Inferring and Using Location Metadata to Personalize Web Search

Paul N. Bennett[1,†], Filip Radlinski[2], Ryen W. White[1], and Emine Yilmaz[3]
[1] Microsoft Research, Redmond, USA; [2] Microsoft, Vancouver, Canada; [3] Microsoft, Cambridge, UK
{pauben,filiprad,ryenw,eminey}@microsoft.com

ABSTRACT
Personalization of search results offers the potential for significant improvements in Web search. Among the many observable user attributes, approximate user location is particularly simple for search engines to obtain and allows personalization even for a first-time Web search user. However, acting on user location information is difficult, since few Web documents include an address that can be interpreted as constraining the locations where the document is relevant. Furthermore, many Web documents – such as local news stories, lottery results, and sports team fan pages – may not correspond to physical addresses, but the location of the user still plays an important role in document relevance. In this paper, we show how to infer a more general location relevance which uses not only physical location but a more general notion of locations of interest for Web pages. We compute this information using implicit user behavioral data, characterize the most location-centric pages, and show how location information can be incorporated into Web search ranking. Our results show that a substantial fraction of Web search queries can be significantly improved by incorporating location-based features.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms
Algorithms, Measurement.

Keywords
Location metadata, personalization, Web search.

1. INTRODUCTION
Web search personalization has recently received tremendous attention from the information retrieval (IR) research community (e.g. [10][21][22][23]). Among the many approaches to personalization, the location of the user has been explored as an implicit feature of search queries (e.g. [12]). Based on the user's location, search engines commonly select the preferred language of results, adapt suggested spelling corrections, and promote search results that are near the user. In particular, location-based personalization has the benefit that it does not require the IR system to have constructed a model of the user in order to adapt search results.

Identifying local search results usually implies that the documents being ranked (or, occasionally documents dynamically created to represent entities such as restaurants or cinemas) must also be associated with a specific geographic location. The distance between the user's location and that of this document can then be taken as a ranking feature when ordering results. Alternatively, if the user provides an explicit location in a query (such as [*pizza new york*]), the location specified in the query provides a reference point from which locations in documents can be measured.

We propose to address location-based personalization in a more general setting. First, consider the geographic sensitivity of a document. Rather than estimating the location of the document from document content, we propose to observe the locations of users who visit this Web page, similarly to Zhuang *et al*. [35]. However, rather than using this to estimate the geographic sensitivity of the query used to reach this document, we simply learn a distribution over user locations for each document, in the form of a density estimate. This density estimate can then be used to recognize the location interest of each document, rather than assuming that each document is relevant to a single location defined at some fixed level of granularity. This approach is beneficial as not all documents that relate to a specific location are equally location sensitive. For example, if a user searches for [*picturehouse cinema*], they are likely interested in a cinema with that name that is within a short drive of their present location. On the other hand, a user who searches for [*disneyland*] is not necessarily interested in the closest Disneyland theme park. We argue that usage statistics, rather than locations mentioned in document content, best represent where a document is most relevant.

Most importantly, this approach is not limited to Web pages that represent clearly localized entities. For instance, consider the online local news section of the Los Angeles (LA) Times newspaper. As expected, Figure 1c shows that this website is of most interest to users located in the greater Los Angeles area. However, as shown in Figure 1d, the crossword section of the same newspaper website is frequently accessed by users from across the United States. This indicates that should a user in Miami issue the query [*la times*], it is relatively more likely that they in fact want the crossword section instead of the local news section desired by a user in Los Angeles. Even more generally, this geographic distribution can tell us that, for instance, service providers' websites are relevant mostly in the areas which they serve, allowing this service area to be directly inferred from usage behavior. For instance, we see in Figure 1a that Sarasota Memorial Hospital serves customers in all of Florida, while the LA Times reviews and recommendations are most relevant in southern California.

In this paper, we present our approach using generalized Gaussian Expectation Maximization to efficiently learn compact density estimates of the distribution of user locations for each relevant Web site. We then show how to use the estimates obtained in this way for both Web documents and search queries to learn a loca-

† Authors are listed in alphabetical order.

SIGIR'11, July 24–28, 2011, Beijing, China.

tion-sensitive IR system, demonstrating that this approach can produce substantially improved search result rankings.

We start by presenting related work in depth. Following this, we describe the data and algorithm used to estimate the location-interest distribution for each Web document. We characterize particularly location-sensitive Web documents and demonstrate that different documents exhibit vastly different location-interest properties. Next, we describe how the document density estimates can be used to infer the likely relevance of documents in response to a query, presenting a learning approach to re-rank documents in a location sensitive manner. Finally, our results show that this approach leads to substantially improved retrieval quality.

2. RELATED WORK

Related work can be grouped into three general areas: (i) research on personalization of search results, (ii) geographic information retrieval, and (iii) inference of Web page locations from various sources such as page content and search engine logs.

Personalized search leverages information about an individual to identify the most relevant search results for that person. A large number of personalization techniques have been proposed in IR research. Some of these methods reside on the server [10] and some on the client [22], some leverage long-term query histories [15][23], and some use short-term implicit feedback [20][21]. A challenge for personalization, especially at Web scale, is in collecting user profiles that are sufficiently rich to be useful in settings such as result ranking, while balancing privacy concerns.

One way that an individual's personalized profile can be augmented is by using data from other people. To better understand whether groups of people can be used to benefit personalized search, Teevan *et al.* [25] explored the similarity of query selection, desktop information, and explicit relevance judgments across people grouped in different ways. They found that some groupings provide valuable insight into what members consider relevant to queries related to the group focus, but that it can be difficult to identify valuable groups implicitly. Building on their findings, Teevan and colleagues show that ranking Web search results based on group leads to a significant relevance gains for group-relevant queries. Along similar lines, Mei and Church [17] proposed a new way to personalize search through back-off based on searcher IP address. They suggest that if there are no relevant data for a particular user, then we should back off to increasingly larger classes of similar users. As a proof of concept, they used the first few bytes of the IP address to define classes and estimated the coefficients of each back-off. In their analysis, Mei and Church examined the effects of backing off based on day-of-week and time-of-day. Our work differs from these personalization methods in that we explicitly use location (rather than implicitly via IP address) and personalize based on the location of the search results, estimated based on usage patterns. Using our approach we can infer that proximal users may have similar information needs. Similar functionality can only be obtained via IP address if they back-off to similar values, but IP addresses can be widely varying, even for proximal users, depending on network connection setup, service provider, and similar factors.

