mobile search, PMSE classifies these concepts into content concepts and location concepts. In addition, users' locations (positioned by GPS) are used to supplement the location concepts in PMSE. The user preferences are organized in an ontology-based, multifacet user profile, which are used to adapt a personalized ranking function for rank adaptation of future search results. To characterize the diversity of the concepts associated with a query and their relevances to the user's need, four entropies are introduced to balance the weights between the content and location facets. Based on the client-server model, we also present a detailed architecture and design for implementation of PMSE. In our design, the client collects and stores locally the clickthrough data to protect privacy, whereas heavy tasks such as concept extraction, training, and reranking are performed at the PMSE server. Moreover, we address the privacy issue by restricting the information in the user profile exposed to the PMSE server with two privacy parameters. We prototype PMSE on the Google Android platform. Experimental results show that PMSE significantly improves the precision comparing to the baseline.

Index Terms—Clickthrough data, concept, location search, mobile search engine, ontology, personalization, user profiling

1 Introduction

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. As a result, mobile users tend to submit shorter, hence, more ambiguous queries compared to their web search counterparts. In order to return highly relevant results to the users, mobile search engines must be able to profile the users' interests and personalize the search results according to the users' profiles.

A practical approach to capturing a user's interests for personalization is to analyze the user's clickthrough data [5], [10], [15], [18]. Leung et al. developed a search engine personalization method based on users' concept preferences and showed that it is more effective than methods that are based on page preferences [12]. However, most of the previous work assumed that all concepts are of the same type. Observing the need for different types of concepts, we present in this paper a personalized mobile search engine (PMSE) which represents different types of concepts in different ontologies. In particular, recognizing the importance of location information in mobile search, we separate concepts into *location concepts* and *content concepts*. For example, a user who is planning to visit Japan may issue the query "hotel," and click on the search results about hotels in Japan. From the clickthroughs of the query "hotel," PMSE can learn the user's content preference (e.g., "room rate" and "facilities") and location preferences ("Japan"). Accordingly, PMSE will favor results that are concerned with hotel information in Japan for future queries on "hotel." The introduction of location preferences offers PMSE 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 PMSE. 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 "Shinjuku, Tokyo," his/her position can be used to personalize the search results to favor information about nearby hotels. Here, we can see that the GPS locations (i.e., "Shinjuku, Tokyo") help reinforcing the user's location preferences (i.e., "Japan") 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 personalization process. To the best of our knowledge, our paper is the first to propose a personalization framework that utilizes a user's *content preferences* and *location preferences* as well as the *GPS locations* in personalizing search results.

In this paper, we propose a realistic design for PMSE by adopting the metasearch approach which replies on one of the commercial search engines, such as Google, Yahoo, or Bing, to perform an actual search. The client is responsible for receiving the user's requests, submitting the requests to the PMSE server, displaying the returned results, and collecting his/her clickthroughs in order to derive his/her personal preferences. The *PMSE server*, on the other hand, is responsible for handling heavy tasks such as forwarding the requests to a commercial search engine, as well as training and reranking of search results before they are returned to the client. The user profiles for specific users are stored on the PMSE clients, thus preserving privacy to the users. PMSE has been prototyped with PMSE clients on the

- *K.W.-T. Leung and D.L. Lee are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. E-mail: {kwtleung, dlee}@cse.ust.hk.*
- *W.-C. Lee is with the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802. E-mail: wlee@cse.psu.edu.*

Manuscript received 15 Feb. 2011; revised 22 July 2011; accepted 9 Dec. 2011; published online 31 Jan. 2012.
Recommended for acceptance by B.C. Ooi.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-02-0071.
Digital Object Identifier no. 10.1109/TKDE.2012.23.

Google Android platform and the PMSE server on a PC server to validate the proposed ideas.

We also recognize that the same content or location concept may have different degrees of importance to different users and different queries. To formally characterize the diversity of the concepts associated with a query and their relevances to the user's need, we introduce the notion of content and location entropies to measure the amount of content and location information associated with a query. Similarly, to measure how much the user is interested in the content and/or location information in the results, we propose *click* content and location entropies. Based on these entropies, we develop a method to estimate the personalization effectiveness for a particular query of a given user, which is then used to strike a balanced combination between the content and location preferences. The results are reranked according to the user's content and location preferences before returning to the client.

The main contributions of this paper are as follows:

- This paper studies the unique characteristics of content and location concepts, and provides a coherent strategy using a client-server architecture to integrate them into a uniform solution for the mobile environment.
- The proposed personalized mobile search engine is an innovative approach for personalizing web search results. By mining content and location concepts for user profiling, it utilizes both the content and location preferences to personalize search results for a user.
- PMSE incorporates a user's physical locations in the personalization process. We conduct experiments to study the influence of a user's GPS locations in personalization. The results show that GPS locations helps improve retrieval effectiveness for location queries (i.e., queries that retrieve lots of location information).
- We propose a new and realistic system design for PMSE. Our design adopts the server-client model in which user queries are forwarded to a PMSE server for processing the training and reranking quickly. We implement a working prototype of the PMSE clients on the Google Android platform, and the PMSE server on a PC to validate the proposed ideas. Empirical results show that our design can efficiently handle user requests.
- Privacy preservation is a challenging issue in PMSE, where users send their user profiles along with queries to the PMSE server to obtain personalized search results. PMSE addresses the privacy issue by allowing users to control their privacy levels with two privacy parameters, $minDistance$ and $expRatio$. Empirical results show that our proposal facilitates smooth privacy preserving control, while maintaining good ranking quality.
- We conduct a comprehensive set of experiments to evaluate the performance of the proposed PMSE. Empirical results show that the ontology-based user profiles can successfully capture users' content and location preferences and utilize the preferences to produce relevant results for the users. It significantly outperforms existing strategies which use either content or location preference only.

TABLE 1
Clickthrough for the Query "Hotel"

Doc	Search Results	c_i	l_i
d_1	Hotels.com	room rate	*international*
$\mathbf{d_2}$	**JapanHotel.net**	**reservation, room rate**	**Japan**
d_3	Hotel Wiki	accommodation	*international*
d_4	US Hotel Guides	map, room rate	USA, California
d_5	Booking.com	online reservation	USA
$\mathbf{d_6}$	**JAL Hotels**	**meeting room**	**Japan**
$\mathbf{d_7}$	**Shinjuku Prince**	**facility**	**Japan, Shinjuku**
d_8	Discount Hotels	discount rate	*international*

The rest of the paper is organized as follows: Related work is reviewed in Section 2. In Section 3, we present the architecture and system design of PMSE. In Section 4, we present our method for building the content and location ontologies. In Section 5, we introduce the notion of content and location entropies, and show how their usage in search personalization. In Section 6, we review the method to extract user preferences from the clickthrough data. In Section 7, we discuss the Ranking SVM (RSVM) method [10] for learning a linear weight vector (consisting both content and location features) to rank the search results. We present the performance results in Section 8, and conclude the paper in Section 9.

2 Related Work

Clickthrough data have been used in determining the users' preferences on their search results. Table 1, showing an example clickthrough data for the query "hotel," composes of the search results and the ones that the user clicked on (bolded search results in Table 1). As shown, c_is are the content concepts and l_is are the location concepts extracted from the corresponding results. Many existing personalized web search systems [6], [10], [15], [18] are based clickthrough data to determine users' preferences. Joachims [10] proposed to mine document preferences from clickthrough data. Later, Ng et al. [15] proposed to combine a spying technique together with a novel voting procedure to determine user preferences. More recently, Leung et al. [12] introduced an effective approach to predict users' conceptual preferences from clickthrough data for personalized query suggestions.

Search queries can be classified as **content (i.e., non-geo)** or **location (i.e., geo)** queries. Examples of location queries are "hong kong hotels," "museums in london," and "virginia historical sites." In [9], Gan et al. developed a classifier to classify geo and non-geo queries. It was found that a significant number of queries were location queries focusing on location information. 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. Yokoji [22] proposed a location-based search system for web documents. Location information was extracted from the web documents, which was

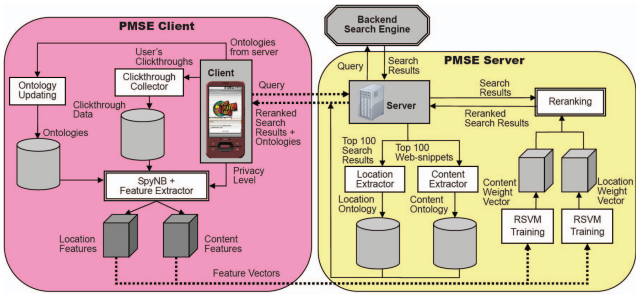


Fig. 1. The general process flow of PMSE.

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.

Later on, Chen et al. [7] studied the problem of efficient query processing in location-based search systems. A query is assigned with a query footprint that specifies the geographical area of interest to the user. Several algorithms are employed to rank the search results as a combination of a textual and a geographic score. More recently, Li et al. [13] proposed a probabilistic topic-based framework for location-sensitive domain information retrieval. Instead of modeling locations in latitude-longitude pairs, the model assumes that users can be interested in a set of location-sensitive topics. It recognizes the geographical influence distributions of topics, and models it using probabilistic Gaussian Process classifiers.

The differences between existing works and ours are

- Most existing location-based search systems, such as [22], require users to manually define their location preferences (with latitude-longitude pairs or text form), or to manually prepare a set of location-sensitive topics. PMSE profiles both of the user's content and location preferences in the ontology-based user profiles, which are automatically learned from the clickthrough and GPS data without requiring extra efforts from the user.
- We propose and implement a new and realistic design for PMSE. To train the user profiles quickly and efficiently, our design forwards user requests to the PMSE server to handle the training and reranking processes.
- Existing works on personalization do not address the issues of privacy preservation. PMSE addresses this issue by controlling the amount of information in the client's user profile being exposed to the PMSE server using two privacy parameters, which can control privacy smoothly, while maintaining good ranking quality.

