An Exploration of Ranking Heuristics in Mobile Local Search

Yuanhua Lv
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
ylv2@uiuc.edu

Dimitrios Lymberopoulos †, Qiang Wu ‡
† Microsoft Research
‡ Microsoft
Redmond, WA 98052, USA
{dlymper, qiangwu}@microsoft.com

ABSTRACT

Users increasingly rely on their mobile devices to search local entities, typically businesses, while on the go. Even though recent work has recognized that the ranking signals in mobile local search (e.g., distance and customer rating score of a business) are quite different from general Web search, they have mostly treated these signals as a black-box to extract very basic features (e.g., raw distance values and rating scores) without going inside the signals to understand how exactly they affect the relevance of a business. However, as it has been demonstrated in the development of general information retrieval models, it is critical to explore the underlying behaviors/heuristics of a ranking signal to design more effective ranking features.

In this paper, we follow a data-driven methodology to study the behavior of these ranking signals in mobile local search using a large-scale query log. Our analysis reveals interesting heuristics that can be used to guide the exploitation of different signals. For example, users often take the *mean* value of a signal (e.g., rating) from the business result list as a "pivot" score, and tend to demonstrate different click behaviors on businesses with lower and higher signal values than the pivot; the clickrate of a business generally is *sublinearly* decreasing with its distance to the user, etc. Inspired by the understanding of these heuristics, we further propose different transformation methods to generate more effective ranking features. We quantify the improvement of the proposed new features using real mobile local search logs over a period of 14 months and show that the mean average precision can be improved by over 7%.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Miscellaneous

General Terms

Algorithms, Human Factors, Measurement

Keywords

Mobile local search, search log analysis, ranking heuristics

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

1. INTRODUCTION

The wide availability of internet access on mobile devices, such as phones and personal media players, has allowed users to search and access Web information while on the go. According to a recent report from comScore [1], as of July 2011, there were over 234 million mobile users in U.S., with nearly 50 percent searching on their mobile devices. The availability of continuous fine-grained location information on these devices has enabled mobile local search, which employs user location as a key factor to search for local entities, to overtake a significant part of the query volume. Sohn et al.'s study [34] found that 38% of mobile information needs are local. This is also evident by recent reports by BIA/Kelsey [2] which show that 30% of all search volume will be local in nature by 2015, as well as by the rising popularity of location-based search applications such as Google Local, Bing Local, and Yelp.

Even though mobile local search is similar to general Web search in that they both boil down to a similar problem of relevance/click prediction and result ranking, there are two fundamental differences and also challenges in developing effective ranking functions for mobile local search.

First, the ranking signals in mobile local search are quite different from general Web search. On the one hand, Web search handles a wide range of Web objects, particularly webpages, while mobile local search focuses mostly on ranking local businesses (e.g., restaurants). Therefore, special domain knowledge about the ranking objects in mobile local search could be exploited to improve ranking accuracy. For instance, businesses may receive *ratings* and *reviews* from their customers thanks to the Web 2.0 services, which have been shown to be useful signals for ranking businesses [3, 43]. On the other hand, local search users generally prefer businesses that are physically close to them; this is particularly critical for mobile users who are on the go and their range of reach might be limited. For example, a user would be more likely to visit a restaurant within 1 kilometer than another one within 2 kilometers to get breakfast, if the two restaurants are similarly good on other aspects. The *distance* between the result business and the user's location has been recognized as an important ranking signal in mobile local search [17, 3, 26]. In fact, the customer rating score, the number of reviews, and the distance are all shown explicitly to users in the search result user interface of mobile local search, as shown in Figure 1, and therefore play an important role in influencing the user's click decision.

[1] http://www.comscore.com/
[2] http://www.biakelsey.com/

Figure 1: Sample mobile local search result.

Properly studying and modeling how this information affects user click behavior is arguably the key to improving ranking accuracy.

In spite of the recognition of these new ranking signals, previous work has mostly treated them as a black-box to extract very basic features (e.g., raw rating scores and distance values) without going inside the signals to study how exactly they affect the relevance or clickrate of a business. For example, it is unclear how the clickrate of a business changes with its rating score: does a lower rating score necessarily lead to a lower clickrate? In the aforementioned restaurant example, a 1 kilometer difference in distances may lead to significantly different clickrates of two restaurants, but would the same 1 kilometer distance difference also cause similar clickrate difference of another two restaurants that are further away from user's location, e.g., 10 and 11 kilometers instead of 1 and 2 kilometers away? It is critical to understand the underlying behaviors/heuristics of a ranking signal, which would guide us to design more effective ranking features; this has been demonstrated extensively in the development of retrieval functions for general Web search, e.g., [30, 33, 11, 25].

Second, similarly to personalization in Web search [32, 36, 35, 13, 37, 28], personal preference affects user click behavior and can therefore constitute an important ranking signal for mobile local search [38]. For instance, knowing that a mobile user searching for restaurants prefers Chinese food, we can rank more Chinese restaurants on the top to avoid bothering the user with other types of restaurants. However, conversely to Web search, it is non-trivial to build user profiles for capturing personal preference of different businesses in mobile local search. On the one hand, the text associated with each business is often very sparse so that it would be hard to build content-based user profiles proposed previously [32, 36]. On the other hand, due to the local nature, a user tends to only click nearby businesses, so it is hard to find users who live far away from each other but share similar business click patterns, making it difficult to apply the collaborative filtering approaches (e.g., [35, 13]) or statistical topic modeling approaches (e.g., [20, 4, 27]), for user profiling.

Inspired by the understanding of these challenges, in this paper, we explore the ranking heuristics behind these new signals in mobile local search to develop more effective ranking features. Specifically, our contributions are threefold.

First, we follow a data-driven methodology to study the behavior of the new ranking signals in mobile local search using a large-scale query log. Our analysis reveals interesting heuristics that can be used to guide the exploitation of different signals. For example, we reveal a common phenomenon for all signals involved in our study, i.e., users often take the *mean* value of a signal from the business result list as a "pivot" score, and tend to demonstrate different click behaviors on businesses with lower and higher signal values than the pivot; the clickrate of a business generally is *sublinearly* decreasing with its distance to the user, etc. Motivated by these heuristics, we further propose different normalization methods to generate more effective ranking features.

Second, we exploit domain knowledge of businesses, i.e., business category information used in a commercial search engine, and employ a back-off strategy to estimate the user preference of business categories instead of specific businesses: two users from different locations, though hardly co-clicking any specific business, may still be interested in similar business categories. We study how such a user preference signal affects the clickrate of a business and design effective strategies to generate personalization features.