Geographic information retrieval (GIR) addresses the retrieval of documents according to geographic criteria of relevance. Previous GIR research has addressed problems such as the recognition and disambiguation of place references in a text [14], the assignment of documents to encompassing geographic scopes [1], or the retrieval of documents considering geographic relevance [2]. Van Kreveld *et al.* [27] retrieved documents by creating a linear combination of textual and geographic similarity. Purves *et al.* [19] extracted location information from documents and linearly interpolated geographic and text-based retrieval scores in the context of free text ranking. GeoCLEF research (e.g., [13]) has used geographic term expansion on the queries and documents and then conventional term matching algorithms on the resulting expanded texts. Jones *et al.* [12] examined the effectiveness of geographic features of the document, the query, and the document-query combined, and trained a ranker to learn to combine textual and geographic similarity features. They trained a relevance model with both a content-based ranking algorithm and geo-spatial features as inputs, and used the learned weights to predict relevance and perform ranking. For queries with *explicit* place names which they could extract and use as the basis for matching to document mentions, they found that the minimum distance between the document locations and query location is the strongest geographical predictor of document relevance, and that combining geographic features with text features yields a 5% improvement in relevance over using text features alone. Yu and Cai [33] proposed a dynamic document ranking scheme to combine the thematic and geographic relevance measures on a per-query basis. They used query specificity to determine how best to combine different sources of ranking evidence for each query, and demonstrated relevance gains. Research on spatial diversity [24] provides search results that are not only relevant but also spatially diversified so that they are from many different locations. The work presented in this paper differs from previous work in that we do not mine locations from Web page or query content, and do not compute distances based on distance estimates between locations in the query and the content. Rather, we build location-interest models, That is, models of the locations from which users view individual Web documents. We then personalize based on properties of these models and how typical the user's location is for each search result.

There has also been work on detecting and using locations in non-retrieval settings. Mehler *et al.* used locations mentioned in online news articles to detect regional biases toward entities such as players in local sports teams and local politicians [16]. Mei *et al.* [18] use the geography of Weblog authors to model spatial patterns of news topics. Zhuang *et al.* [35] use click information in order to determine whether a search query is geo-sensitive, model and detect, disambiguate, and visualize the associated geographical locations. Wang *et al.* [30] present a method to automatically determine the dominant locations of search queries by mining the top search results and/or query logs. Wang *et al.* [28] tackle the problem of detecting provider, content, and serving location based on content features of Web pages, hyperlinks and queries. In our work, rather than modeling locations based on explicit mentions of location names in logs, we use implicit location information inferred from aggregated locations of users accessing pages.

It is clear from this section that there is a substantial amount of related work in areas similar to that covered by our research. However, our work extends previous research in three key ways. First, we perform personalization based on location metadata, showing how to efficiently and compactly maintain this metadata. Second, we use that personalization metadata for the purpose of re-ranking search engine results. Finally, we infer our location-interest models from log data, estimating the location meta-data to associate with results based on the aggregated location information of those who access the pages, rather than based on content of the results or the query itself. It is worth noting that alt-

hough the framework of inferring metadata to improve rankings that we present in this paper is focused on the use of locations, it generalizes to other problem scenarios where metadata is available for documents (e.g., readability levels).

3. ESTIMATING WEBSITE LOCATION SENSITIVITY

In this section, we present the first stage of our approach, namely efficiently estimating a geographic distribution for each website. In particular, our approach creates a compact model of the locations in which each website is likely of interest. In the following sections, we will then show how this model can be used for information retrieval in particular.

3.1 Data Collection

The primary source of data for this study is a proprietary data set consisting of the anonymized logs of URLs visited by users who consented to provide interaction data through a widely-distributed Web browser add-on. The data set consists of browsing logs (with both Web search and general browsing episodes) consisting of tuples including a random user identifier, the time and date, and the Web page visited. These data provide us with examples of real-world searching behavior that may be useful in understanding and modeling location-based search. Further, each user's IP address is resolved into geographic location information for the user (i.e., city, state, latitude, and longitude) and this geographic location is recorded. All log entries resolving to the same town or city were assigned the same latitude and longitude coordinates. To remove variability caused by cultural and linguistic variation in search behavior, we only include log entries generated by users in the English-speaking United States locale.

The models we construct are based on URL visits during three months from July through September 2010. The evaluation results described in this paper are based on URL visits during the first week of October 2010, representing millions of Web page visits from hundreds of thousands of unique users. From these data we extracted *search sessions* on a commercial Web search engine, using a session extraction methodology similar to [31]. Search sessions begin with a query, occur within the same browser and tab instance (to lessen the effect of any multi-tasking that users may perform), and terminate following 30 minutes of inactivity.

3.2 Location-Interest Models

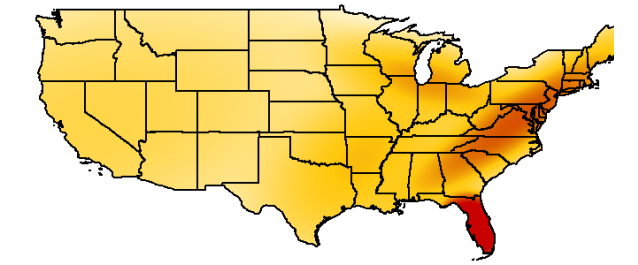
For each URL with more than 50 visits during the three month period used to collect model data, we infer a location-interest model (referred to as *location model* or *distribution* for brevity). This model estimates the probability of the location of the user given they view this particular URL. For compactness, instead of representing each URL by the particular locations from which it was visited, we learn a mixture of Gaussians[1] that can be written:

$$P(location = x|URL) = \sum_{i=1}^{n} w_i \; N(x; \mu_i, \Sigma_i)$$

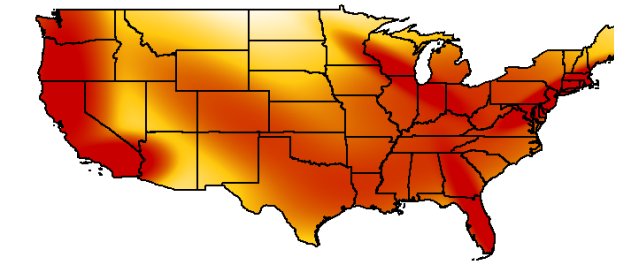
$$= \sum_{i=1}^{n} \frac{w_i}{(2\pi)^2 |\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$$

where $\{\mu_i\}$ and $\{\Sigma_i\}$ are inferred from the data.