3 System Design

Fig. 1 shows PMSE's client-server architecture, which meets three important requirements. First, computation-intensive tasks, such as RSVM training, should be handled by the PMSE 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 PMSE clients in order to preserve user privacy.

In the PMSE's client-server architecture, PMSE clients are responsible for storing the user clickthroughs and the ontologies derived from the PMSE server. Simple tasks, such as updating clickthoughs and ontologies, creating feature vectors, and displaying reranked search results are handled by the PMSE clients with limited computational power. On the other hand, heavy tasks, such as RSVM training and reranking of search results, are handled by the PMSE 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 PMSE server, and the server would automatically return a set of reranked 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 personalization process. PMSE's design addressed the issues: 1) limited computational power on mobile devices, and 2) data transmission minimization.

PMSE consists of two major activities:

1. **Reranking the search results at PMSE server.** When a user submits a query on the PMSE client, the query together with the feature vectors containing the user's content and location preferences (i.e., filtered ontologies according to the user's privacy setting) are forwarded to the PMSE server, which in turn obtains the search results from the back-end search engine (i.e., Google). The content and location concepts are extracted from the search results and organized into ontologies to capture the relationships between the concepts. The server is used to perform ontology extraction for its speed. The feature vectors from the client are then used in RSVM training to obtain a *content weight vector* and a *location weight vector*, representing the user interests based on the user's content and location preferences for the reranking. Again, the training process is performed on the server for its speed. The search results are then reranked according to the weight vectors obtained from the RSVM training. Finally, the reranked results and the extracted ontologies for the personalization of future queries are returned to the client.
2. **Ontology update and clickthrough collection at PMSE client.** The ontologies returned from the PMSE server contain the concept space that models the relationships between the concepts extracted from the search results. They are stored in the ontology database on the client.[1] When the user clicks on a search result, the clickthrough data together with the associated content and location concepts are stored in the clickthrough database on

1. Note that the ontologies stored on the client are the same as what was extracted on the PMSE server.

the client. The clickthroughs are stored on the PMSE clients, so the PMSE server does not know the exact set of documents that the user has clicked on. This design allows user privacy to be preserved in certain degree. Two privacy parameters, $minDistance$ and $expRatio$, are proposed to control the amount of personal preferences exposed to the PMSE server. If the user is concerned with his/her own privacy, the privacy level can be set to high so that only limited personal information will be included in the feature vectors and passed along to the PMSE server for the personalization. On the other hand, if a user wants more accurate results according to his/her preferences, the privacy level can be set to low so that the PMSE server can use the full feature vectors to maximize the personalization effect.

Since the ontologies can be derived online at the PMSE server, an alternative system design is for the user to pass only the clickthrough data to the PMSE server, and to perform both feature extraction and RSVM training on the PMSE server to train the weight vectors for reranking. However, if all clickthroughs are exposed to the PMSE server, the server would know exactly what the user has clicked. To address privacy issues, clickthroughs are stored on the PMSE client, and the user could adjust the privacy parameters to control the amount of personal information to be included in the feature vectors, which are forwarded to the PMSE server for RSVM training to adapt personalized ranking functions for content and location preferences.

4 User Interest Profiling

PMSE 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 (in Section 4.1) and location ontology (in Section 4.2). The ontologies indicate *a possible concept space* arising from a user's queries, which are maintained along with the clickthrough data for future preference adaptation. In PMSE, we adopt ontologies to model the concept space because they not only can represent concepts but also capture the relationships between concepts. Due to the different characteristics of the content concepts and location concepts, in Section 4.1, we first discuss our method to mine and build the content ontology from the search results. In Section 4.2, we present our method to derive a location ontology from the search results.

4.1 Content Ontology

Our content concept extraction method first extracts all the keywords and phrases (excluding the stop words) from the web-snippets[2] arising from q. If a keyword/phrase exists frequently in the web-snippets arising from the query q, we would treat it as an important concept related to the query, as it coexists in close proximity with the query in the top documents. The following support formula, which is inspired by the well-known problem of finding frequent item sets in data mining [8], is employed to measure the importance of a particular keyword/phrase c_i with respect to the query q:

$$support(c_i) = \frac{sf(c_i)}{n} \cdot |c_i|, \tag{1}$$

where $sf(c_i)$ is the snippet frequency of the keyword/phrase c_i (i.e., the number of web-snippets containing c_i), n is the number of web-snippets returned and $|c_i|$ is the number of terms in the keyword/phrase c_i. If the support of a keyword/phrase c_i is higher than the threshold s ($s = 0.03$ in our experiments), we treat c_i as a concept for q.

We adopt the following two propositions to determine the relationships between concepts for ontology formulation:

- **Similarity.** Two concepts which coexist a lot on the search results might represent the same topical interest. If $coexist(ci, cj) > \delta_1$ (δ_1 is a threshold), then c_i and c_j are considered as similar.
- **Parent-child relationship.** More specific concepts often appear with general terms, while the reverse is not true. Thus, if $pr(cj|ci) > \delta_2$ (δ_2 is a threshold), we mark c_i as c_j's child. For example, the more specific concept "meeting facility" tends to occur together with the general concept "facilities," while the general concept "facilities" might also occur with concepts such as "meeting room" or "swimming pool," i.e., not only with the concept "meeting facility."

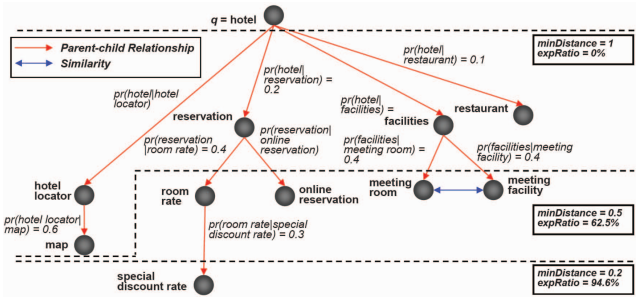


Fig. 2. Ontology for $q = "hotel"$ with $p = 0.2, 0.5, 1.0$.

Fig. 2 shows an example content ontology created for the query "hotel," where content concepts linked with a one-sided arrow ($\rightarrow$) are parent-child concepts, and concepts linked with a double-sided arrow ($\leftrightarrow$) are similar concepts. Fig. 2 shows the possible concept space determined for the query "hotel," while the clickthrough data determine the user preferences on the concept space. In general, the ontology covers more than what the user actually wants. The concept space for the query "hotel" consists of "map," "reservation," "room rate,"..., etc. If the user is indeed interested in information about hotel rates and clicks on pages containing "room rate" and "special discount rate" concepts, the captured clickthrough favors the two clicked concepts. Feature vectors containing the concepts "room rate" and "special discount rate" as positive preferences will

2. "Web-snippet" denotes the title, summary, and URL of a Webpage returned by search engines.

TABLE 2
Statistics of the Location Ontology

No. of Countries	7	Total No. of Nodes	16899
No. of Regions	190	Country-Region Edges	190
No. of Provinces	6699	Region-Province Edges	1959
No. of Towns	10003	Province-City Edges	14897

be created corresponding to the query "hotel." As indicated in Fig. 2, when the query is issued again later, these feature vectors will be transmitted to the PMSE server and transformed into a content weight vector to rank the search results according to the user's content preferences. The details of the transformation will be discussed in Section 7.1.

4.2 Location Ontology

Our approach for extracting location concepts is different from that for extracting content concepts. We observe two important issues in location ontology formulation. First, a document usually embodies only a few location concepts, and thus only very few of them co-occur with the query terms in web-snippets. To alleviate this problem, we extract location concepts from the full documents. Second, the similarity and parent-child relationship cannot be accurately derived statistically because the limited number of location concepts embodied in documents. Furthermore, many geographical relationships among locations have already been captured as facts. Thus, we obtain about 17,000 city, province, region, and country names [2] and [4], and create a predefined location ontology among these locations. We organize all the cities as children under their provinces, all the provinces as children under their regions, and all the regions as children under their countries. The statistics of our location ontology are provided in Table 2.

The predefined location ontology is used to associate location information with the search results. All of the keywords and key-phrases from the documents returned for query q are extracted. If a keyword or key-phrase in a retrieved document d matches a location name in our predefined location ontology, it will be treated as a location concept of d. For example, assume that document d contains the keyword "Los Angeles." "Los Angeles" would then be matched against the location ontology. Since "Los Angeles" is a location in our location ontology, it is treated as a location concept related to d. Furthermore, we would explore the predefined location hierarchy, which would identify "Los Angeles" as a city under the state "California." Thus, the location "/United States/California/Los Angeles/" is associated with document d. If a concept matches several nodes in the location ontology, all matched locations will be associated with the document.

Similar to the content ontology, the location ontology together with clickthrough data are used to create feature vectors containing the user location preferences. They will then be transformed into a location weight vector to rank the search results according to the user's location preferences.

5 Diversity and Concept Entropy

PMSE consists of a content facet and a location facet. In order to seamlessly integrate the preferences in these two facets into one coherent personalization 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 personalization based on preferences from the content facet is more effective than based on the preferences from the location facets, more weight should be put on the content-based preferences, and vice versa. The notion of *personalization effectiveness* is derived based on the diversity of the content and location information in the search results as discussed in Section 5.1, and the diversity of user interests the content and location information associated with a query as discussed in Section 5.2. We show that it can be used to effectively combine a user's content and location preferences for reranking the search results in Section 8.4.