Third, we develop a clickrate prediction function to leverage the complementary relative strengths of various signals, by employing a state-of-the-art predictive modeling method, MART [15, 16, 40]. In doing this, we hope to exploit the strength of machine learning to quantify the improvement of the proposed features. We evaluate the performance of the proposed new features using real mobile local search logs over a period of 14 months, with an emphasis on those difficult queries, and show that the mean average precision can be improved by over 7%.

2. RELATED WORK

There have been several large scale studies in the past on mobile query log analysis for deciphering mobile search query patterns [22, 23, 9, 41, 8, 38]. The goal of these studies is to provide quantitative statistics on various aspects of mobile search that can help gain better insight on the mobile users' information needs. However, few of these efforts have provided insight about the ranking issue.

In early studies of local search or geographic search, some work attempted to identify queries that may not contain an explicit geographic reference (e.g., a city name) but have a "geo-intent" nevertheless [39, 2, 42], while some others focused on improving the query processing and ranking efficiency [7, 12, 10]. They are all orthogonal to our work in that we study how to improve the ranking accuracy for "geo-intent" queries.

Recently, the task of improving the ranking accuracy of mobile local search has also begun to attract efforts [1, 24, 3, 26] which have already recognized that the ranking signals in mobile local search (e.g., distance and rating score of a business) are quite different from general Web search. However, these studies have mostly treated such ranking signals as a black-box to extract very basic features (e.g., raw distance values and rating scores). Although some statistics of these signals are also often used as complementary features, such as the average distance and the standard deviation of distance in the current location [26], existing work relies purely on machine learning techniques to combine all features without going inside the signals to understand how exactly they affect user click behaviors. In contrast to existing studies, our work is a first attempt at understanding the behaviors and heuristics of these ranking signals for click prediction in mobile local search.

It has been previously demonstrated that understanding the behaviors/heuristics of a ranking signal is critical in the development of retrieval functions for Web search [30, 33, 14, 11, 25]. For example, the term frequency signal, which assigns a higher score to a document if it contains more occurrences of the query term, should be normalized to prevent the contribution of repeated occurrences from growing too large due to the burstiness phenomenon [30, 14], and

the term frequency signal should also be normalized by document length since long documents tend to use the same terms repeatedly [30, 33, 14]. All effective retrieval models in Web search have implemented these heuristics [14], and previous work has also shown that a retrieval function tends to work poorly if any desirable retrieval heuristic is violated [14, 25]. Inspired by the successes and lessons from the development of retrieval models, we thus explore the ranking heuristics in mobile local search and try to understand how exactly a ranking signal in mobile local search is related to the clickrate/relevance of a business.

Additionally, opinionated content has also been exploited in some existing search tasks. For example, in opinion retrieval [29], the goal of the task is to locate documents (primarily blog posts) that have opinionated content; Ganesan and Zhai [18] studied the use of the content of customer reviews to represent an entity (e.g., business) in entity ranking; Zhang et al. [43] proposed different methods to aggregate the counts of thumb-ups and thumb-downs for rating prediction, etc. Different from previous work, our work aims at understanding the relationship between the clickrate of a business and its rating score and number of reviews in mobile local search.

Personalized search has attracted much attention in Web search [32, 36, 13, 28, 38]. To the best of our knowledge, no previous work has exploited personalization for mobile local search. In this paper, we design appropriate approaches to exploiting personalization in mobile local search and quantify their impact on ranking accuracy using real mobile local query logs.

3. EXPERIMENTAL SETTING

3.1 Dataset

To the process of acquiring relevance judgments and evaluating mobile local search is a challenging problem. First, the Cranfield style evaluation that has been used in the evaluation of many traditional information retrieval tasks [31] would not work here, since the relevance judgments in mobile local search are particularly dependent on the search context, e.g., location of the user [24, 26]. Second, asking users to make explicit relevance judgments can be very costly because it is necessary to cover a diverse set of queries in different contexts. Third, although Joachims et al. have developed methods for extracting relative relevance judgments from user clickthrough data in general Web search [21], it is unclear if these methods also work for mobile local search where the position bias of clickthrough and the interpretation of clickthrough may be different from general Web search. Due to these problems in applying traditional evaluation methods, in our work, we choose to follow the previous work on mobile local search [3, 26] and simply use clicks to approximate the relevance judgments. Although each individual user click may not be very reliable, the aggregation of a great number of user clicks from a large-scale query log could still provide powerful indicator of relevance. Thus, it is very likely that features that help improve click prediction will be useful in ranking as well.

Therefore our task is, given a query, to predict if a candidate business would be clicked, and then rank the candidate businesses based on the click prediction. In our experiments, the query is sampled from the search log, while the candidate businesses are all those businesses that were shown to the user for that query. To learn and evaluate a click prediction model, we split the query log into four parts. The first 9 months of data is kept out as the "history" data, and is used purely for estimating the popularity of a business and the user's personal preference. We sample queries from the next 3, 1, and 1 months of data respectively for training, validating, and testing the click prediction models.

We apply three preprocessing steps to make these four datasets more practical and representative: (1) We exclude queries that did not receive any click or received clicks for every candidate business, since these queries will not influence the average ranking performance in the experiments. (2) We identify and filter out queries that match a business name exactly, e.g., "Starbucks"; solving such kind of queries is a relatively easy task, since users usually have clearer information needs (e.g., visiting a nearby Starbucks) as compared to other more general queries, e.g., "Coffee". In doing this, we place an emphasis on difficult queries in our study. (3) We empirically remove queries (from all the four datasets) by "users" who issued more than 1000 queries in total in the training, validation, and test datasets, because such "users" are more likely to be robots. Finally, we obtain 60475, 18491, and 23152 queries as the training, validation, and test queries; the average number of clicks and the average number of candidate businesses for each query are 1.34 and 17 respectively.

3.2 Learning Model for Click Prediction

Since we need to leverage multiple signals for click prediction, we seek help from machine learning. We adopt MART [40], a learning tool based on Multiple Additive Regression Trees, to provide a common learning framework on top of which we can compare the performance of different ranking heuristics and ranking features. MART is based on the stochastic gradient boosting approach described in [15, 16] which performs gradient descent optimization in the functional space. In our experiments on click prediction, we used the log-likelihood as the loss function, used steepest-decent (gradient descent) as the optimization technique, and used binary decision trees as the fitting function.

We construct a training instance for each query-business pair, which consists of a set of features (e.g., distance, rating, etc.) and a click label which indicates if the user clicks the business (1 for click and 0 otherwise). The training and validation data are fed into MART to build a binary classification model, which we use to estimate the probability of clicks in the test data.