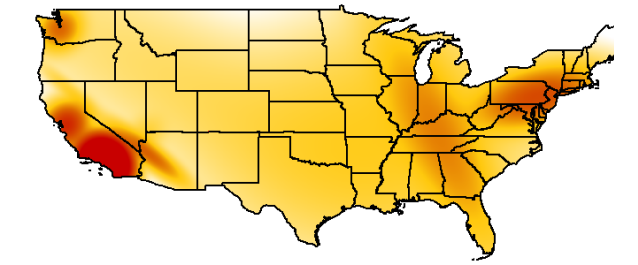
[1] For simplicity, capital "P" is used for (continuous) probability density functions and (discrete) probability mass functions.



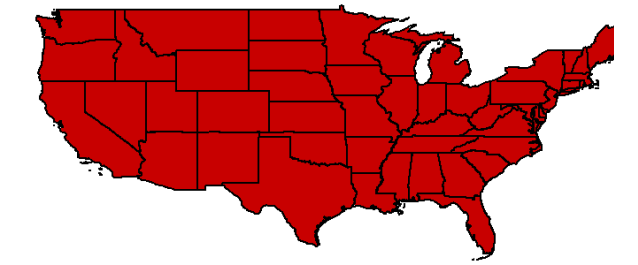
(a) Sarasota Memorial Health, http://smh.com/



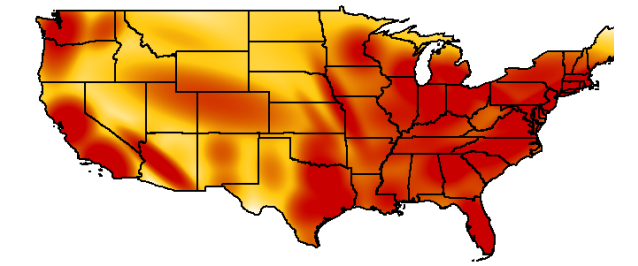
(b) Sydney Morning Herald, http://smh.com.au/



(c) Los Angeles Times: Reviews and Recommendations http://findlocal.latimes.com/



(d) Los Angeles Times: Crossword Puzzles and Games http://games.latimes.com/



(e) Background Model

Figure 1. Example location density estimates. Red indicates higher density, orange and yellow lower density. (a), (b): For two results returned for the query *[smh]* (c), (d): For two results returned for the query *[la times]* (e) Population background model.

Specifically, for a given URL we start with a set of locations {(latitude, longitude)} from which this URL was viewed. To avoid any one user having a large impact on the model, only one location data point is allowed for each user for each day for each URL. Moreover, if a URL has more than 50,000 (user, day) pairs, we uniformly subsample 50,000 location samples.

Using these location data, we learn a density estimate consisting of between 5 and 25 Gaussians (depending on the amount of location data available for each URL) using Expectation Maximization (EM). The EM algorithm specifically adapted to estimating general two-dimensional Gaussians is presented in Algorithm 1, where $f_g(x)$ is the inner term of $P(location = x|URL)$ and $\beta = 0.9$. Intuitively, the algorithm iterates between estimating the probability that each point belongs to each Gaussian (p_{gx}), and estimating the most likely mean, covariance and weight of each Gaussian (μ_g, Σ_g, w_g). The Gaussian locations are initialized at a random observed location, with a high initial variance of 50 degrees in each direction (about 5,500km). As the algorithm progresses, each Gaussian tends to narrow and migrate to a high density area, or broaden to cover a background probability over large geographic areas. Examples of the output of the algorithm can be seen in Figure 1.

In addition, when working at Web scale it is essential to minimize the size of metadata used for ranking. As such, Algorithm 1 merges Gaussians that are too similar (i.e., those with means that are within one degree of each other and with covariance matrices which are also very similar), modifying the EM algorithm (by setting $\beta = 0.9$ instead of the standard value of 1) to encourage Gaussians to be nearby in the E step.

EM exhibits many additional useful properties making it particularly suitable for this setting – such as being efficient at finding a reliable density estimate that dynamically adapts the complexity of the model to the data. The precise mathematical properties of the algorithm are beyond the scope of this paper. We refer the reader to [8] for further details of Gaussian EM and [26] for details about the modification used.

In addition to learning a density estimate per URL, we also learn a general background model describing the density of all users who have opted to provide these interaction data. Aggregating the location information for all URLs yields the background model ($P(location)$). The background model obtained in this way is shown in Figure 1e. From the figure, it can be clearly seen that this model is reasonably representative of the population distribution in the United States.

Finally, we also learn a location-interest model for each query. For each distinct query observed in our data, we take the locations of the users who issued this query, and use Algorithm 1 in exactly the same way as for URLs. This provides an estimate of $P(location|query)$.

Algorithm 1: Generalized Gaussian EM.

1. $X \leftarrow$ location data; $n \leftarrow$ initial number of Gaussians;
2. For each Gaussian $g_i \in G$ *[Initialize model]*
   1. $\mu_i =$ distinct random point $x \in X$
   2. $\Sigma_i = \begin{bmatrix} 50^2 & 0 \\ 0 & 50^2 \end{bmatrix}$
   3. $w_i = \frac{1}{n}$
3. For iteration = 1 to 10
   1. Until convergence
      1. For every $g \in G, x \in X$ *[E step]*

$$p_{gx} = \frac{\left(w_g f_g(x)\right)^\beta}{\sum_{g' \in G} \left(w_{g'} f_{g'}(w)\right)^\beta}$$

      2. For every $g \in G$, update parameters concurrently *[M step]*
         1. $\mu_g = \frac{\sum_x p_{gx}\, x w_g}{\sum_x p_{gx} w_g}$
         2. $\Sigma_g = \frac{\sum_x p_{gx}(x-\mu_g)(x-\mu_g)^T w_g}{\sum_x p_{gx} w_g}$
         3. $w_g = \frac{\sum_x p_{gx} w_g}{|X|}$
   2. For every $g, g' \in G$ *[Merge near-duplicate Gaussians]*
      If g and g' are too close, merge g and g'

4. LOCATION SENSITIVE FEATURES

Given the location model generated for each URL, as well as for each query and the background model for the entire population, we can now leverage these models for personalized search. We now describe the features we use to represent geographic locality of search results.

4.1 Non-Contextual Features

The first class of features we investigate involve characterizing the query and results without considering the specific user. For example: Is the query issued location sensitive? Are these URLs location sensitive? These are likely to act as indicators as to when location should be taken into account by a ranker.