5.1 Diversity of Content and Location Information

Different queries may be associated with different amount of content and location information. To formally characterize the content and location properties of the query, we use *entropy* to estimate the amount of content and location information retrieved by a query. In information theory [17], *entropy* indicates the uncertainty associated with the information content of a message from the receiver's point of view. In the context of search engine, entropy can be employed in a similar manner to denote the uncertainty associated with the information content of the search results from the user's point of view. Since we are concerned with content and location information only in this paper, we define two entropies, namely, **content entropy** $H_C(q)$ and **location entropy** $H_L(q)$, to measure, respectively, the uncertainty associated with the content and location information of the search results

$$H_C(q) = -\sum_{i=1}^{k} p(c_i)\log p(c_i) \quad H_L(q) = -\sum_{i=1}^{m} p(l_i)\log p(l_i), \tag{2}$$

where k is the number of content concepts $C = \{c_1, c_2, \ldots, c_k\}$ extracted, $|c_i|$ is the number of search results containing the content concept c_i, $|C| = |c_1| + |c_2| + \cdots + |c_k|$, $p(c_i) = \frac{|c_i|}{|C|}$, m is the number of location concepts $L = \{l_1, l_2, \ldots, l_m\}$ extracted, $|l_i|$ is the number of search results containing the location concept l_i, $|L| = |l_1| + |l_2| + \cdots + |l_m|$, and $p(l_i) = \frac{|l_i|}{|L|}$.

5.2 Diversity of User Interests

Apart from the uncertainty associated with the content and location information of the search results, we also introduce *click content entropy* and *click location entropy* to indicate, respectively, the diversity of a user's interest on the content and location information returned from a query. The entropy equations for click content and location concepts are similar to (2), but only the clicked pages, and hence the clicked concepts, are considered in the formula. Since the click entropies reflects the user's actions in response to the search results, they can be used as an indication of the diversity of the user's interests. Formally, the click content entropy $H_{\overline{C}}(q,u)$ and click location entropy $H_{\overline{L}}(q,u)$ of a query q submitted by the user u are defined as follows:

$$H_{\overline{C}}(q,u) = -\sum_{i=1}^{t} p(\overline{c}_{iu}) \log p(\overline{c}_{iu}) \quad (3)$$

$$H_{\overline{L}}(q,u) = -\sum_{i=1}^{v} p(\overline{l}_{iu}) \log p(\overline{l}_{iu}), \quad (4)$$

where t is the number of content concepts clicked by user u, $\overline{C}_u = \{\overline{c}_{1u}, \overline{c}_{2u}, \ldots, \overline{c}_{tu}\}$, $|\overline{c}_{iu}|$ is the number of times that the content concept c_i has been clicked by

$$u, |\overline{C}_u| = |\overline{c}_{1u}| + |\overline{c}_{2u}| + \cdots + |\overline{c}_{tu}|, p(\overline{c}_i, u) = \frac{|\overline{c}_{iu}|}{|\overline{C}_u|}, v$$

is the number of location concepts

$$\overline{L}_u = \{\overline{l}_{1u}, \overline{l}_{2u}, \ldots, \overline{l}_{vu}\}$$

clicked by u, $|\overline{l}_{iu}|$ is the number of times that the location concept l_i has been clicked by u, $|\overline{L}_u| = |\overline{l}_{1u}| + |\overline{l}_{2u}| + \cdots + |\overline{l}_{vu}|$, and $p(\overline{l}_i, u) = \frac{|\overline{l}_{iu}|}{|\overline{L}_u|}$.

5.3 Personalization Effectiveness

As discussed in the last section, a query result set with high content/location entropy indicates that it has a high degree of ambiguity. Thus, applying personalization on the search results helps the user to find out the relevant information. On the other hand, when the content/location entropy is low, meaning that the returned result set is already very focused and should have matched the query quite precisely, personalization can do very little in further improving the precision of the result.

For click entropies, we expect that the higher the click content/location entropies, the worse the personalization effectiveness, because high click content/location entropies indicate that the user is clicking on the search results with high uncertainty, meaning that the user is interested in a diversity of information in the search results. When the user's interests are very broad (or the clickthroughs could be "noisy" due to irrelevant concepts existing in the clicked documents), it is difficult to 1) find out the user's actual needs and 2) personalize the search results toward the user's interest. On the other hand, if the click content/location entropies are low, the personalization effectiveness would be high because the user has a focus on certain precise topic in the search results (only a small set of content/location concepts has been clicked by the user). Hence, the profiling process can identify the user's information needs and the personalization process can personalize the results to meet those needs.

Based on the above reasoning, we propose to estimate the personalization effectiveness using the extracted content and location concepts with respect to user u as follows:

$$e_C(q,u) = \frac{H_C(q)}{H_{\overline{C}}(q,u)} \qquad e_L(q,u) = \frac{H_L(q)}{H_{\overline{L}}(q,u)}. \quad (5)$$

We expect that queries with high $e_C(q,u)$ and $e_L(q,u)$ would yield better personalization results as described in [11].

6 User Preferences 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 PMSE server for search result reranking. Instead of transmitting all the detailed personal preference information to the server, PMSE 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 PMSE, and then discuss how PMSE preserves user privacy.

SpyNB [15] 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 [14] to predict a negative set of documents from the unlabeled document set. The details of the SpyNB method can be found in [15]. 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. Thus, user preference pairs can be obtained as follows:

$$d_i < d_j\,, \quad \forall l_i \in P\,, \quad l_j \in PN. \quad (6)$$

The preference pairs together with the extracted ontologies are used to derive a set of feature vectors on the PMSE client for submission along with future queries to the PMSE server which in turn finds a linear ranking function that best describes the user preferences using RSVM. In our client-server model, the click histories are entirely stored on the PMSE clients as shown in Fig. 1. The back-end search engine has no knowledge of a user's click history. Hence, the user's privacy is ensured. The PMSE server is a trusted server, which would not store all the clickthrough data. It is aware of the user's preferences, but the how much it knows is controlled by the privacy settings set by the client. The PMSE client stores the user's clickthrough and has control on the privacy setting. It would create a feature vector based on its clickthrough data and the filtered ontology according to the privacy settings at different $expRatio$. The feature vector is then forwarded to the PMSE server for the personalization. Thus, the PMSE server only knows about the filtered concepts that the client prefers in the form of a feature vector.

To control the amount of personal information exposed out of users' mobile devices, PMSE filters the ontologies according to the user's privacy level setting, which are specified with two privacy parameters, $minDistance$ and $expRatio$. The privacy preserving technique in PMSE aims at filtering concepts that are too specific. Thus, $minDistance$ is used to measure whether a concept is far away from the root (i.e., too specific) in the ontology-based user profiles. For example, a user who searches for medicine information may not want to reveal the specific drugs she/he is looking for. Additionally, an information-theoretic parameter $expRatio$, proposed by Xu et al. [21] is employed, to

measure the amount of private information exposed in the user profiles. There is a close relationship between privacy and personalization effectiveness. The lower the privacy level (the more information being provided to the PMSE server for the personalization), the better the personalization results. Thus, there is a tradeoff between them. If the user is concerned with his/her own privacy, the privacy level can be set to high to provide only limited personal information to the PMSE server. Nevertheless, the personalization effect will be less effective. On the other hand, if a user wants more accurate results according to his/her preferences, the privacy level can be set to low, such that the PMSE server can use the full user profile for the personalization process, and provide better results.

PMSE employs *distance* to filter the concepts in the ontology. If a concept c_{i+1} is a child of another concept c_i in our ontology-based user profile, then c_i and c_{i+1} are connected with an edge whose distance is defined by $d(c_i, c_{i+1}) = \frac{1}{pr(c_{i+1}|c_i)}$ (the higher the chance that c_i and c_{i+1} are related, the shorter the edge between c_i and c_{i+1}). We aim at filtering the concepts that are $minDistance$ close to the leaf concepts (i.e., the most specific concepts), and the concept c_i will be pruned when the following condition is satisfied:

$$\frac{D(c_{i-1}, c_k)}{D(root, c_{i-1}) + D(c_{i-1}, c_k)} < minDistance, \tag{7}$$

where c_{i-1} is the direct parent of c_i, and c_k is the leaf concept, which is furthest away from

$$c_i(\text{argmax}_{c_k} D(c_{i+1}, c_k)),$$

in the ontology. $D(c_{i-1}, c_k) = d(c_{i-1}, c_i) + d(c_{i+1}, c_{i+2}) + \cdots + d(c_{k-1}, c_k)$ is the total distance from c_{i-1} to c_k, and $D(root, c_i)$ is the total distance from the root node to c_{i-1}.

The filtered user profiles (with specific concepts c_{i+1} pruned) are transmitted to the PMSE server. Here, $expRatio$ is employed to measure the amount of information being pruned in the filter user profiles. Note that the complete user profile is $U_{q,0}$, while the protected user profile for the query q with $minDistance = p$ is $U_{q,p}$. Thus, the concept entropy $H_C(U_{q,p})$ of the user profiles can be computed using the following equation:

$$H_C(U_{q,p}) = -\sum_{c_i \in U_{q,p}} pr(c_i) \log pr(c_i), \tag{8}$$

where c_i is any concept that exists in the user profile $U_{q,p}$ for the query q. Given $H_C(U_{q,0})$ and $H_C(U_{q,p})$, the exposed privacy $expRatio_{q,p}$ can be computed as

$$expRatio_{q,p} = \frac{H_C(U_{q,p})}{H_C(U_{q,0})}. \tag{9}$$

Fig. 2 shows $U_{hotel,0.2}$, $U_{hotel,0.5}$, and $U_{hotel,1.0}$ for the query "hotel." When $minDistance$=0.2, only the very specific concept "special discount rate" is pruned from $U_{hotel,0}$. The exposed privacy $expRatio_{hotel,0.2}$ is 94.6 percent

$$\left(expRatio_{disneyland,0.2} = \frac{H_C(U_{hotel,0.2})}{H_C(U_{hotel,0})} = \frac{0.682}{0.721} = 94.6\%.\right)$$

When $minDistance = 0.5$, four specific concepts ("room rate," "online reservation," "meeting room," and "meeting facility") are pruned. Notice that "map" is not removed when $minDistance = 0.5$, because both "map" and "hotel locator" are rare concepts with low *support*. Since "map" and "hotel locator" are closely related with $pr(hotellocator|map) = 0.6$, if "hotel locator" is pruned, "map" will likely be pruned too. If both of them are pruned, the protected user profile can no longer determine the user's preferences on these two concepts. Thus, "map" is retained unless $minDistance$ is very high ($minDistance > 0.92$). The exposed privacy $expRatio_{hotel,0.5}$ is 62.5 percent

$$\left(expRatio_{hotel,0.5} = \frac{H_C(U_{hotel,0.5})}{H_C(U_{hotel,0})} = \frac{0.451}{0.721} = 62.5\%\right).$$

Finally, when $minDistance = 1.0$, all concepts in the user profile are pruned

$$\left(expRatio_{hotel,1.0} = \frac{H_C(U_{hotel,1.0})}{H_C(U_{hotel,0})} = \frac{0}{0.721} = 0\%\right).$$

7 Personalized Ranking Functions

Upon reception of the user's preferences, Ranking SVM [10] 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, *SVMlight* available at [3], 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.