Note that the choice of machine learning algorithms is not critical in our paper: we only take machine learning as a black-box tool to evaluate the proposed heuristics and features. We chose MART mainly because it can potentially handle non-linear combination of features, and it is widely adopted in current commercially available search and advertisement ranking engines.

The main goal of our experiments is to explore and evaluate effective ranking heuristics for boosting the special ranking signals in mobile local search. Our baseline feature set contains 6 representative features that are selected based on previous research studies [17, 24, 26]. These features are: (1) the distance between the query and the business locations, (2) the popularity measure of the business as defined by the number of clicks in the history search logs, (3) the clickrate of the business in the history data as defined by the

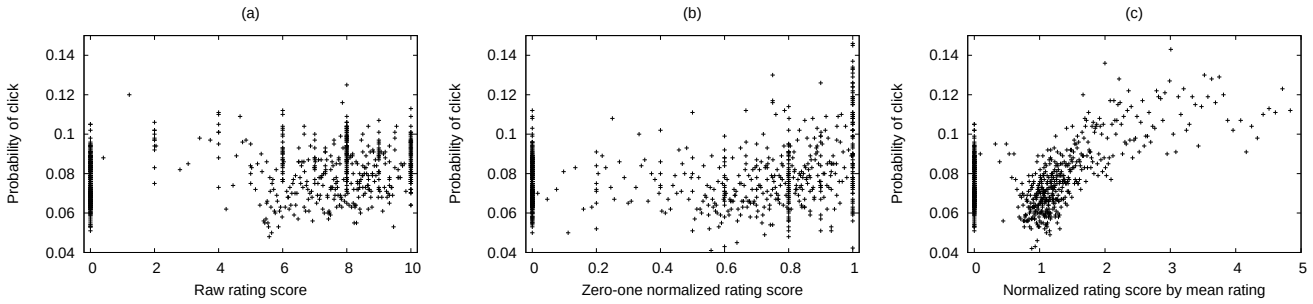

Figure 2: Probability of click for businesses from a bin, plotted against the mean rating score of this bin.

Distance	1	Rating	0.3794
#Clicks	0.7193	#Reviews	0.3462
ClickRate	0.9976	TimeFrame	0.1752

Table 1: Relative feature importance in baseline

number of clicks divided by the number of impressions and defined as 0 if it did not occur in the history data, (4) the customer rating score of the business in a range of $[0, 10]$, (5) the number of customer reviews of the business, and (6) a time code that represents both the time frame within a day (one out of five time frames) that the query was submitted and the day of the week that the query was submitted (weekend or weekday). Since we only re-rank top-ranked businesses from a commercial mobile local search engine, and those businesses that do not match the query keywords at all have already been eliminated, we thus do not involve any textual matching feature and focus only on the special signals of mobile local search.

We evaluate the retrieval performance in terms of MAP (Mean Average Precision) and the precision at different recall levels. Because we are only re-ranking a set of top-ranked businesses in response to a query, the recall score will be the same for any ranking model. In other words, MAP will be only influenced by the positions of the relevant results. So we believe that, in our study, MAP is a good measure to capture the ranking accuracy of top-ranked results. Besides, we also compare the importance of each feature in constructing the MART model, following the relative importance measure proposed in [15].

4. RANKING SIGNALS AND HEURISTICS

In mobile local search, the rating score, the number of reviews, and the distance are not only the key back-end ranking signals, but also important information displayed explicitly in the search result UI as shown in Figure 1. Although the "personal preference" signal is not explicitly shown to users, users certainly know their own preference. That being said, these four ranking signals are directly observable to users, and users' click behaviors presumably would be heavily dependent on these signals. Therefore, understanding how exactly a user's decision relates to these signals would potentially lead to better ways of modeling these signals and thus to improving mobile local search. In this section, we study in detail these four ranking signals.

4.1 Customer Rating

Intuitively, the customer rating score of a business would significantly affect users' click behaviors in mobile local search, since users are used to consulting other people's ratings and opinions about an entity to help make their own decision [29]. To verify this intuition, we trained a click prediction model using the baseline features described in the previous section and examined the relative importance of the different features. The results shown in Table 1 indicate that conversely to our intuition, the importance of the rating score as a feature is relatively low in our baseline system.

4.1.1 Likelihood of Click/Relevance

To examine this observation, we analyze the likelihood of relevance (i.e., clickrate) of businesses of all rating scores, and plot these likelihoods against the rating score to obtain a "click pattern". Intuitively, the likelihoods should increase monotonically with the rating score.

To estimate these likelihoods, we sort all the businesses in the training data in order of increasing rating score, and divide them into several equal sized (i.e., 1000) "bins", yielding 1032 different bins. We select the mean rating score in each bin to represent the bin on the graphs used in later analysis. We can then compute the probability of a randomly selected business from the ith bin getting clicked, which is the ratio of the number of clicked businesses from the ith bin, and the bin size. In terms of conditional probability, given a business b, this ratio of the ith bin can be represented by $p(b \text{ is clicked} \mid b \in Bin_i)$.

Figure 2(a) shows how the probabilities obtained from the above analysis relate to the mean rating score in a bin. Surprisingly, there seems to be no clear relationship between the likelihood of click and the rating score.

This anti-intuitive observation could be possibly caused by the potentially incomparable rating scores across different queries: result businesses retrieved by some queries may have higher rating scores than that retrieved by some other queries. To verify this, we adopt the popular zero-one score normalization method, which linearly normalizes the rating scores of every query to a range of $[0, 1]$. Such a normalization strategy has been widely used for feature normalization in many learning to rank tasks, e.g., [6]. After that, we do a similar analysis of the probabilities of click, but against the normalized rating score. The results are shown in Figure 2(b). Unfortunately, there is still no clear relationship, suggesting that the ineffectiveness of the rating score as a feature is not purely caused by the incomparable score range.

4.1.2 The "Mean" Normalization Scheme

To diagnose the problem, we further look into the distribution of rating scores of businesses that get clicked. Since the distribution of rating scores is intuitively related to the

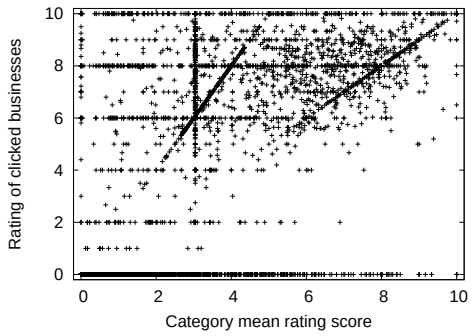

Figure 3: Rating of clicked businesses, plotted against the mean rating score of the corresponding business category.

type/category of businesses, we do this analysis in a category-aware way. Specifically, we compare the rating score of a clicked business with the mean rating score of all businesses from the same category. In doing this, we hope to understand if the clicked businesses are among the highly rated businesses in a category. The comparison results are illustrated in Figure 3, in which we show 10% of randomly selected *clicked* businesses from the training data.