4.1.1 Features of the URL alone and the query alone

Let M_u be the location model for a given URL u, and M_{bg} be our background model. Our URL features include the aggregate popularity of the URL (n_{URL}, the number of distinct (user, day) pairs observed), as well as the overall entropy of the location distribution and its KL divergence from the background.

As computing the entropy of a mixture of Gaussians exactly is intractable, we computed the entropy by sampling from the location distribution of the URL:

$$Entropy(M_u) = E_{loc}\left[-\log\left(P(loc|M_u)\right)\right] \approx \langle -\log(P(loc|M_u)\rangle$$

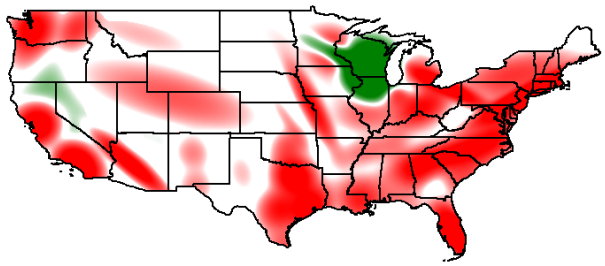
where loc is a location drawn from M_u and $\langle f \rangle$ represents an empirical mean of f across many samples. The KL divergence between the location model for u and the background location model is defined as:

$$KLScore(u, background) = KL\left(M_u \,||\, M_{bg}\right) = \int_{loc} P(loc|M_u) \log\left[\frac{p(loc|M_u)}{p\left(loc|M_{bg}\right)}\right] d\, loc$$

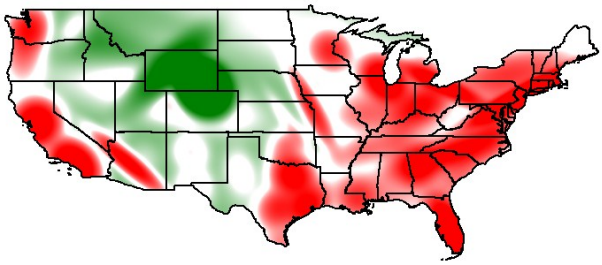
We computed the KL divergence both using sampling for the entropy (*KLScore-Sam*), and using a simple variational upper bound (*KLScore-Var*) [9].

Finally, we also compute the mean width of each URL model, $ModelWidth(u)$, by sampling from the distribution and computing the mean distance from the sampled mean of the distribution. Figure 1a shows an example of a low width model, and Figure 1b shows an example of a high width model.

The same features were also computed for each query.



(a) University of Wisconsin homepage



(b) University of Wyoming homepage

Figure 2. *NormLocUrl* for two websites as a function of location. Green (red) indicates that the URL is more (less) likely than predicted by the background (and also non-trivial).

4.1.2 Features of (URL, query) pair

We also compute the KL divergence between the URL and query location distribution, $KL(URL||Query)$, again both using sampling and the variational upper bound. Intuitively, if a query and URL have a very similar distribution, with low KL divergence, we would expect the URL to more likely be relevant to users who issue this query.

4.2 Contextual Features

Contextual features take into account the user's particular location, and we expect them to be particularly important in personalizing search results. We used the following contextual features:

4.2.1 Features of the user

The user's location (latitude, longitude) is included as a feature.

4.2.2 Features of the (user, URL) pair

The simplest interesting contextual feature is the probability of the user's location given the URL u, $P(loc|M_u)$, estimated by evaluating the URL location model at the user's location. We call this feature *LocUrl*. If this user is in a location where this URL is popular, the feature would be high.

If the personalization model were a perfect estimator of the location distribution of the URL, and location was the only determining feature (i.e., there were no query), the best we could do would be to rank by $P(u|loc)$, the probability of the URL u given the user's location. Using Bayes rule, we can estimate this quantity as follows:

$$P(u|loc) = \frac{P(loc|u)\,P(u)}{P(loc)}$$

Given that the ranking task involves ranking URLs for a user in a particular location, we can ignore the $P(loc)$ term. $P(u)$ can be estimated from the frequency with which this URL was viewed overall. Hence, we use:

$$UrlLoc(u, loc) = n_{URL}\,P(loc|M_u)$$

However, relying on $P(loc|M_u)$ for our features suffers from every large population center having a higher probability of location for all URLs. As such, when training a ranker, this feature will always be large when the user is in a high population region, and always small otherwise. To obtain more useful ranking scores, we also use a normalized probability of location given URL, subtracting the background model:

$$NormLocUrl(u, loc) = \frac{LocUrl(u, loc)}{LocUrl(background, loc)}$$

As an illustration of this feature, consider **Error! Reference source not found.**. It shows the *NormLocUrl* as a function of location for two universities both returned for the query *UW*, namely the University of Wyoming, and the University of Wisconsin. We see that the popularity of the University of Wyoming relative to the background frequency is higher over a larger geographic area than that of the University of Wisconsin. Note that higher popularity relative to the background over a large region does not necessarily imply greater popularity in terms of number of users: the University of Wyoming is popular in less populated areas than the University of Wisconsin.

In addition, we implemented two variants of this feature. The first (*NormLocUrl-Thresh*) thresholds the normalized feature, setting it to 1 whenever the above ratio is less than 1, i.e., whenever the user location is less likely under the URL model than under the background model. This has the effect of emphasizing URLs that are more likely for this query. The second variant (*NormLocUrl-Renorm*) renormalizes the background model so that it sums to 1 over the area where $P(loc|URL) > \epsilon$, for a small ϵ. This in effect allows the background normalization to only take into account the population distribution where this URL has ever been clicked, avoiding URLs that are of limited geographical interest having feature values much larger than URLs that are of broad geographical interest.

Finally, we compute general properties of the URL distribution in the context of the user: (1) *TotalVolume(u,loc,d)*, the percent of the URL u probability mass within a particular distance d of the user location; (2) *DistMean(u,loc)*, the distance from the user's location to the mean of the URL model; (3) *PeakDist(u,loc)*, the distance from the user's location to the nearest individual Gaussian component of the URL model as well as the weight of this Gaussian in the model (*PeakWeight*). These features attempt to capture features of the neighborhood of the URL location model close to the user.

4.2.3 Features of the (user, query) pair

We also computed exactly the same features taking the query location model instead of the URL location model, naming them equivalently (e.g., *LocQuery* instead of *LocUrl*). These represent how typical the user location is of this query.

4.3 Standard Ranking Features

As our experiments will involve learning a re-ranking of Web results that take user location into account, we also incorporate relevance of the URL to the user's query in the form of two simple features:

1. The rank of the URL in the non-personalized results returned by an underlying ranking function of the Bing search engine.
2. The score of this (query, URL) pair as produced by Bing (monotonically decreasing with the rank of the URL).