7.1 Extracting Features for Training

We propose two feature vectors, namely, **content feature vector** (denoted by $\phi_C(q, d)$) and **location feature vector** (denoted by $\phi_L(q, d)$) to represent the content and location information associated with documents. The feature vectors are extracted by taking into account the concepts existing in a documents and other related concepts in the ontology of the query. For example, if a document d_k embodies the content concept c_i and location concept l_i, the weight of component c_i in the content feature vector $\phi_C(q, d_k)$ of document d_k is incremented by one as defined in (10), and the weight of l_i in the location feature vector $\phi_L(q, d_k)$ is incremented by one as defined in (12). The similarity and parent-child relationships of the concepts in the extracted concept ontologies are also incorporated in the training based on the following four different types of relationships:

1. **Similarity**,
2. **Ancestor**,
3. **Descendant**, and
4. **Sibling**

in our ontologies. We argue that all of the above relationships may help the users to find more related information in

the same class. Therefore, we assign the predetermined weights to related concepts. The related concepts components in content and location feature vectors are thus incremented by the weights as defined in (11) and (13).

The extraction of content feature vector and location feature vector are defined formally as follows:

1. **Content feature vector**. If content concepts c_i is in a web-snippet s_k, their values are incremented in the content feature vector $\phi_C(q, d_k)$ with the following equation:

$$\forall c_i \in s_k, \phi_C(q, d_k)[c_i] = \phi_C(q, d_k)[c_i] + 1. \quad (10)$$

For other content concepts c_j that are related to the content concept c_i (either they are similar or c_j is the ancestor/descendant/sibling of c_i) in the content ontology, they are incremented in the content feature vector $\phi_C(q, d_k)$ according to the following equation:

$$\begin{aligned}\forall c_i \in s_k, \phi_C(q, d_k)[c_j] &= \phi_C(q, d_k)[c_j] \\ &+ sim_R(c_i, c_j) + ancestor(c_i, c_j) \\ &+ descendant(c_i, c_j) + sibling(c_i, c_j).\end{aligned} \quad (11)$$

2. **Location feature vector**. If location concept l_i is in a web-snippet d_k, its value is incremented in the location feature vector $\phi_L(q, d_k)$ with the following equation:

$$\forall l_i \in d_k, \phi_L(q, d_k)[l_i] = \phi_L(q, d_k)[l_i] + 1. \quad (12)$$

For other location concepts l_j that are related to the concept l_i (l_j is the ancestor/descendant/sibling of l_i) in the location ontology, they are incremented in the location feature vector $\phi_L(q, d_k)$ according to the following equation:

$$\begin{aligned}\forall l_i \in d_i,\ \phi_L(q, d_k)[l_j] &= \phi_L(q, d_k)[c_j] + ancestor(l_i, l_j) \\ &+ descendant(l_i, l_j) + sibling(l_i, l_j).\end{aligned} \quad (13)$$

7.2 GPS Data and Combination of Weight Vectors

The content feature vector $\phi_C(q, d)$ together with the document preferences obtained from SpyNB are served as input to RSVM training to obtain the **content weight vector** $\overrightarrow{w_{C,q,u}}$. The **location weight vector** $\overrightarrow{w_{L,q,u}}$ is obtained similarly using the location feature vector $\phi_L(q, d)$ and the document preferences. $\overrightarrow{w_{C,q,u}}$ and $\overrightarrow{w_{L,q,u}}$ represent the content and location user profiles for a user u on a query q in our method.

$\overrightarrow{w_{C,q,u}}$ and $\overrightarrow{w_{L,q,u}}$ represent the user preferences derived from the clickthrough data only. As discussed in Section 1, GPS locations are important information that can be useful in personalizing the search results. For example, a user may use his/her mobile device to find movies on show in the nearby cinemas. Thus, PMSE incorporates the GPS locations into the personalization process by tracking the visited locations. This function is realized by the embedded GPS modules on the PMSE client. We believe that users are possibly interested in locations where they have visited. Thus, our goal is to integrate the factor of GPS locations in $\overrightarrow{w_{L,q,u}}$ to reflect the possible preferences. Thus, if a user has visited the GPS location l_r, the weight of the location concept in $\overrightarrow{w_{L,q,u}}[l_r]$ is incremented according the following equation:

$$\forall l_r \text{ that } u \text{ has visited }, \overrightarrow{w_{L,q,u}}[l_r] = \overrightarrow{w_{L,q,u}}[l_r] + w_{GPS}(u, l_r, t_r), \quad (14)$$

where $w_{GPS}(u, l_r, t_r)$ is the weight being added to the GPS location l_r, and t_r is the number of location visited since the user visit l_r ($t_r = 0$ means the current location).[3] Hence, we assume that the location that the user has visited a long time ago is less important than the location that the user has recently visited. The weight $w_{GPS}(u, l_r, t_r)$ being added to the $\overrightarrow{w_{L,q,u}}[l_r]$ according to the following decay equation:

$$w_{GPS}(l_r, t_r) = w_{GPS\_0} \cdot e^{-t_r}, \quad (15)$$

where w_{GPS_0} is the initial weight for the decay function when $t_r = 0$. In the experiments, different w_{GPS_0} are employed in order to study the influence of the GPS locations in the personalization. We observe that the GPS locations help improving retrieval effectiveness for location queries. By default, w_{GPS_0} is set to 0.1 in order to maximize the effect of the improvement as discussed in Section 8.5.

The set of location concepts $\{l_s\}$ that are closely related to the GPS location l_r (l_s is the ancestor/descendant/sibling of l_r) in the location ontology are also possible candidates that the user may be interested in. Thus, the weight of the location concept l_s in the weight vector $\overrightarrow{w_{L,q,u}}[l_s]$ is incremented according to the following equation:

$$\begin{aligned}\forall l_r\ \overrightarrow{w_{L,q,u}}[l_s] &= \overrightarrow{w_{L,q,u}}[l_s] + w_{GPS}(u, l_r, t_r) \\ &\cdot (ancestor(l_i, l_j) + descendant(l_i, l_j) + sibling(l_i, l_j)).\end{aligned} \quad (16)$$

As discussed in Section 5.3, the higher $e_C(q, u)$ and $e_L(q, u)$ are, the more effective the personalization in content and location facets, respectively. To optimize the personalization effect, we use the following formula to combine the two weight vectors, $\overrightarrow{w_{C,q,u}}$ and $\overrightarrow{w_{L,q,u}}$, linearly according to the values of $e_C(q, u)$ and $e_L(q, u)$, to obtain the final weight vector $\overrightarrow{w_{q,u}}$ for user u's ranking. The two weight vectors, $\overrightarrow{w_{C,q,u}}$ and $\overrightarrow{w_{L,q,u}}$, are first normalized before the combination

$$\overrightarrow{w_{q,u}} = \frac{e_C(q, u)}{e_C(q, u) + e_L(q, u)} \cdot \overrightarrow{w_{C,q,u}} + \frac{e_L(q, u)}{e_C(q, u) + e_L(q, u)} \cdot \overrightarrow{w_{L,q,u}}. \quad (17)$$

Let $e(q, u) = \frac{e_C(q,u)}{e_C(q,u)+e_L(q,u)}$, then we get the following equation from (17):

$$\overrightarrow{w_{q,u}} = e(q, u) \cdot \overrightarrow{w_{C,q,u}} + (1 - e(q, u)) \cdot \overrightarrow{w_{L,q,u}}. \quad (18)$$

After $\overrightarrow{w_{q,u}}$, PMSE will rank the documents in the returned search according to the following formula:

$$f(q, d) = \overrightarrow{w_{q,u}} \cdot \phi(q, d), \quad (19)$$

3. GPS data are usually seen in geometric form (i.e., coordinates), the coordinates are mapped into semantic locations using Google maps.

where q is a query, d is a document in the search results, $\overrightarrow{w_{q,u}}$ is the weight vector defined in (17), and $\phi(q,d)$ is a feature vector representing the match between q and d.

8 Experimental Evaluation

In this section, we evaluate the effectiveness of PMSE. We describe the experimental setup in Section 8.1. Then, we evaluate the ranking quality of PMSE with different user profiles in Section 8.2. In Section 8.3, we study the effect of noise clicks on the personalization quality. The accuracy of the estimated facet combination threshold is evaluated in Section 8.4. The influence of the GPS locations in PMSE is evaluated in Section 8.5. Finally, we evaluate the effectiveness of $minDistance$ and $expRatio$ in Section 8.6.