Interestingly, we can see that the rating scores of most of the clicked businesses are above their corresponding category mean rating score. For example, when the category mean rating score is 4, few businesses with a rating score lower than 4 get clicked. This shows that the rating score is indeed useful and is directly related to users' click behaviors. However, how does a user know the mean rating score of a category so as to click businesses above this score?

In reality, we expect that users do not really know the mean score of a category. However, users may be able to have an approximate estimation of this mean score through looking over the retrieved business list: the retrieved businesses for a query often belong to the same category, and thus could be used as a sample set of businesses from that category, the mean rating score of which can be viewed as approximately the mean category rating. If the above assumption is true, it may suggest that, users often take the mean rating score from the business result list as a "pivot" score, and tend to click businesses with higher scores than this pivot. This intuitively makes sense: a user's click decision, although influenced by the rating score, is not entirely based on it, but if the rating of a business is above his/her expectation (i.e., the pivot) which is learned from the result list in an ad hoc way, the business would be more likely to be clicked.

Inspired by this analysis, we propose to normalize rating scores using the mean rating score of the retrieved businesses. A straightforward approach is to divide the original rating score using this mean value, which not only makes the rating scores more comparable across queries but also aligns them at the corresponding mean rating score of each query. We plot the probabilities of click against the normalized rating score in Figure 2(c). It is clear that the probability of click increases monotonically with the normalized rating when the normalized rating is larger than 1 (i.e., the mean point), while the probability tends to be random when the normalized rating is lower than 1. This is an empirical

Methods	MAP	P@0.3	P@0.5	P@0.8
Baseline	.419	.441	.434	.403
ZeroOneNorm	.419	.442	.434	.403
MeanNorm*	$.425^{bz}$	$.448^{bz}$	$.440^{bz}$	$.409^{bz}$
AutoNorm-C	$.422^{b}$	$.445^{b}$	$.437^{b}$	$.406^{b}$
AutoNorm-Q*	$.431^{mc}$	$.454^{mc}$	$.446^{mc}$	$.415^{mc}$
RatingPred*	$.428^{b}$	$.451^{b}$	$.443^{b}$	$.413^{b}$
Norm+Pred	$.438^{qr}$	$.461^{qr}$	$.453^{qr}$	$.421^{qr}$

Table 2: Comparison of methods for modeling rating scores. "Norm+Pred" combines methods tagged using $*$. $b/z/m/c/q/r$ indicates the significance over Baseline, ZeroOneNorm, MeanNorm, AutoNorm-C, AutoNorm-Q, and RatingPred respectively, at the 0.001 level using the Wilcoxon non-directional test.

verification of the fact that users tend to take the mean rating score from the observed results as a "pivot" score, and a clear demonstration of different click behaviors on businesses with lower and higher scores than the pivot score.

To examine if the proposed simple mean normalization scheme can really improve ranking accuracy, we use the normalized rating score to replace the original rating score and learn a new model, labeled as "MeanNorm", and compare its performance with the baseline model. In addition, we also take the widely used zero-one strategy for rating normalization as another baseline run, which is labeled as "ZeroOneNorm". The comparison results are reported in Table 2, and show that the mean normalization scheme works the best, achieving significant improvement over both baseline runs. At the same time, the zero-one normalization does not improve the accuracy of the baseline model. This suggests that we can improve ranking performance by pointing out the mean rating value.

Another approach to explicitly normalizing the rating of a business with the mean rating of all result businesses would be to encode the mean rating of result businesses as an additional feature and let the training algorithm itself decide how to do the normalization. To evaluate this approach, we train a new model, labeled as "AutoNorm-Q", by adding the mean rating score of businesses from the same single *query* into the baseline feature set. We also train another model, labeled as "AutoNorm-C", in which we add the mean rating score of businesses from the same *category* into the baseline. We present the performance of these two runs in Table 2. The results demonstrate that AutoNorm-Q works much better than AutoNorm-C, confirming our analysis that users select the "pivot" from the businesses shown to them. AutoNorm-Q improves over the baseline by approximately 3%. Moreover, AutoNorm-Q improves over MeanNorm, suggesting that the mean normalization scheme can be boosted by optimizing the way of exploiting the "mean" in a supervised manner.

4.1.3 Cluster-based Smoothing of Ratings

It is often the case that a business does not receive any customer rating. In the presence of missing rating scores, a default value of 0 is often used, which, however, may be inaccurate: (1) a business that does not receive any customer rating does not necessarily mean that it should be rated low; (2) it could be unfair to use the same default rating score for all businesses. Even if a business receives a rating score, it

may still be inaccurate if the rating is only contributed by a very small number of customers. Therefore, more accurate prediction/smoothing of ratings could potentially improve ranking accuracy, and we propose a cluster-based method to predict/smooth rating values.

The basic idea is based on the cluster hypothesis [19], and averages rating scores $r(x)$ from all businesses x in the same cluster C to smooth the rating $r(b)$ of business b so as to obtain an updated rating $r'(b)$. The intuition is that businesses in the same cluster should receive similar ratings, and the rating stability of a cluster would benefit an individual business. Formally,

$$r'(b) = f\left(r(b), \frac{1}{|C|}\sum_{x\in C, b\in C} r(x)\right) \quad (1)$$

where f is a function to control rating update. The key component is thus the business cluster. We use two types of clusters: business category and business chain. The former allows us to use the rating of all businesses in a given category to estimate the rating of an unrated business in that category (e.g. use all businesses in the "Coffee & Tea" category to estimate the rating score of a Starbucks business). The latter approach, allows us to estimate the rating score of a business by exploiting the rating score of other businesses belonging to the same chain (e.g. use different Strabucks coffeehouses rating scores to estimate the rating score of an unrated Starbucks coffeehouse).

There are two challenges with this approach: how to choose function f and how to leverage the evidences from two types of clusters. Inspired by the effective performance of automatic feature normalization in the previous section, we also let the learning algorithm optimize these two factors in a supervised way. Specifically, we provide both a category mean rating and a business-chain mean rating as two separate features to the learning algorithm. In addition, we also introduce two description variables for these two new features, i.e., the size of the category and the size of the business chain, to the learning algorithm.

This method is labeled as "RatingPred", and we present the experiment results in Table 2. We can see that RatingPred improves over the baseline significantly, suggesting the proposed cluster-based smoothing can indeed improve rating values. Furthermore, we combine RatingPred with the proposed feature normalization methods, leading to a new run labeled as "Norm+Pred". It is observed from Table 2 that Norm+Pred outperforms either single method alone, suggesting that smoothing ratings and normalizing ratings are complementary to each other. Norm+Pred improves over the baseline by more than 4.5%.