5. EXPERIMENTAL METHOD

Having described our data and features in the preceding sections, we now detail the evaluation of our method. We start with an analysis of the properties of our dataset, followed by quantitative experimentation on learning to rank.

5.1 Evaluation Dataset Construction

From the week-long sample of search sessions described in Section 3.1, we generate a dataset for our re-ranking experiments. The dataset comprises a set of approximately one million queries selected uniformly at random from the search sessions. For each query, the top ten search results retrieved by the Bing Web search engine were included, along with the latitude and longitude of the user, any location models available for each of the top ten search results, and the location model built for the query, if available.

For evaluation, we need a personalized relevance judgment for each result. Obtaining many relevance judgments from real users in a wide range of geographic locations is impractical, and there is no known approach to train expert judges to provide reliable location-sensitive judgments that reflect real user preferences. Hence we obtained these judgments using a log-based methodology inspired by [7]. Specifically, we assign a positive judgment to one of the top 10 URLs if it is the last satisfied result click in the session. We define a satisfied result click in a similar way to previous work [23][29], as either a click followed by no further clicks for 30 seconds or more, or the last result click in the session. The remaining top-ranked URLs receive a negative judgment. This gives us one positive judgment and nine negative judgments for each of the top-10 URLs for each session.

The rank position of this single positive judgment is used to evaluate retrieval performance before and after re-ranking. Specifically, we will measure our performance using the inverse of the rank of the relevant document, otherwise known as the mean reciprocal rank (MRR). Queries for which we cannot assign a positive judgment to any top-10 URL are excluded from the evaluation dataset. We also exclude queries for which we cannot assign a location model to at least one of the top-10 results (approximately 16% of queries were removed in this way), or where one of several high precision rich graphical results is shown (for example detailing information about a celebrity, corresponding to approximately 2% of remaining queries). Note that this means that up to nine of the URLs may not have a location model, and thus may have zero values for all location features. Additionally, the query may or may not have a location model depending on the query frequency during the previous three months.

As the correctness of our evaluation relies on the assumption that promoting satisfied click documents improves overall relevance, we consider these labels further. Previous work on inferring relevance from clicks has shown that assuming clicks to indicate relevance does provide reliable evaluation metrics (e.g., [29]). Further, similar models have also been used to infer relevance directly from clicks (e.g., [6][34]). Moreover, even if the assumption that results lacking satisfied clicks are non-relevant does not hold, promoting the satisfied clicked results is still likely to improve relevance by displacing higher ranked skipped results, with skipping followed by clicking having been reliably shown to indicate that the skipped result is less relevant [11]. More generally our labeling method can be considered studying the task of predicting only the *most* relevant item. Furthermore, when we consider all satisfied clicks, 97% of the queries in our evaluation set have two or fewer satisfied clicks. Thus, considering all satisfied clicks does not change a large proportion of the data. Finally, it is also worth noting that our approach does not require training data to come from documents labeled in this way. If personalized relevance judgments were available, our model could be trained from such judgments without any modifications.

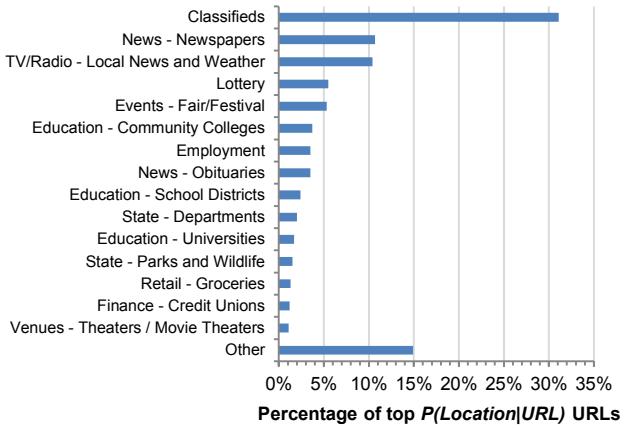


Figure 3. Distribution of topics in most location-centric URLs.

5.2 Labeling Location-Centric Pages

We next studied our dataset to understand which of the visited pages in our sessions are most location centric, and to better understand the nature of these pages.

We first ranked all URLs observed in our week-long evaluation dataset in descending order based on average log-likelihood of the user's actual location given the URL. After we removed pages consisting of search engine result lists and a small number of obviously non-local other URLs (e.g., online gaming sites), the top 750 remaining URLs were manually labeled by one of the authors of this paper, who created a taxonomy to categorize the most location-focused URLs based on their subject matter. This process involved visiting each URL and assigning it to an existing category or creating a new category as appropriate. We iterated and refined the taxonomy, ending up with 56 distinct labels on a broad range of location-centric topics ranging from classifieds to education. In Figure 3, we show the distribution across the 15 most popular categories, capturing 85% of the URLs in the labeled set. Remaining URLs were grouped in the category *Other*.

The figure shows that *Classifieds*, *News* (e.g., online versions of local newspapers, local television and radio stations, obituaries), and *Education* (community/technical colleges, school districts, smaller universities or outreach campuses of larger universities, grade tracking, student portals) are among the most location-centric URLs. Those labeled as *Other* in Figure 3 included pages associated with *Justice* (criminal records, court cases, inmate searches), *Property* (appraisals, auditors), *Transit* (public, traffic, tolls), *Utilities* (power or communication companies) and *Government* (city or state homepage). It is clear from this analysis that $P(location|URL)$ finds pages with a clear local intent.

Interestingly, URLs classified as *Retail* were usually associated with items that shoppers would typically be expected to purchase in person – such as furnishings, groceries, and medication. Stores selling electronics or other items that could easily be obtained online did not emerge in our analysis as strongly location centric.

Also, we observe that knowing the address associated with the URL does not always equate to knowing the locations from which

Table 1. Summary learning results, split between navigational and other queries as well as by click entropy (CE).

Queries			Change in MRR	Fraction of queries
All			1.9	100%
Navigational	All		3.5	34%
	CE Sample	All	4.5	1.2%
		CE > 3	7.2	3.2%
		1 < CE < 3	4.6	63.8%
		CE < 1	3.1	33.0%
Other	All		1.1	66%
	CE Sample	All	1.4	1.8%
		CE > 3	4.5	22.8%
		1 < CE < 3	4.2	55.7%
		CE < 1	2.7	21.5%

people would want to access that URL. One solution to identifying locations from URLs would be to simply extract addresses directly from Web page content, and use these to build the location model for the page. For example, from parsing Web pages, we would establish that the Massachusetts Institute of Technology (MIT) is located in Boston, MA. However, that does not tell us whether or not only people located in the Boston area would want to access the MIT page. The website of the university Boston College, which is less than five miles from MIT, has a much different location-interest profile with a much higher proportion of visits being from local users rather than distant users. This demonstrates the value of our method for inferring location metadata for URLs from usage patterns (rather than page content).