8.1 Experiment Setup

8.1.1 Methodology

The experiment aims to answer the following question: Given that a user is only interested in some specific aspects of a query, can PMSE generate a ranking function personalized to the user's interest from the user's clickthroughs? To answer this question, we need to evaluate the search results before and after personalization. The difficulty of the evaluation is that only the user who conducted the search can tell which of the results are relevant to his/her search intent. This is in contrast to the evaluation of traditional information retrieval systems [20], where expert judges are employed to judge the relevance of a set of documents (e.g., TREC) based on a detailed description of the information need. The relevance judgment is then considered the standard to judge the quality of the search results. This evaluation method clearly cannot be applied to personalized search, because what an expert judge considered as relevant to a query needs not be relevant from another user's point of view because the same query issued by two different users may have different goals behind it.

Another difficulty of evaluating personalized search systems is that since relevance judgment is highly dependent on the users, care must be taken to ensure that the users' behaviors are not affected by experimental artifacts. Two important issues are considered in the experiment setup. First, it is not advisable to ask the user to conduct the same search on two systems, one with personalization and one without, and compare the two search results. This is because once the user has conducted a search on one system, his/her behavior would be affected by it and thus would be biased in the other system. In our experiment, the user only conducts search on the system before personalization as if he/she is using a regular search engine. Then, the user evaluates the relevance of the search results manually (as in traditional information retrieval evaluation) according to his/her search intents. After these steps, the training of PMSE and the measurement of retrieval effectiveness are both conducted offline without the involvement of the user (see the next section for details). Second, as a user becomes more experienced with the system, answers of the subsequent queries could become more and more accurate. Thus, in the experiments, we limit the number of queries for each user to five. In other words, instead of using a small number of users each searching a large number of queries we use a large number of users each searching a small number of queries to prevent the users from overly adapted to the system.

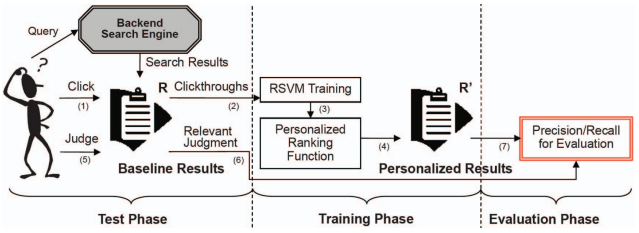


Fig. 3. Flow of the evaluation process.

Finally, since we are interested in seeing how PMSE can adapt to a user's personal interests even when the queries have multiple interpretations, the test queries used in the experiments by intention all have broad meanings. In addition, the topical category associated with each query is also very broad, providing the users enough room to decide which specific aspects of the query he/she wants to focus on. For example, when the topical category is "photography" and the query is "canon," the user will look for information about "canon" digital cameras but not "canon" laser printers or "canon" as a location name. Yet, within the "photography" category, the user can decide what to look for, e.g., specific products, photo gallery, etc. A similar evaluation approach has been used in [19].

8.1.2 Limitations

While the methodology tries to minimize the user's involvement in the experiment, it is nevertheless a controlled experiment and thus has some limitations. First, the number of users and queries in the experiments are small. This means that the results from the experiments cannot be construed as representative in diverse situations. Second, since users are given with predefined queries and topical interests, they have to synthesize their information needs from the given queries and topical interests and conduct their searches correspondingly. Thus, their search behaviors in the experiments may be quite different from what they might have exhibited when they attempt to resolve real-life information needs. Ideally, a large-scale user study should be conducted in which PMSE is subjected to real-life use, users' behaviors are monitored transparently and satisfaction of the users is analyzed and compared with other systems, but a large-scale, in-the-wild study is beyond the scope of this paper. We believe the positive experimental results as discussed in the rest of this section are strong evidence of PMSE's effectiveness.

8.1.3 Experiment Procedure

Fig. 3 shows the flow of the test and evaluation processes. Fifty users, who were students of the computer science department, were invited to submit a total of 250 test queries (see [1]) on PMSE. Each user is assigned five test queries of the same topical category, randomly selected from 15 different categories [1]. In the test phase, a user submits a test query and receives the top 100 search results R from the back-end search engine (i.e., Google) without any personalization. The user then clicks on any number of

TABLE 3
Statistics of the Collected Clickthrough Data

# users	50	# queries per user	5
# URLs	25,000	# unique URLs	21,257
# content c_i	31,542	# unique c_i	23,147
# location l_i	173,366	# unique l_i	5,840

results that he/she judges to be relevant to his/her personal interest in much the same way that a standard search engine would have been used.

After the users finished all of the five test queries in the test phase, the training phase begins. The clicked results from the test phase are treated as positive training samples P in RSVM training. The clickthrough data, the extracted content concepts, and the extracted location concepts are employed in RSVM training to obtain the personalized ranking function as described in Section 7.

After the training phase, the evaluation phase is performed to decide if the personalized ranking function obtained in the training phase can indeed return more relevant results for the user. Each user was asked to provide *relevance judgment* on *all* of the top 100 results R for each query he/she has tested in the test phase by grading each result with one of the three levels of relevancy ("Relevant," "Fair," and "Irrelevant"). To this end, the user scans through the full-text of the results using the *preview* function provided by the prototype and then gives relevance ratings to all of the results returned by the search engine. Documents rated as "Relevant" are considered *correct*, while those rated as "Irrelevant" are considered *incorrect* to the user's needs. The ranking of the "Relevant" documents in R and R' is used to compute the *average relevant rank* (i.e., *ARR*, the average rank of the relevant documents, for which a lower value indicates better ranking quality) and *top N precisions* of the baseline and personalized results. Since R' contains the results clicked by the user (i.e., the positive training samples P), their inclusion in precision/ARR computation will unfairly improve the precision/ARR of R' [16]. Thus, P is removed from R' when computing the ARR and top N precisions of R' for fairness. We introduce ARR in this paper to measure the *overall average* performance of the proposed methods in ranking the retrieved documents.

In the experiments, we observed that the average rank of the clicked documents for the baseline method, which composes of the ranked results R (see Fig. 3) returned by the back-end search engine (i.e., Google), was 21.784, and the standard deviation was only 8.934. The low standard deviation shows that the users' click behaviors were quite uniform throughout the five queries assigned to them. The threshold for content concept extraction was set to 0.03. A small mining thresholds was chosen because we want to include as many content concepts as possible in the user profiles. As discussed in Section 4.2, the location concepts were prepared from [2] and [4]. They consist of 18,955 cities in 200 countries. Table 3 shows the statistics of the clickthrough data collected.

8.1.4 Query and User Classes

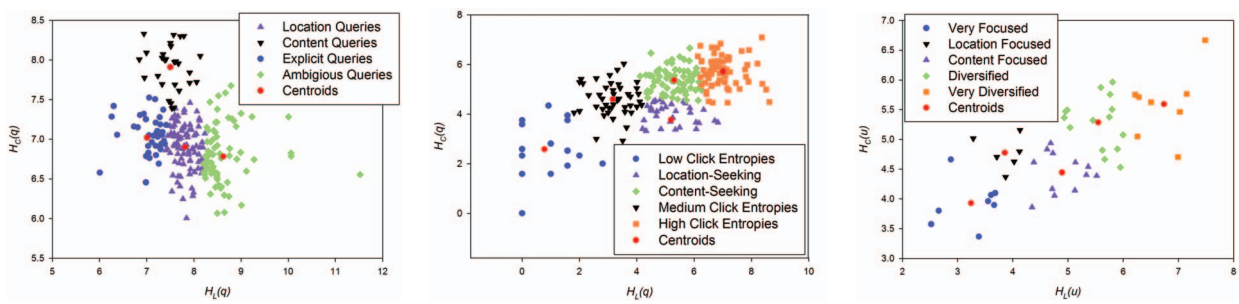
To characterize queries and users with the proposed content and location entropies, we employ K-Means to cluster the queries and users into different classes, and evaluate the performance of PMSE on the different classes. To classify the 250 queries into different classes, we compute their content and location entropies and display them on a scatter plot with location entropy as x-axis and content entropy as y-axis (see Fig. 4a). K-Means is then employed to cluster the queries into four classes, marked with different colors in Fig. 4a. We characterize the four ($H_C(q)$,$H_L(q)$) query classes as follows:

- **Explicit queries.** Queries with low degree of ambiguity, i.e., $H_C(q) + H_L(q)$ is small.
- **Content queries.** Queries with $H_C(q) > H_L(q)$.
- **Location queries.** Queries with $H_L(q) > H_C(q)$.
- **Ambiguous queries.** Queries with high degree of ambiguity, i.e., $H_C(q) + H_L(q)$ is large.

As with the four ($H_C(q)$, $H_L(q)$) query classes, we display the queries on a scatter plot with click location entropy as x-axis and click content entropy as y-axis. Again, the test queries are clustered into five classes with K-Means according to their click entropies. The five ($H_{\overline{C}}(q)$,$H_{\overline{L}}(q)$) query classes are

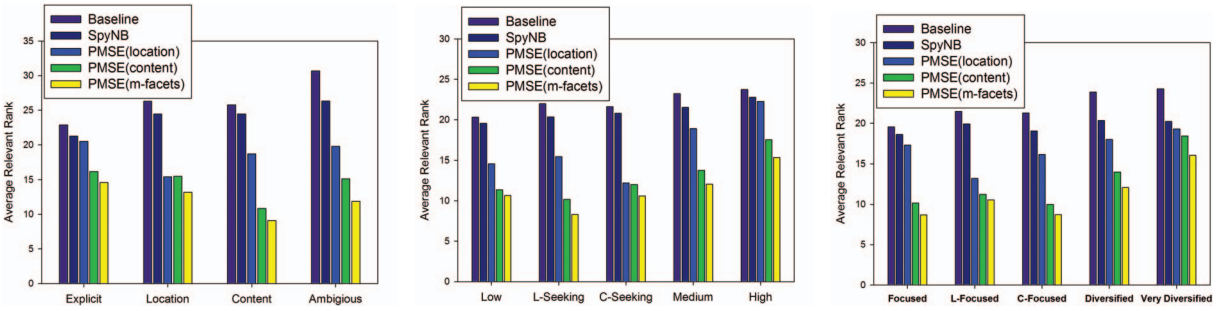
- **Low click entropies.** $H_{\overline{C}}(q) + H_{\overline{L}}(q)$ is small.
- **Content-seeking.** $H_{\overline{C}}(q,u) > H_{\overline{L}}(q,u)$.
- **Location-seeking.** $H_{\overline{L}}(q,u) > H_{\overline{C}}(q,u)$.
- **Medium click entropies.** $H_{\overline{C}}(q) + H_{\overline{L}}(q)$ is intermediate.
- **High click entropies.** $H_{\overline{C}}(q) + H_{\overline{L}}(q)$ is large.