4.2 Review Count

The count of reviews represents another signal from the opinionated content, which can intuitively reflect the popularity of a business. However, we find that the importance of this signal is also low in the baseline model, as shown in Table 1. Similarly to the rating score analysis, we reveal that this is because users often take the mean review count from their observed businesses as a "pivot", and demonstrate different click patterns on businesses with lower and higher review count than the pivot. However, the learning algorithm fails to capture this important information. To better exploit the strengths of this signal, we need to feed this pivot number (i.e., mean review count) to the learning algorithm. That is, we should either manually normalize the review count using the pivot, or introduce the pivot as an additional feature for automatic normalization.

Methods	MAP	P@0.3	P@0.5	P@0.8
Baseline	.419	.441	.434	.403
ZeroOneNorm	.421	.444	.436	.405
MeanNorm*	$.425^{bz}$	$.448^{bz}$	$.441^{bz}$	$.409^{bz}$
AutoNorm-C	$.423^{b}$	$.445^{b}$	$.438^{b}$	$.407^{b}$
AutoNorm-Q*	$.431^{mc}$	$.454^{mc}$	$.446^{mc}$	$.414^{mc}$
ReviewsPred*	$.429^{b}$	$.451^{b}$	$.443^{b}$	$.413^{b}$
Norm+Pred	$.438^{qr}$	$.461^{qr}$	$.453^{qr}$	$.421^{qr}$

Table 3: Comparison of methods for modeling review counts. "Norm+Pred" combines methods tagged using $*$. The description of notations $b/z/m/c/q/r$ are the same as Table 2.

The experimental results presented in Table 3 show that the ranking performance can be significantly improved by the "mean" normalization scheme. Different notations in the table are defined similarly to their counterparts in Table 2 but applied to normalize review count. Specifically, the simple mean normalization method (i.e., MeanNorm) performs significantly better than the widely used score normalization method (i.e., ZeroOneNorm) which does not leverage the mean value; also, automatic normalization (i.e., AutoNorm-Q) works more effectively than manual normalization (i.e., MeanNorm).

In addition, similar to the rating score, we can also improve ranking performance by smoothing review count based on the cluster hypothesis to make it more accurate (this run is labeled as "ReviewsPred"). The mean normalization scheme and the cluster-based smoothing can be leveraged together (i.e., Norm+Pred) to further improve performance, achieving 4.5% improvement in MAP.

4.3 Distance

Local search differs from other search tasks mainly because its ranking signals feature geographical distance. In fact, distance has also been shown to be one of the most important features in both previous work, e.g., [26], and our work, as shown in Table 1.

To understand how distance affects the click patterns of users, we first plot in Figure 4 (left) the likelihood of click for a business against its distance from the user. Interestingly, there is indeed a monotonically decreasing trend of the likelihood with respect to distance; this may explain why distance appears to be the most important feature in our experiments. Furthermore, we observe that the likelihood is decreasing sub-linearly with distance: the likelihood decreases with distance, but the decreasing speed drops as distance becomes large. This intuitively makes sense: a restaurant within 1 mile would have clear advantages over another similar restaurant within 2 miles, but two restaurants within 9 miles and 10 miles may not have much difference. That is, the relevance of a business is more sensitive to its distance when the distance value is smaller.

With the Box-Cox transformation analysis [5], we find there is approximately a logarithm transformation. To illustrate it, we plot the likelihood of click with respect to the logarithm transformation of distance in Figure 4 (right).

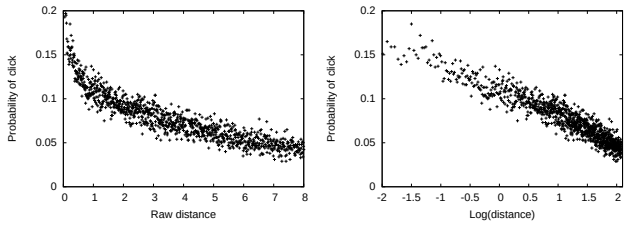

Figure 4: Probability of click for businesses from a bin, plotted against the mean distance (left) and Log(distance) (right) scores of this bin.

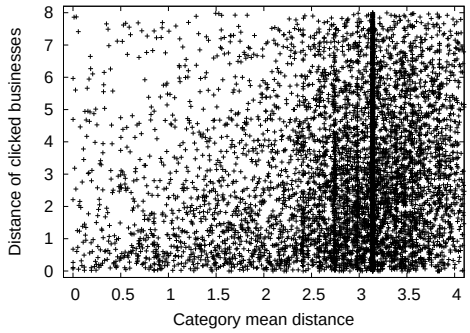

Figure 5: Distance of clicked businesses, plotted against the mean (traveling) distance of the corresponding business category.

Linear scaling in distance would overly penalize businesses that are relatively far away from the user, while the logarithm transformation generally improves modeling distance.

Similar to ratings and review counts, distance is also observable to users. One interesting question is if there is also a "pivot" click phenomenon. To answer this question, we plot the distance of clicked businesses against the mean traveling distance to businesses in the same category, as shown in Figure 5. Indeed, the plot shows that users tend to click businesses closer than the category mean distance, suggesting that the "mean" normalization scheme could also be applicable in the case of the distance feature. Yet, the "pivot" phenomenon of distance is not as clear as that of ratings and review counts. One possible reason is that users can generally understand distance better than ratings and review counts, because distance is a concrete concept, while ratings and review counts appears to be more abstract and subjective; as a result, users tend to rely more on the statistics of the observed business list to make sense of ratings and review counts, but absolute distance itself may have already made much sense. This is also consistent with our previous observation that there is a clear relationship between raw distance values and the probability of click, as shown in Figure 4 (left).

We now verify our analysis using empirical experiments, the results of which are reported in Table 4. We first apply the simple mean normalization method to divide distance by the mean distance value of all observed businesses for the same query. This run is labeled as "MeanNorm". We can see it improves over the baseline system significantly, which suggests that feature normalization still helps even though absolute distance values are already largely comparable. We next compare "MeanNorm" with "MeanNorm+" in which the simple mean normalization method is applied to log(distance). Apparently, "MeanNorm+" works more effectively, confirming our analysis that the logarithm transformation is useful for better modeling distance. In addition, we create another run "ZeroOneNorm+" which differs from "MeanNorm+" in that the zero-one normalization is used. We observe that "ZeroOneNorm+" works significantly worse than "MeanNorm+", confirming our analysis that the "mean" normalization scheme works well for distance, and suggesting that users would also like to click businesses with a distance smaller than the pivot (i.e., mean distance in the result list).