5.3 Learning to Rank

We next turn to the motivating task of personalizing Web search results based on the user location. For all the labeled rankings observed during the week of training, we compute all the features described in Section 4. From this dataset we then subsampled approximately half a million queries by uniformly randomly choosing one query per session. This was done to avoid giving extra importance to long sessions where the same user location would be seen repeatedly. The queries were then partitioned into ten parts in order to conduct ten-fold cross validation. For each fold, 10% of the training set is used as a validation set for model selection. All results presented below are the means of performance on the ten folds.

Using this dataset, we train a ranking model using the LambdaMART learning algorithm [32] for re-ranking the top ten results of the query. LambdaMART is an extension LambdaRank [3] based on boosted decision trees. LambdaMART has recently been shown to be one of the best algorithms for learning to rank. Indeed, an ensemble model in which LambdaMART rankers were the key component won Track 1 of the 2010 Yahoo! *Learning to Rank Challenge* [4]. In our experiments, we use LambdaMART with 500 decision trees. However, we also note that the choice of learning algorithm is not central to this work, and any reasonable learning to rank algorithm would likely provide similar results.

Our baseline is the original ranking of the top-10 provided by the Bing search engine, presenting a very competitive baseline. Because of the proprietary nature of its performance, we do not report absolute MRR, but instead we report the change in MRR value in the scale of 0 to 100, i.e., $100 \times \big(MRR(learned) - MRR(baseline)\big)$.

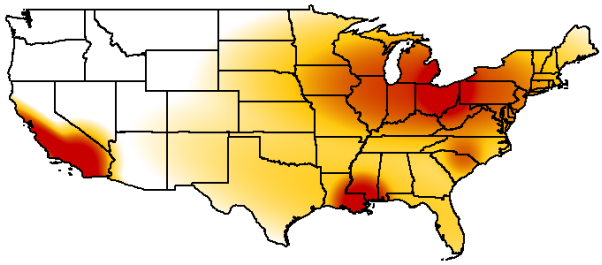


Figure 4. Query location model for the query [*rta bus schedule*]. We see three peaks: in Ohio, Louisiana, and California.

6. RESULTS AND DISCUSSION

We now present results from our learning experiments, both in terms of ranking performance, and analyzing the impact of the different classes of features proposed.

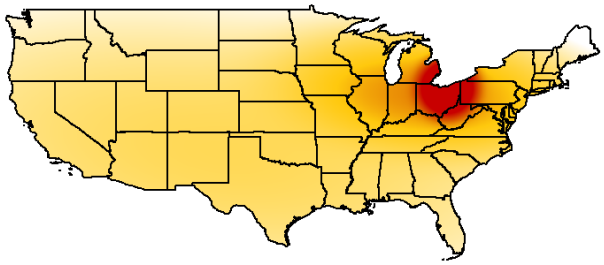
6.1 Ranking Performance

Table 1 shows the summary results for LambdaMART versus the baseline ranker performance. In the first row, we see that the learned model improves by 1.9 (on a scale of 0 to 100) over the baseline ranker in terms of MRR. All the improvements in the table are statistically significant with 95% confidence according to the Wilcoxon sign-rank test. The learned model changes the position of the relevant item for 16.8% of the queries and improves 10.4% of the queries. This shows that the location of the user is important for a substantial fraction of Web search queries. The learned model boosts the position of the relevant item in 61.8% of queries where the relevant item's position changes. On average, over the queries where the relevant item shifts, the learned model boosts the relevant item by 0.54 positions. In other words, every other query experiences an improvement of about one position in rank. Given the importance of the first position on user satisfaction, it is worth considering impact on that position separately. The learned model moves a relevant item out of position one 2.3% of the time and moves a relevant item into position one 4.3% of the time.

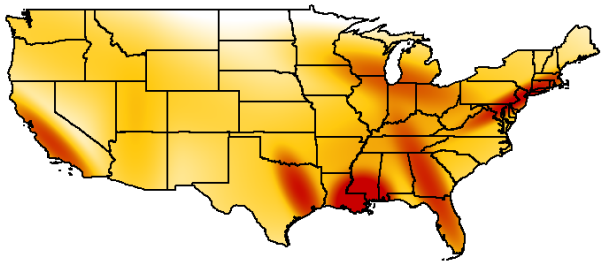
6.2 Effect of Query Type

Next, we break down our results by separating out navigational queries. We define navigational queries as queries that are particularly frequent and where one popular URL dominates user clicks. In our dataset, 34% of the queries are marked as navigational in this way. We find more substantial improvements for navigational queries than the remaining queries. This is particularly interesting as most clicks on the navigational queries are the same for all users, suggesting that some of the improvements are on queries issued predominantly in a confined geographic area, where the original ranker is not taking this location into account.

As an example of this, consider the query [*rta bus schedule*]. In our dataset, this query was issued by a user in New Orleans in the state of Louisiana. Figure 4 shows the location distribution of this query. We see that it is most frequently issued in Southern California, in Ohio and in Louisiana. The top result returned by the baseline system for this query was most relevant in Ohio, as can be seen in Figure 5a. However, in this case the user clicked on an



(a) http://www.riderta.com/maps-schedules.asp



(b) http://www.norta.com/

Figure 5. User location model for (a) the top original result for the query [*rta bus schedule*], and (b) for the result eventually clicked by a particular user in New Orleans, LA.

appropriate result for Louisiana, with the location model shown in Figure 5b. The original ranking for this query, as well as the re-ranked results produced by our learned model are shown in Table 2. For each of the original top ten results, we see the URL, our estimate of the most relevant location (through manual inspection of the page), the approximate location of the largest Gaussian peak in the URL model, as well as the distance between the model peak and the manually determined URL location. We see that overall the URL models reflect the true page location reasonably well. In the case of the user in New Orleans, we also see that the correct result was moved from position 8 to position 2, resulting in a large improvement in the quality of the results for this particular user.