Since the clicks on the test queries are performed by the users after they have read and judged the result snippets



(a) $H_L(q)$ vs $H_C(q)$ (b) $H_{\overline{L}}(q,u)$ vs $H_{\overline{C}}(q,u)$ (c) $H_L(u)$ vs $H_C(u)$

Fig. 4. $H_L(q)$ versus $H_C(q)$, $H_{\overline{L}}(q,u)$ versus $H_{\overline{C}}(q,u)$, and $H_L(u)$ versus $H_C(u)$, and the corresponding clusters.



(a) ARRs for $(H_C(q), H_L(q))$ query classes (b) ARRs for $(H_{\overline{C}}(q), H_{\overline{L}}(q))$ query classes (c) ARRs for different user classes

Fig. 5. ARRs for PMSE, SpyNB, and baseline methods with different query/user classes.

with respect to the relevance of the results to their individual needs, the click entropies can be used as a mean to identify user behaviors. We use the following formula to compute the content/location click entropy of a user ($H_C(u)$ and $H_L(u)$):

$$H_C(u) = \frac{1}{n}\sum_{i=1}^{n} H_{\overline{C}}(q_i, u) \qquad H_L(u) = \frac{1}{n}\sum_{i=1}^{n} H_{\overline{L}}(q_i, u), \quad (20)$$

where $\{q_1, q_2, \ldots, q_n\}$ are the queries submitted by user u. We compute $H_C(u)$ and $H_L(u)$ for each of the 50 users and display the users on a scatter plot with $H_L(u)$ as x-axis and $H_C(u)$ as y-axis.

Again, K-Means is employed to cluster the users into five classes, as shown in Fig. 4c. It is interesting to note that the users are more or less distributed along the diagonal, i.e., a user with diversified/focused location interest also has diversified/focused content interest, and vice versa. The five classes of users are characterized as follows:

- **Very focused.** Users with low content and location entropies. They have very clear topic focuses in the search results, and can be considered as *careful/knowledgeable search engine users*.
- **Content focused.** Users with high location entropy, but low content entropy (i.e., $H_L(u) > H_C(u)$).
- **Location focused.** Users with high content entropy, but low location entropy (i.e., $H_C(u) > H_L(u)$).
- **Diversified.** Users with even higher content and location entropies, and more diversified topical interests.
- **Very diversified.** Users with high content and location entropies. They can be considered as *novice search engine users*, who tend to trust the search engine and click on many results it returns [5].

We provide experimental evaluation of the personalization effectiveness for each user class in Section 8.2.

8.2 Ranking Quality

To evaluate the ranking quality of PMSE, we compare the effectiveness of three alternative PMSE implementations, labeled as *PMSE(content)*, *PMSE(location)*, and *PMSE(m-facets)*, against a baseline approach and the SpyNB method proposed in [15].[4] PMSE (location) employs only the location-based features in personalization, while PMSE (content) uses only the content-based features in personalization. PMSE (m-facets) employs *both* the content-based and location-based features, weighted by their personalization effectiveness (see (18)). The baseline composes of the ranked results returned by the back-end search engine (i.e., Google). We evaluate the effectiveness of different personalization methods using *average relevant ranks*, which is the average rank of the documents rated as "Relevant."

Fig. 5a shows the ARRs of different classes of queries grouped by $H_C(q)$ and $H_L(q)$, as defined in Section 8.1.4. We observe several interesting properties of the baseline method. First, the ARR for the baseline method is low on explicit queries, which is expected to have good performance because they are very focused. Second, it has high ARR for ambiguous queries, showing that the general purpose search engines by design do not handle the ambiguity of queries well. Finally, the ARRs for content and location queries are slightly lower than the ARR on ambiguous queries. The observations show that the commercial search engines perform well for explicit queries, but suffer in various degrees for vague queries.

We observe that PMSE (location) method performs the best on location queries from Fig. 5a, lowering the ARR from 26.28 to 15.11 (43 percent decrease in ARR). It also perform well on ambiguous queries, lowering the ARR from 30.65 to 19.77 (35 percent decrease in ARR). The performance of PMSE (location) method is not good for explicit and content queries, because only a limited amount of location information exists in them. On the other hand, PMSE (content) method performs the best on content queries, lowering the ARR from 25.77 to 10.85 (58 percent decrease in ARR). The ARR is also significantly lowered for ambiguous queries from 30.65 to 15.11 (51 percent decrease in ARR). PMSE (content) performs fine on location queries, because location queries also contain a certain amount of content information. It lowered the ARR of location queries from 26.28 to 15.51 (41 percent decrease in ARR). Finally, as expected, the precisions are the best for explicit queries. However, the improvement is not as significant as in other query classes because the baseline method already performs reasonably well for explicit queries. PMSE (content) lowered the ARR of explicit queries from 22.86 to 16.20 (29 percent decrease in ARR). We also observe that PMSE (content) performs better than PMSE (location) in general, showing that content information is an important factor in the personalization.

4. Note that both SpyNB and SVM can be used for feature extraction in PMSE. Both of them have been evaluated in the preliminary version of this paper. Here, we consider only SpyNB due to the space constraint.

Although PMSE (location) by itself does not perform as well as PMSE (content), it does provide additional improvement for personalization. By employing *both* the content-based and location-based features in the personalization (i.e., PMSE (m-facets)), the ARRs is further lowered. The ARRs of explicit, content, location, and ambiguous queries using PMSE (m-facets) method can greatly reduced to 14.59, 13.19, 9.09, and 11.85, showing that both of the content and location information are useful in the personalization process.

Fig. 5b shows the ARRs of different classes of queries grouped by $H_{\overline{C}}(q)$ and $H_{\overline{L}}(q)$ using different personalization methods. We observe that the baseline method performs the best on queries with low click entropies, but the worst on queries with high click entropies. As discussed in Section 5.3, the higher the click content/location entropies, the worse the personalization effectiveness, because high click content/location entropies indicate that the user has clicked on the search results with high uncertainty. We observe that PMSE (location) method is working the best on Content-Seeking Queries, lowing the ARR from 21.96 to 12.20 (44 percent decrease in ARR), because the user is clicking on the search results with low location entropy for these queries, meaning that the users' location preferences in these queries are highly certain. On the other hand, PMSE (content) method is working the best on Location-Seeking Queries, lowing the ARR from 21.96 to 10.17 (54 percent decrease in ARR), because the user is clicking on the search results with low content entropy, meaning that the users prefer only a small set of content concepts in the search results. Finally, PMSE (m-facets) performs the best on all Low, Location-Seeking, Content-Seeking, Medium, High click entropies, yielding 10.64, 8.32, 10.60, 12.06, and 15.33 ARRs, respectively.

Fig. 5c shows the ARRs of different classes of users grouped by $H_C(u)$ and $H_L(u)$ using different personalization methods. We observe that the baseline method yields high ARR for Very Diversified users, because they have broad interests, and high uncertainty on the search results. On the other hand, the baseline method yields low ARR for Very Focused users because they have very specific needs on the search results. PMSE (location) method performs the best on Location-Focused users (lowing the ARR from 21.48 to 13.19 (39 percent decrease in ARR)), who are focusing on a small set of location concepts. PMSE (content) method performs the best on Content-Focused users (lowing the ARR from 21.27 to 9.98 (53 percent decrease in ARR)), who are focusing on a small set of content concepts. Again, PMSE (m-facets) method performs very well on all types of users, yielding 10.67, 10.56, 8.73, 12.07, and 16.03 ARRs for Focused, Location-Focused, Content-Focused, Diversified, Very Diversified users, respectively. Moreover, it performs better on Focused, Content-Focused, and Location-Focused users (56, 50, and 58 percent decrease in ARRs, respectively), comparing to Diversified and Very Diversified users (49 and 34 percent decrease in ARRs, respectively), conforms with our expectation in Section 5.3 that Focused (either content or location) users are expected to have more significant gain of precisions through personalization compared to the other user classes.

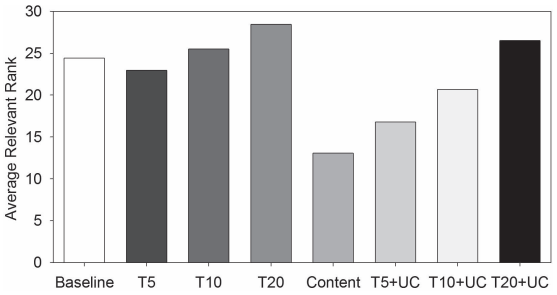


Fig. 6. ARRs for PMSE with top N results as noise.