Methods	MAP	P@0.3	P@0.5	P@0.8
Baseline	.419	.441	.434	.403
MeanNorm	$.429^{b}$	$.451^{b}$	$.443^{b}$	$.414^{b}$
ZeroOneNorm+	$.428^{b}$	$.451^{b}$	$.443^{b}$	$.413^{b}$
MeanNorm+	$.435^{mz}$	$.457^{mz}$	$.449^{mz}$	$.420^{mz}$
AutoNorm	$.434^{m}$	$.456^{m}$	$.449^{m}$	$.419^{m}$
AutoNorm+	$.435^{ma}$	$.457^{ma}$	$.450^{ma}$	$.420^{ma}$

Table 4: Comparison of methods for modeling distance. Methods with an indicator "+" applies logarithm transformation. $b/z/m/a$ indicates the significance over Baseline, ZeroOneNorm+, MeanNorm, and AutoNorm respectively, at the 0.05 level using the Wilcoxon non-directional test.

Finally, we also evaluate two automatic mean normalization runs, namely "AutoNorm" and "AutoNorm+", where we introduce the mean value of distance and log(distance) in the search results as additional features into the training process to let the learning algorithm automatically normalize the distance and the log(distance) features respectively. First, "AutoNorm+" outperforms "AutoNorm", though the improvement is small; this suggests that sub-linear transformation of distance is beneficial, yet its advantage tends to be weakened as we use automatic feature normalization, because MART can potentially handle non-linear combination automatically. Second, by comparing "AutoNorm(+)" with "MeanNorm(+)", we can see that automatic normalization can work better than or comparable to the simple mean normalization. Overall, both automatic normalization and manual normalization can improve over baseline by approximate 4% with the proposed new features.

4.4 Personal Preference

Our goal is to build user profiles so that we can compute the user preference of a business so as to rank businesses in a user adaptive way. However, it is non-trivial to build content-based user profiles [32, 36] in mobile local search, since the text associated with each business is often very sparse. Thus, we choose to use the collaborative filtering approach [20, 27, 4], based on the history click data, to estimate the likelihood that a user u likes business b, formally the conditional probability $P(b|u)$. Yet there is another challenging problem: due to the local nature of the task, a user tends to only click nearby businesses, so the co-occurrences are also of local nature and thus very sparse. For example, it is very hard to find users who live far away from each other but share similar business click patterns. To solve this problem, we exploit the domain knowledge of businesses and instead estimate the likelihood that a user u

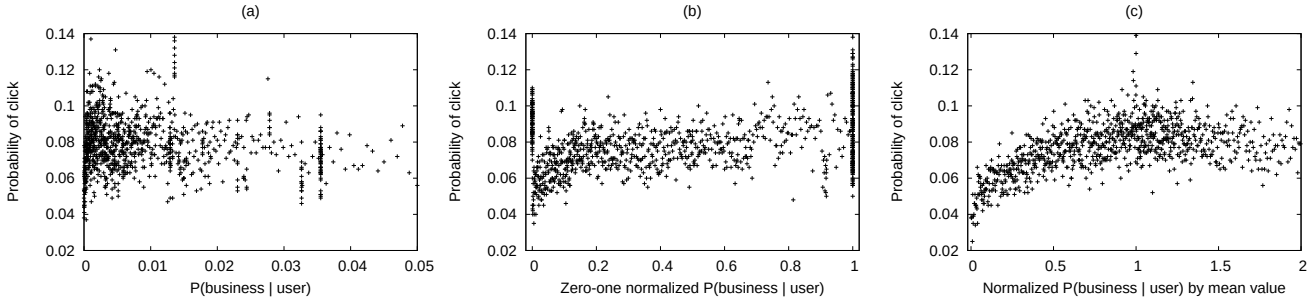

Figure 6: Probability of click for businesses from a bin plotted against the mean user preference of this bin.

likes business category c, i.e., $P(c|u)$: as a business category can cover businesses from different locations, co-occurrences of categories and users can happen across locations. As a by-product, the number of categories, which is about 3000, is only about 1/5000 of the number of businesses, significantly reducing the dimension. Our experiments show that we can build profiles for 1 million users in several hours on a single machine.

Although the category information of a business that the user has clicked can be obtained directly from the history data, due to the data sparseness problem of many users, we follow the idea of statistical topic modeling to estimate $P(c|u)$ in a more smoothing way. First, we introduce hidden variables Z with states z for every user-category pair. The possible set of states z is assumed to be finite and of size k. We empirically set $k = 100$ in our work. We can map our problem to the standard topic modeling problem: the original document-term matrix is replaced by a user-category matrix, and the original co-occurrence relationship is replaced by a click. In our work, we adopt PLSA [20] and LDA [4]. Since these two models perform similarly effectively in our experiments, we only show the results based on PLSA.

Considering observations in the form of clicks (c, u) of categories and users, PLSA models the probability of each click (i.e., a user u clicks a business of category c) as a mixture of conditionally independent multinomial distributions:

$$P(c, u) = \sum_z P(z)P(u|z)P(c|z) = P(u)\sum_z P(z|u)P(c|z) \quad (2)$$

Since our problem is to estimate user preference, we will work with the conditional model

$$P(c|u) = \sum_z P(z|u)P(c|z) \quad (3)$$

The model can be estimated using the Expectation Maximization (EM) algorithm to obtain parameters $P(z|u)$ and $P(c|z)$. Now, we have constructed a user profile $P(c|u)$. Then given any business b, since b may belong to multiple categories, we average the conditional probabilities of these corresponding categories as the user preference of a business, i.e., $P(b|u)$.

Some users may have more history clicks, and the profiles of these users would intuitively be more reliable than that of some other users who make fewer clicks. In order to make the model more intelligent so as to be able to automatically learn how much personalization we should apply for each user, we encode both the user preference, i.e., $P(b|u)$, and the number of history clicks of the user into the learning algorithm. We do not do any normalization, and this run is labeled as "NoNorm". We compare it with the baseline and observe that, though "NoNorm" outperforms the baseline significantly, the improvement is indeed minor.