Returning to the summary results in Table 1, we show the results for navigational and other queries broken down further. In particular, we study improvements as a function of query click entropy (CE) [5] over a sample of queries with sufficient frequency to estimate click entropy. This sample constituted 3% of our data. The query click entropy measures the distribution of URLs previously clicked by users, where a high value indicates that many different URLs are frequently clicked by different users, while a low value indicates that the same URL is clicked reliably. Note that the values of click entropy we report have been scaled by a factor of 50 to simplify presentation.

In general, navigational queries are expected to have low click entropy as there is usually one destination URL that is clicked by most users. A navigational query with high click entropy is likely to be affected by user location: the destination URL is different for users from different locations, hence different URLs are clicked by different users. Thus, one would expect the learned model to achieve higher gains for these queries. As expected, we see that queries where many different URLs are frequently clicked show the largest improvement in performance due to location-based personalization.

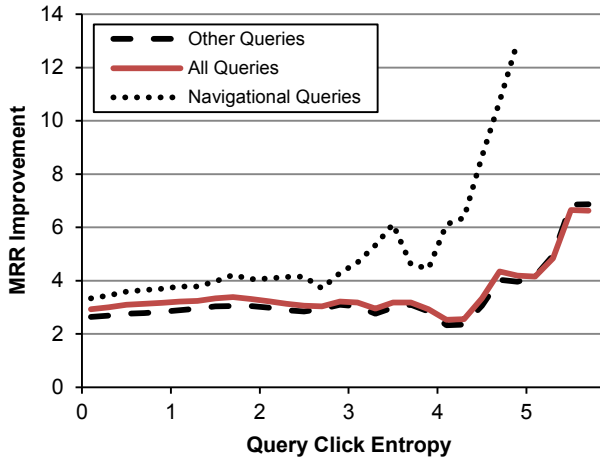


Figure 6. MRR improvement as a function of query click entropy for frequent queries.

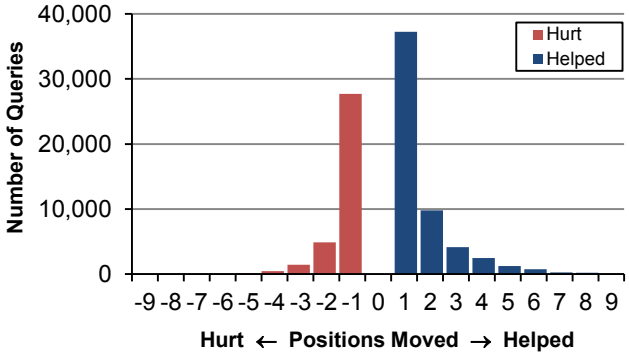


Figure 7. Positions by which clicked item in evaluation data moved up or down due to personalization.

Considering the click entropy in more detail, Figure 6 shows the MRR improvement of the learned model for the same query types as a function of click entropy. Overall, the performance numbers are higher than for all queries due to this analysis being limited to queries with known click entropy. We see that the small fraction of navigational queries improve more at all click entropy levels, but that click entropy is a good indicator of potential for personalization (as also noted by [5]). However, we also see that even queries with little or no variation in the URL clicked (CE $\approx$ 0) benefit from our approach.

As a further analysis, we measure how often location based personalization hurts or helps on a per query basis, across all queries. Figure 7 shows the number of queries for which the satisfied clicked result moved up, or down, by the given number of positions. We see that ranking changes are most often of one or two positions, with improvements substantially more frequent than degradation of performance. We also note that results are sometimes promoted by more than 5 positions, moving search results to substantially more prominent positions. Such high impact changes have a more substantial effect on user satisfaction.

Table 2. Learned re-ranking of results for the query [*rta bus schedule*] from a user in New Orleans, LA. The user's last satisfied click was on the shaded item, which has moved from rank 8 in the baseline to rank 2 in the personalized ranking.

New Rank	Top 10 Ordered by New Ranking	Actual Location (of resource on page)	Most Probable Inferred Peak	Orig. rank	Distance (inferred to actual)
1	http://www.riderta.com/maps-schedules.asp	Cleveland, OH	Ohio (South of Cleveland)	1	25 km
2	http://www.norta.com/	New Orleans, LA	New Orleans, LA	8	0 km
3	http://www.riderta.com/	Cleveland, OH	Ohio (South of Cleveland)	3	25 km
4	http://www.riversidetransit.com/home/index.htm	Riverside, CA	California (East of Riverside)	2	55 km
5	http://www.rtachicago.com/	Chicago, IL	Chicago, IL	4	0 km
6	http://norta.com/routes/	New Orleans, LA	New Orleans (Lake Pontchartrain)	6	5 km
7	http://www.riversidetransit.com/bus_info/schedules.htm	Riverside, CA	California (East of Riverside)	5	55 km
8	http://slorta.org/	San Luis Obispo, CA	California (Northwest of San Luis Obispo)	9	10 km
9	http://www.cctexas.com/?fuseaction=main.view&page=2847	Corpus Christi, TX	*Insufficient data*	10	–
10	http://gunnisonvalleyrta.org/	Gunnison, Co	Colorado (Between Denver and Gunnison)	7	170 km

6.3 Feature Analysis

Next we turn to analysis of the features contributing to the results in the previous section.

In our ranking model, we found that the most important feature is the initial rank determined by the original ranking function. This may be because the location features do not depend on the match between the query and the URL. The second most important feature is *UrlLoc*, which estimates the probability of the URL given the user's location. This suggests that the feature is indeed providing a reasonable estimate of the utility of each URL at each user location. We also saw that URL popularity plays a strong role, followed by the KL divergence of the URL model from the background model. This indicates that URLs whose popularity merely mirrors the background distribution are less likely to be good candidates for promotion or demotion based on user location.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an approach for inferring the locations of interest for URLs based on user browsing behavior. We showed that these models are more informative than content alone. We showed how locations can be efficiently encoded as a Gaussian model describing the probability of the location given a URL, and further how this basic model can be transformed into a number of informative features. We demonstrated that these features allow location based personalization of search results, leading to significant gains in offline evaluation, changing the position of the relevant item in 16.8% of the queries, improving it for 10.4% of queries, and improving overall MRR by 1.9%.

Natural next steps include a comparison with content-based methods, and further validation of our approach in an online setting by dynamically re-ranking search results, and evaluating with an appropriate online metric. Additionally, although 84% of queries return results where at least one in the top-10 has a location model, smoothing approaches could allow location information to be shared between related URLs and allow this approach to be extended to URLs which are visited less frequently or are entirely new. However, this work also suggests broader applications. Location-based personalization is applicable beyond standard Web search, also encompassing advertising, product recommendation, and social networking. Similar models can also be constructed of locations of interest to individual user or specific user cohorts.