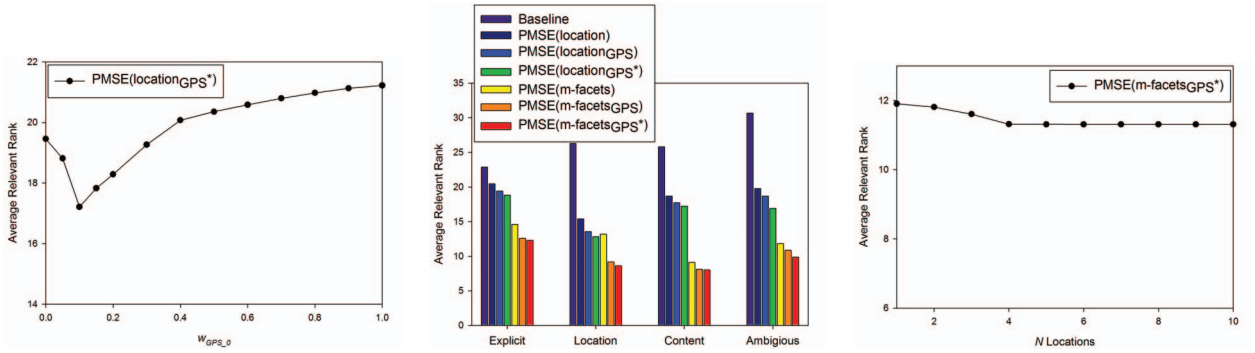
8.3 Top Results as Noise

We assume so far that user clicks truly reflect a user's interest. However, Agichtein et al. showed that search engine users tend to click on top results no matter they are relevant or not [6]. This experiment studies the robustness of PMSE in the presence of *noise*. We assume that the top results are always clicked by the user and study the effect of the noise on the personalized ranking. Fig. 6 shows the ARRs of PMSE methods with different Top N Results as noise. T5, T10, and T20, respectively, treat all top 5, 10, and 20 results as positive samples P in the personalization process, $\mathrm{T5} + \mathrm{UC}$, $\mathrm{T10} + \mathrm{UC}$, and $\mathrm{T20} + \mathrm{UC}$, respectively, treat the top 5, 10, and 20 results *together* with the user clicked results as P in the personalization process, and *Content* is the PMSE (content) method that uses only the content-based features in personalization. We observe that PMSE (content) performs the best because it contains the least *noise* among all methods. For T5, T10, and T20, we observe that they yield similar ARRs as the baseline, and the more top results included as noise, the worse the personalized ranking. On the other hand, when the actual *user clicked* results are included, $\mathrm{T5} + \mathrm{UC}$ and $\mathrm{T10} + \mathrm{UC}$ are always better than the baseline but $\mathrm{T20} + \mathrm{UC}$ yields ARR which is slightly worse than the baseline. We note that it is rare for a user to click on all of the top 20 results. This shows that, in general, PMSE can improve the ranking quality even when noisy clicks exist in the clickthrough data.

8.4 Estimated Combination Threshold $e(q, u)$

In (18), we define $e(q, u)$ to linearly combine the content weight vector $\overrightarrow{w_{C,q,u}}$ and the location weight vector $\overrightarrow{w_{L,q,u}}$. In this section, we evaluate the performance of the estimated $e(q, u)$ by comparing it against the optimal combination threshold $oe(q, u)$. To find $oe(q, u)$ (i.e., the optimal value of $e(q, u)$), we repeat the experiment to find the ARRs for each query by setting $e(q, u) \in [0, 1]$ in 0.05 increments. The $oe(q, u)$ value is then obtained as the value that results in the lowest ARR. Accordingly, we evaluate the retrieval effectiveness of the two combination thresholds ($e(q, u)$ and $oe(q, u)$) by analyzing their top N precisions.

We obtain the average $e(q, u)$ and $oe(q, u)$ values obtained from all queries, and find that the average $e(q, u)$ and $oe(q, u)$ are very close to each other in PMSE (m-facets) ($e(q, u) = 0.4789$ and $oe(q, u) = 0.4754$) method. The average error between them is only 0.1642. Moreover, notice that the combination threshold ($e(q, u)$ and $oe(q, u)$) are close to 0.5, showing that the content preferences $\overrightarrow{w_{C,q,u}}$ and the location preferences $\overrightarrow{w_{L,q,u}}$ are both very important for determining users preferences in personalization.



(a) PMSE(location$_{GPS}$) methods with different w_{GPS_0} (b) ARRs for PMSE methods with/without GPS locations on different query classes (c) ARRs for PMSE(m-facets$_{GPS}$*) with different number of GPS locations

Fig. 7. ARRs for PMSE methods with/without GPS locations at different w_{GPS_0}.

8.5 GPS Locations in Personalization

In this section, we evaluate the impact of GPS locations, as defined in (14) and (16), in PMSE. PMSE (location$_{GPS^*}$) employs only the location-based features which take into account both the location concepts and the GPS locations. The user's GPS locations and locations closely related to the GPS locations receive higher weights in the location weight vector as described in (14) and (16). Fig. 7a shows the ARRs of PMSE (location$_{GPS^*}$) with different initial weights w_{GPS_0} for the decay function as described in (15). We observe that the lowest ARR is achieved when $w_{GPS\_0} = 0.1$. When w_{GPS_0} increases beyond 0.1, the ranking quality degrades, because the ranking has a bias toward the GPS locations, while ignoring the location information extracted from the clickthrough data. In order to optimize the performance of the GPS locations, $w_{GPS\_0} = 0.1$ is used in the following comparisons.

For comparison, we also implement PMSE (location$_{GPS}$) which employs only the location-based features, and only the GPS locations receive higher weights in the location weight vector as described in (14). Fig. 7b shows the ARRs of different methods with/without GPS locations on different query classes. As shown, PMSE (location$_{GPS}$) and PMSE (location$_{GPS^*}$) perform the best on location queries. The ARR of PMSE (location) is 15.41. After including the GPS locations in PMSE (location$_{GPS}$), ARR is further lowered to 13.55 (12 percent decrease in ARR). PMSE (location$_{GPS^*}$) is similar to PMSE (location$_{GPS}$), but it also includes the locations related to the GPS locations using (16). By employing the location ontology with the GPS locations, PMSE (location$_{GPS^*}$) further lowering the ARRs of location and ambiguous queries from 13.55 and 18.70 to 12.85 and 16.91 (5 and 9 percent decrease in ARRs) comparing to PMSE (location$_{GPS}$), showing that the locations related with the GPS locations are also possible candidates that the users may be interested in.

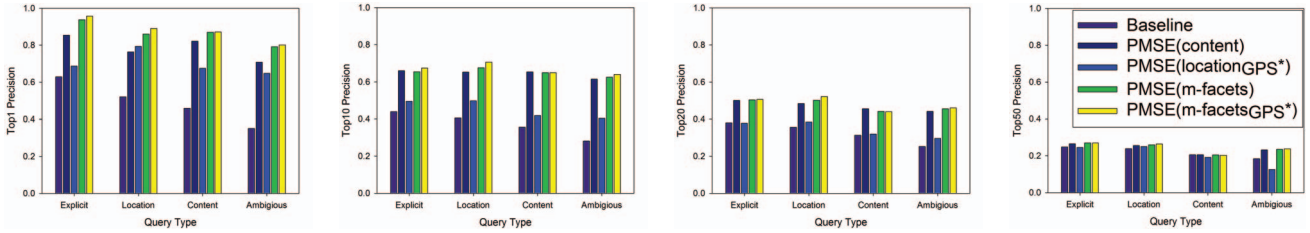
The ARR of PMSE (m-facets) method on location queries is also decreased from 13.19 to 9.18 (30 percent decrease in ARR) after the GPS locations are included as PMSE (m $-$ facets$_{GPS}$) using (14), showing that the GPS locations have a significant impact on location queries. The ARRs of explicit, content, and ambiguous queries are also slight lowered after the GPS locations are included in PMSE (m-facets) method, lowering the ARRs from 14.59, 9.10, and 11.85 to 12.60, 8.11, and 10.86, respectively (14, 11, and 8 percent decrease in ARRs, respectively). PMSE (m-facets$_{GPS^*}$) is the method which also includes the locations related with the GPS locations using (16). Again, PMSE (m-facets$_{GPS^*}$) further lowers the ARRs of location and ambiguous queries from 9.18 and 10.86 to 8.61 and 9.86 (6 and 9 percent decrease in ARRs) comparing to PMSE (m-facets$_{GPS}$), showing that the location ontology is also useful capturing the user preferences on the locations related with the GPS locations.

Fig. 7c shows the ARRs for PMSE (m-facets$_{GPS^*}$) with respect to different number of GPS locations. We observe that the more GPS locations being used, the better the personalization effectiveness (the lower the ARRs). The four most recent GPS locations are the most important ones among all the GPS locations, because the decrease ARRs are obvious with the four most recent GPS locations, while the ARRs remain almost the same even the fifth or more recently GPS locations are included. This shows that the more recent the GPS locations (especially four most recent GPS locations), the higher the chance that the users may be interested in them.

Fig. 8 shows the top N precisions of the compared methods over various query groups. We observe that PMSE (m-facets$_{GPS^*}$) method performs the best among all the methods. By including the GPS locations in the reranking, PMSE (m-facets$_{GPS^*}$) further boosts the top 1, 10, 20, and 50 precisions comparing to the PMSE (m-facets) method. PMSE (m-facets$_{GPS^*}$) method performs the best on location queries, boosting the top 1, 10, 20, and 50 precisions of location queries from 0.5208, 0.4063, 0.3563, and 0.2392 to 0.8902, 0.7063, 0.5222, and 0.2638 (71, 73, 47, and 10 percent in percentage gain). Moreover, it achieves the best precisions among all types of queries, showing that PMSE (m-facets$_{GPS^*}$) can successfully promote the relevant results according to both the user's content and location preferences.