To examine the reason, we follow Section 4.1.1, and plot the probability of click for businesses against the user preference, as shown in Figure 6 (a). We can see that the probability of click generally does not vary a lot when the user preference changes; this may be one possible reason why "NoNorm" does not work very well. We then apply the zero-one normalization and generate another plot in Figure 6 (b). It shows that the zero-one normalization essentially stretches the plot along the x-axis. There also appears to be an increasing trend only when the user preference is very small. Next, we try the mean normalization method in Figure 6 (c). It shows clearly that when the user preference is below the mean value (i.e., $x = 1$) of the current search results, the probability of click increases monotonically with user preference and the increasing speed decreases as the user preference approaches its mean value. However, after the user preference reaches the mean value, the probability of click even has a tendency to decrease slightly. Again, this observation shows that users choose the mean value as a pivot score and have different click behaviors on the two sides of the pivot. Furthermore, it is interesting to see that the probability of click is maximized when the user preference is around the mean value: too low preference may mean that the business is not interesting to the user (i.e., irrelevant business), while too high preference may indicate that the business could be too similar to what the user clicked before (i.e., redundant business). The pivot observation seems to demonstrate that a user's decision may like an exploration-exploitation tradeoff: exploit what he/she knows but meanwhile explore what he/she does not know.

Inspired by the analysis above, we develop a manual mean normalization run ("MeanNorm") and an automatic mean normalization run ("AutoNorm") for feature normalization. According to the results shown in Table 5, "AutoNorm" improves over both the baseline and "MeanNorm" significantly, while "MeanNorm" does not perform very well. We hypothesize that this could be because of the sublinear increasing curve of the mean normalization method, as shown in Figure 6 (c): similar to our observations of distance normalization, automatic normalization using MART can potentially handle non-linear combination well, while manual normalization cannot. To verify our intuition, we first apply a logarithm transformation based on the Box-Cox transformation analysis [5] onto user preference and then add the two normalization methods on top of the transformed feature, leading

Personalization	MAP	P@0.3	P@0.5	P@0.8
Baseline	.419	.441	.434	.403
NoNorm	$.420^{b}$	$.442^{b}$	$.434^{b}$	$.404^{b}$
MeanNorm	$.420^{b}$	$.442^{b}$	$.434^{b}$	$.404^{b}$
MeanNorm+	$.428^{nm}$	$.450^{nm}$	$.442^{nm}$	$.411^{nm}$
AutoNorm	$.427^{nm}$	$.450^{nm}$	$.442^{bm}$	$.411^{bm}$
AutoNorm+	$.428^{nm}$	$.450^{nm}$	$.443^{nm}$	$.412^{nm}$

Table 5: Comparison of methods for modeling user preference. Methods with an indicator "+" applies logarithm transformation. $b/n/m$ indicates the significance over Baseline, NoNorm, and MeanNorm respectively, at the 0.01 level using the Wilcoxon non-directional test.

Methods	MAP	P@0.3	P@0.5	P@0.8
Baseline	.419	.441	.434	.403
All	.449	.472	.464	.433
All-Rating	.448	.471	.463	.432
All-Reviews	.449	.472	.464	.433
All-Distance	.441	.464	.456	.425
All-Personalization	.448	.471	.463	.431
All-Rating-Reviews	.442	.464	.456	.426
All-Rating-Reviews -Personalization	.436	.458	.450	.420

Table 6: Sensitivity analysis. It shows that combining the proposed new features (i.e., "All") can improve the Baseline over 7%.

to two new runs "MeanNorm+" and "AutoNorm+". Table 5 shows that these two runs perform similarly well and the best among all methods, verifying our hypothesis and also showing the necessity of sublinear transformation. Overall, by normalizing the personalization features, we can obtain over 2% MAP improvements.

4.5 Sensitivity Analysis

We combine the most effective modeling methods for all the four signals into our final model, including "Norm+Pred" from Table 2 for modeling rating, "Norm+Pred" from Table 3 for modeling review count, "AutoNorm+" and "MeanNorm+" from Table 4 for modeling distance, and "AutoNorm+" and "MeanNorm+" from Table 5 for modeling user preference. The final model is labeled as "All" and shown in Table 6. Table 7 lists the 23 features in the "All" model as well as where each feature comes from. We can see that the "All" model outperforms the baseline by more than 7%, suggesting that understanding the behaviors and heuristics behind ranking signals can indeed lead to better modeling methods and thus improving ranking accuracy.

We have shown that the selected modeling methods (now as components in the final "All" model) perform very well in modeling each individual signal. Now we turn to examine how sensitive of these methods when we combine them together. We remove some new modeling method(s) at a time, while keeping all other modeling methods and the baseline features. For example, "All-Rating" in Table 6 is constructed by excluding from the "All" model the proposed novel features in the "Norm+Pred" method for rating modeling, while the features occurring in other models, including all baseline features, are kept. From Table 6, we can see that when we remove "Distance", the performance drops clearly, suggesting that "Distance" is very sensitive in the final model and its effect cannot be replaced. However, when we exclude "Rating", "Reviews", or "Personalization", the performance only decreases slightly or even does not change. It suggests that the effect of these signals may have a large overlap; as a result, although they perform well as individual signals, their performance could not add together.

	Features	Imp.	Contributor
1	MeanNorm_Log_Distance	1	Distance
2	ClickRate	0.5859	Baseline
3	#Clicks	0.5152	Baseline
4	#Reviews_Mean	0.3692	Reviews
5	Log_UserPref_Mean	0.3646	Personalization
6	Rating_Mean	0.3453	Rating
7	MeanNorm_#Review	0.3137	Reviews
8	|BusinessChain|	0.3132	Rating&Reviews
9	Log_MeanNorm_UserPref	0.2897	Personalization
10	MeanNorm_Rating	0.2242	Rating
11	|Category|	0.2229	Rating&Reviews
12	Log_Distance_Mean	0.1814	Distance
13	#CategoryReviews	0.1731	Reviews
14	CategoryRating	0.1665	Rating
15	User_#Clicks	0.1651	Personalization
16	Log_Distance	0.1434	Distance
17	#BusinessChainReviews	0.1394	Reviews
18	Distance	0.1122	Baseline
19	BusinessChainRating	0.1100	Rating
20	Rating	0.1071	Baseline
21	Log_UserPref	0.1057	Personalization
22	#Reviews	0.0686	Baseline
23	TimeFrame	0.0660	Baseline

Table 7: Relative feature importance in the final model

To go a step further, we remove "Rating" and "Reviews" at the same time, and find that its performance degrades much more than that of "All-Rating" and "All-Reviews". This observation confirms that ratings and review counts are highly redundant to each other. After removing "Rating" and "Reviews", we also remove "Personalization" in the last row of Table 6. We can see the performance degradation is much larger than when we remove "Personalization" from the "All" model, suggesting that the personalization features also tend to be redundant to "Rating" and "Reviews".

Finally, we go in depth to the feature level to analyze what are the relative importance of each feature, as shown in Table 7. Apparently, distance, with the proposed normalization method, appears to be the most important feature. The two popularity measures from the Baseline are the second and the third most important features. These observations are consistent with the findings from previous work (e.g., [26]) that distance and popularity dominate the ranking signals of mobile local search. However, other signals have also contributed many useful features. For example, the top-6 features covers all feature contributors.