8. ACKNOWLEDGMENTS

We would like to thank Nick Craswell and Tom Minka for helpful comments and advice on this research.

REFERENCES

[1] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. (2004). Web-a-where: geo-tagging web content. In *Proceedings of the ACM SIGIR Conference on Research and Development on Information Retrieval*, pp. 273–280.

[2] L. Andrade and M. Silva. (2006). Relevance ranking for geographic IR. In *Proceedings of the ACM SIGIR Workshop on Geographic Information Retrieval*.

[3] C.J.C. Burges, R. Ragno, and Q.V. Le. (2006). Learning to rank with non-smooth cost functions. *Advances in Neural Information Processing Systems*, pp. 193–200.

[4] O. Chapelle, Y. Chang, and T.-Y. Liu. (2010). The Yahoo! Learning to Rank Challenge. *http://learningtorankchallenge.yahoo.com.*

[5] Z. Dou, R. Song, and J.R. Wen. (2007). A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the International Conference on the World Wide Web*, 581–590.

[6] G. Dupret and B. Piwowarski. (2008). A user browsing model to predict search engine click data from past observations. In *Proceedings of ACM SIGIR Conference on Research and Development on Information Retrieval*, pp. 331–388.

[7] J. Gao, W. Yuan, X. Li, K. Deng, and J.-Y. Nie. (2009). Smoothing clickthrough data for web search ranking. In *Proceedings of ACM SIGIR Conference on Research and Development on Information Retrieval*, pp. 355–362.

[8] T. Hastie, R. Tibshirani, and J. Friedman. (2001). *Elements of Statistical Learning*. Springer.

[9] J.R. Hershey and P.A. Olsen (2007). Approximating the kullback leibler divergence between gaussian mixture models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

[10] G. Jeh and J. Widom. (2003). Scaling personalized web search. In *Proceedings of the International Conference on the World Wide Web*, pp. 271–279.

[11] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. (2007). Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems*, 25(2):7.

[12] R. Jones, A. Hassan, and F. Diaz. (2008). Geographic features in Web search retrieval. In *Proceedings of ACM GIR Workshop on Geographic Information Retrieval*, pp. 57-58.

[13] Y. Li, N. Stokes, L. Cavedon, and A. Moffat. (2006). NICTA I2D2 group at GeoCLEF 2006. In *Proceedings of Cross-Language Evaluation Forum*, pp. 938–945.

[14] B. Martins, I. Anastacio, and P. Calado. (2010). A machine learning approach for resolving place references in text. In *Proceedings of the AGILE Conference on Geographical Information Science*.

[15] N. Matthijs and F. Radlinski. (2011). Personalizing Web search using tong term browsing history. In *Proceedings of the ACM WSDM Conference on Web Search and Data Mining*, pp. 25–34.

[16] A. Mehler, Y. Bao, X. Li, Y. Wang, and S. Skiena. (2006). Spatial analysis of news sources. *IEEE Transactions on Visualization and Computer Graphics*, 12(5): 765–772.

[17] Q. Mei and K. Church. (2008). Entropy of search logs: how hard is search? with personalization? with backoff? In *Proceedings of ACM WSDM Conference on Web Search and Data Mining*, pp. 45–54.

[18] Q. Mei, C. Liu, H. Su, and C. Zhai. (2006). A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of the International Conference on the World Wide Web*, pp. 533–542.

[19] R. Purves, P. Clough, C. Jones, A. Arampatzis, B. Bucher, D. Finch, G. Fu, H. Joho, A. Syed, S. Vaid, et al. (2007). The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the internet. *International J. of Geographical Information Science*, 21(7).

[20] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. (2005). In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 239–248.

[21] X. Shen, B. Tan, and C. Zhai. (2005). Context-sensitive information retrieval using implicit feedback. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–50.

[22] X. Shen, B. Tan, and C. Zhai. (2005). UCAIR: A personalized search toolbar. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 681–681.

[23] B. Tan, X. Shen, and C. Zhai. (2006). Mining long-term search history to improve search accuracy. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining,* pp. 718–723.

[24] J. Tang and M. Sanderson. (2010). Spatial diversity, do users appreciate it? In *Proceedings of the ACM GIR Workshop on Geographic Information Retrieval*.

[25] J. Teevan, M.R. Morris, and S. Bush. (2009). Discovering and using groups to improve personalized search. In *Proceedings of the ACM Conference on Web Search and Data Mining*, pp. 15–24.

[26] N. Ueda and R. Nakano. (1998). Deterministic annealing EM algorithm. *Neural Networks*, 11(2), pp. 271–282.

[27] M. J. van Kreveld, I. Reinbacher, A. Arampatzis, and R. van Zwol. (2005). Multi-dimensional scattered ranking methods for geographic information retrieval. *GeoInformatica*, 9(1): 61–84.

[28] C. Wang, X. Xie, L. Wang, Y. Lu, and W.-Y. Ma. (2005). Web resource geographic location classification and detection. In *Proceedings of the International Conference on the World Wide Web*, pp. 1138–1139.

[29] K. Wang, T. Walker, and Z. Zheng. (2009). PSkip: estimating relevant ranking quality from Web search clickthrough data. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1355–1364.

[30] L. Wang, C. Wang, X. Xie, J. Forman, Y. Lu, W.-Y. Ma, and Y. Li. (2005). Detecting dominant locations from search queries. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 424–431.

[31] R.W. White, P.N. Bennett, and S.T. Dumais. (2010). Predicting short-term interests using activity-based search context. In *Proceedings of the ACM CIKM Conference on Information and Knowledge Management*, pp. 1009-1018

[32] Q. Wu, C. J.C. Burges, K. M. Svore and J Gao. (2008). Ranking, Boosting, and Model Adaptation. *Microsoft Research Technical Report MSR-TR-2008-10.*

[33] B. Yu and G. Cai (2007). A query-aware document ranking method for geographic information retrieval. In *Proceedings of the ACM GIR Workshop on Geographical Information Retrieval*.

[34] F. Zhong, D. Wang, G. Wang, W. Chen, Y. Zhang, Z. Chen, and H. Wang. (2010). Incorporating post-click behaviors into a click model. In *Proceeding of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 355–362.

[35] Z. Zhuang, C. Brunk, and C. L. Giles. (2008). Modeling and visualizing geo-sensitive queries based on user clicks. In *Proceedings of the International Workshop on Location and the Web*, pp. 73–76.