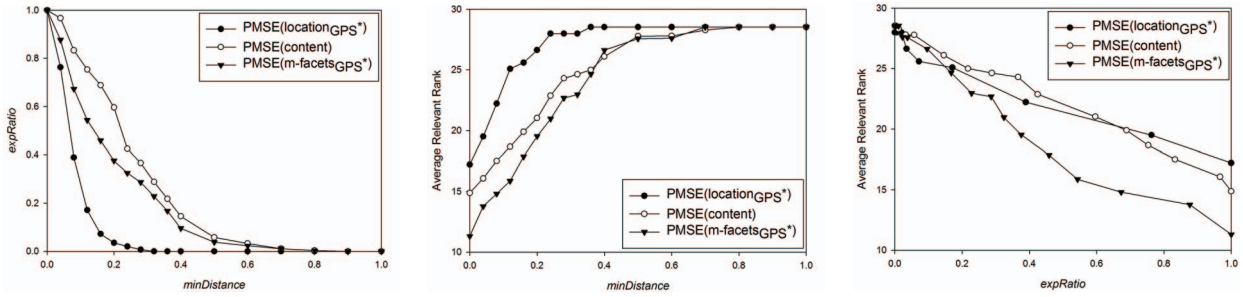
8.6 Privacy versus Ranking Quality

In this section, we evaluate PMSE's privacy parameters, $minDistance$ and $expRatio$, against the ranking quality. We plot $expRatio$ (the amount of private information exposed) against $minDistance$ for a number of PMSE methods in Fig. 9a. The $expRatio$ of PMSE(content), which employs



(a) Top 1 (b) Top 10 (c) Top 20 (d) Top 50

Fig. 8. Top 1, 10, 20, and 50 precisions for PMSE and baseline methods with different query classes.



(a) $minDistance$ vs $expRatio$ (b) $minDistance$ vs Average Relevant Rank (c) $expRatio$ vs Average Relevant Rank

Fig. 9. Relationship between privacy parameters and ranking quality with different PMSE methods.

content ontology only, decreases uniformly from 1 to nearly zero when $minDistance$ increases from 0 to 0.7. $minDistance$ measures the distance of a concept away from the root (i.e., too specific). Since the heights of the trees in the content ontology are mostly less than 0.7, most of the concepts are pruned when $minDistance > 0.7$ in PMSE (content). On the other hand, the $expRatio$ of PMSE (location_{GPS^*}), which employs location ontology only, decreases uniformly from 1 to nearly zero when $minDistance$ increases from 0 to 0.3. The heights of the trees in the location ontology are mostly less than 0.3. We observe that a node in the location ontology can associate many children (e.g., a country has many provinces or states, a province/state has many cities). Once a node is pruned in the location ontology, all the children will also be pruned, thus $expRatio$ decreases much faster than that in PMSE (content). Finally, the $expRatio$ of PMSE (m-facets_{GPS^*}), which employs both content and location ontologies, decreases faster than PMSE (content), but slower than PMSE (location_{GPS^*}). The $expRatio$ of PMSE (m-facets_{GPS^*}) decreases uniformly from 1 to nearly zero when $minDistance$ increases from 0 to 0.6.

We then study the relationships between the privacy parameters and the ranking quality of the search results for PMSE (content), PMSE (location_{GPS^*}), and PMSE (m-facets_{GPS^*}). We plot the ARR of the search results against $minDistance$ in Fig. 9b. As discussed before, the amount of private information exposed ($expRatio$) in PMSE (content) drops uniformly when $minDistance$ increases from 0 to 0.7. Thus, the ARR of PMSE (content) increases uniformly when $minDistance$ increases from 0 to 0.7. Similarly, the ARR of PMSE (location_{GPS^*}) increases uniformly when $minDistance$ increases from 0 to 0.3, and the ARR of PMSE (m-facets_{GPS^*}) increases uniformly when $minDistance$ increases from 0 to 0.6. Finally, Fig. 9c shows the relationships between expRatio and ARR for different PMSE methods. The more privacy information exposed (i.e., higher $expRatio$), the better the ranking quality. We observe that PMSE's privacy parameters produce a smooth increase in ARR when $minDistance$ increases, and a smooth decrease in ARR when $expRatio$ decreases, and thus provide a smooth privacy settings for the users.

9 Conclusions

We proposed PMSE to extract and learn a user's content and location preferences based on the user's clickthrough. To adapt to the user mobility, we incorporated the user's GPS locations in the personalization process. We observed that GPS locations help to improve retrieval effectiveness, especially for location queries. We also proposed two privacy parameters, $minDistance$ and $expRatio$, to address privacy issues in PMSE by allowing users to control the amount of personal information exposed to the PMSE server. The privacy parameters facilitate smooth control of privacy exposure while maintaining good ranking quality. 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 PMSE.

Acknowledgments

The authors would like to express their sincere thanks to the editors and the reviewers for giving very insightful and encouraging comments.

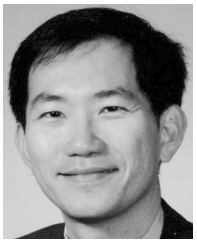
References

[1] Appendix, http://www.cse.ust.hk/faculty/dlee/tkde-pmse/appendix.pdf, 2012.
[2] Nat'l geospatial, http://earth-info.nga.mil/, 2012.
[3] svm^{light}, http://svmlight.joachims.org/, 2012.
[4] World gazetteer, http://www.world-gazetteer.com/, 2012.

[5] E. Agichtein, E. Brill, and S. Dumais, "Improving Web Search Ranking by Incorporating User Behavior Information," *Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR),* 2006.
[6] E. Agichtein, E. Brill, S. Dumais, and R. Ragno, "Learning User Interaction Models for Predicting Web Search Result Preferences," *Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR),* 2006.
[7] Y.-Y. Chen, T. Suel, and A. Markowetz, "Efficient Query Processing in Geographic Web Search Engines," *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR),* 2006.
[8] K.W. Church, W. Gale, P. Hanks, and D. Hindle, "Using Statistics in Lexical Analysis," *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon,* Psychology Press, 1991.
[9] Q. Gan, J. Attenberg, A. Markowetz, and T. Suel, "Analysis of Geographic Queries in a Search Engine Log," *Proc. First Int'l Workshop Location and the Web (LocWeb),* 2008.
[10] T. Joachims, "Optimizing Search Engines Using Clickthrough Data," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* 2002.
[11] K.W.-T. Leung, D.L. Lee, and W.-C. Lee, "Personalized Web Search with Location Preferences," *Proc. IEEE Int'l Conf. Data Mining (ICDE),* 2010.
[12] K.W.-T. Leung, W. Ng, and D.L. Lee, "Personalized Concept-Based Clustering of Search Engine Queries," *IEEE Trans. Knowledge and Data Eng.,* vol. 20, no. 11, pp. 1505-1518, Nov. 2008.
[13] H. Li, Z. Li, W.-C. Lee, and D.L. Lee, "A Probabilistic Topic-Based Ranking Framework for Location-Sensitive Domain Information Retrieval," *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR),* 2009.
[14] B. Liu, W.S. Lee, P.S. Yu, and X. Li, "Partially Supervised Classification of Text Documents," *Proc. Int'l Conf. Machine Learning (ICML),* 2002.
[15] W. Ng, L. Deng, and D.L. Lee, "Mining User Preference Using Spy Voting for Search Engine Personalization," *ACM Trans. Internet Technology,* vol. 7, no. 4, article 19, 2007.
[16] J.Y.-H. Pong, R.C.-W. Kwok, R.Y.-K. Lau, J.-X. Hao, and P.C.-C. Wong, "A Comparative Study of Two Automatic Document Classification Methods in a Library Setting," *J. Information Science,* vol. 34, no. 2, pp. 213-230, 2008.
[17] C.E. Shannon, "Prediction and Entropy of Printed English," *Bell Systems Technical J.,* vol. 30, pp. 50-64, 1951.
[18] Q. Tan, X. Chai, W. Ng, and D. Lee, "Applying Co-Training to Clickthrough Data for Search Engine Adaptation," *Proc. Int'l Conf. Database Systems for Advanced Applications (DASFAA),* 2004.
[19] J. Teevan, M.R. Morris, and S. Bush, "Discovering and Using Groups to Improve Personalized Search," *Proc. ACM Int'l Conf. Web Search and Data Mining (WSDM),* 2009.
[20] E. Voorhees and D. Harman, *TREC Experiment and Evaluation in Information Retrieval.* MIT Press, 2005.
[21] Y. Xu, K. Wang, B. Zhang, and Z. Chen, "Privacy-Enhancing Personalized Web Search," *Proc. Int'l Conf. World Wide Web (WWW),* 2007.
[22] S. Yokoji, "Kokono Search: A Location Based Search Engine," *Proc. Int'l Conf. World Wide Web (WWW),* 2001.

Kenneth Wai-Ting Leung received the BSc degree in computer science from the University of British Columbia, Canada, in 2002, and the MSc and PhD degrees in computer science from the Hong Kong University of Science and Technology in 2004 and 2010, respectively. He is currently a visiting assistant professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His research interests include information retrieval and mobile computing, in particular, search log mining, personalized web search, mobile web search, mobile location search, and collaborative web search.

Dik Lun Lee received the BSc degree in electronics from the Chinese University of Hong Kong, and the MS and PhD degrees in computer science from the University of Toronto, Canada. He is currently a professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. He was an associate professor in the Department of Computer Science and Engineering at the Ohio State University. His research interests include information retrieval, search engines, mobile computing, and pervasive computing.



Wang-Chien Lee received the BS degree from the Information Science Department, National Chiao Tung University, Taiwan, the MS degree from the Computer Science Department, Indiana University, and the PhD degree from the Computer and Information Science Department, the Ohio State University. He is an associate professor of computer science and engineering at Pennsylvania State University. He leads the Pervasive Data Access (PDA) Research Group at Penn State University which performs cross-area research in database systems, pervasive/mobile computing, and networking. He is particularly interested in developing data management techniques for supporting complex queries in a wide spectrum of networking and mobile environments such as peer-to-peer networks, mobile ad hoc networks, wireless sensor networks, and wireless broadcast systems.