Two particularly interesting observations are that (1) three mean values are ranked very high, and (2) 8 out of the top-12 features are directly related to the mean normalization scheme, suggesting that the proposed "mean" normalization scheme indeed helps model signals in mobile local search. Due to the effectiveness of the proposed new features, many features from the baseline have been pushed to the bottom of Table 7.

5. CONCLUSIONS

In this paper, we follow a data-driven methodology to study the ranking heuristics/behaviors of ranking signals, including the customer rating score, the number of reviews, the geographic distance, and the user preference, in mobile local search using a large-scale query log.

Our analysis reveals interesting heuristics that can be used to guide the exploitation of different signals. First, we reveal a common phenomenon for all these four signals: users often take the mean value of a signal from the business result list as a "pivot" score, and tend to demonstrate different click behaviors on businesses with lower and higher signal values than the pivot. Inspired by this understanding, we proposed a "mean" normalization scheme to encode the pivot information into feature normalization or into model training, which has been shown to improve modeling these signals significantly. Second, we find that the clickrate of a business is increasing/decreasing sublinearly with user-preference/distance; as a result, linear scaling in these signals would bias to/against some particular businesses. In order to overcome this problem, we propose sublinear transformation methods for modeling these signals, which work very well. We quantify the improvement of the proposed methods using real mobile local search logs over a period of 14 months and show that the mean average precision can be improved significantly by over 7%.

6. REFERENCES

[1] A. Amin, S. Townsend, J. Ossenbruggen, and L. Hardman. Fancy a drink in canary wharf?: A user study on location-based mobile search. In *INTERACT '09*, pages 736–749, 2009.

[2] L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak. Spatial variation in search engine queries. In *WWW '08*, pages 357–366, 2008.

[3] K. Berberich, A. C. König, D. Lymberopoulos, and P. Zhao. Improving local search ranking through external logs. In *SIGIR '11*, pages 785–794, 2011.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[5] Box, G. E. P. and Cox, D. R. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 26(2):211–252, 1964.

[6] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research - Proceedings Track*, 14:1–24, 2011.

[7] Y.-Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *SIGMOD '06*, pages 277–288, 2006.

[8] K. Church and N. Oliver. Understanding mobile web and mobile search use in today's dynamic mobile landscape. In *MobileHCI '11*, pages 67–76, 2011.

[9] K. Church, B. Smyth, P. Cotter, and K. Bradley. Mobile information access: A study of emerging search behavior on the mobile internet. *ACM Trans. Web*, 1, May 2007.

[10] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *Proc. VLDB Endow.*, 2:337–348, August 2009.

[11] N. Craswell, S. Robertson, H. Zaragoza, and M. Taylor. Relevance weighting for query independent evidence. In *SIGIR '05*, pages 416–423, 2005.

[12] I. De Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *ICDE '08*, pages 656–665, 2008.

[13] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW '07*, pages 581–590, 2007.

[14] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *SIGIR '04*, pages 49–56, 2004.

[15] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[16] J. H. Friedman. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38:367–378, February 2002.

[17] J. Froehlich, M. Y. Chen, I. E. Smith, and F. Potter. Voting with your feet: An investigative study of the relationship between place visit behavior and preference. In *Ubicomp'06*, pages 333–350, 2006.

[18] K. Ganesan and C. Zhai. Opinion-based entity ranking. *Information Retrieval*, 2011.

[19] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR '96*, pages 76–84, 1996.

[20] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99*, pages 50–57, 1999.

[21] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25, April 2007.

[22] M. Kamvar and S. Baluja. A large scale study of wireless search behavior: Google mobile search. In *CHI '06*, pages 701–709, 2006.

[23] M. Kamvar and S. Baluja. Deciphering trends in mobile search. *Computer*, 40:58–62, August 2007.

[24] N. D. Lane, D. Lymberopoulos, F. Zhao, and A. T. Campbell. Hapori: context-based local search for mobile phones using community behavioral modeling and similarity. In *Ubicomp '10*, pages 109–118, 2010.

[25] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *CIKM '11*, pages 7–16, 2011.

[26] D. Lymberopoulos, P. Zhao, A. C. König, K. Berberich, and J. Liu. Location-aware click prediction in mobile local search. In *CIKM '11*, 2011.

[27] B. Marlin. Modeling user rating profiles for collaborative filtering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[28] N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. In *WSDM '11*, pages 25–34, 2011.

[29] I. Ounis, C. Macdonald, M. de Rijke, G. Mishne, and I. Soboroff. Overview of the trec 2006 blog track. In *TREC*, 2006.

[30] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94*, pages 232–241, 1994.

[31] M. Sanderson. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375, 2010.

[32] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR '05*, pages 43–50, 2005.

[33] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR '96*, pages 21–29, 1996.

[34] T. Sohn, K. A. Li, W. G. Griswold, and J. D. Hollan. A diary study of mobile information needs. In *CHI '08*, pages 433–442, 2008.

[35] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW '04*, pages 675–684, 2004.

[36] B. Tan, X. Shen, and C. Zhai. Mining long-term search history to improve search accuracy. In *KDD '06*, pages 718–723, 2006.

[37] J. Teevan, S. T. Dumais, and E. Horvitz. Potential for personalization. *ACM Trans. Comput.-Hum. Interact.*, 17:4:1–4:31, April 2010.

[38] J. Teevan, A. Karlson, S. Amini, A. J. B. Brush, and J. Krumm. Understanding the importance of location, time, and people in mobile local search behavior. In *MobileHCI '11*, pages 77–80, 2011.

[39] L. Wang, C. Wang, X. Xie, J. Forman, Y. Lu, W.-Y. Ma, and Y. Li. Detecting dominant locations from search queries. In *SIGIR '05*, pages 424–431, 2005.

[40] Q. Wu, C. J. Burges, K. Svore, and J. Gao. Ranking, boosting, and model adaptation. Technical Report MSR-TR-2008-109, Microsoft Research, 2008.

[41] J. Yi, F. Maghoul, and J. Pedersen. Deciphering mobile search patterns: a study of yahoo! mobile search queries. In *WWW '08*, pages 257–266, 2008.

[42] X. Yi, H. Raghavan, and C. Leggetter. Discovering users' specific geo intention in web search. In *WWW '09*, pages 481–490, 2009.

[43] D. Zhang, R. Mao, H. Li, and J. Mao. How to count thumb-ups and thumb-downs: user-rating based ranking of items from an axiomatic perspective. In *ICTIR'11*, pages 238–249, 2011.