Learning User Interaction Models for Predicting Web Search Result Preferences

Eugene Agichtein
Microsoft Research
eugeneag@microsoft.com

Eric Brill
Microsoft Research
brill@microsoft.com

Susan Dumais
Microsoft Research
sdumais@microsoft.com

Robert Ragno
Microsoft Research
rragno@microsoft.com

ABSTRACT

Evaluating user preferences of web search results is crucial for search engine development, deployment, and maintenance. We present a real-world study of modeling the behavior of web search users to predict web search result preferences. Accurate modeling and interpretation of user behavior has important applications to ranking, click spam detection, web search personalization, and other tasks. Our key insight to improving robustness of interpreting implicit feedback is to model query-dependent deviations from the expected "noisy" user behavior. We show that our model of clickthrough interpretation improves prediction accuracy over state-of-the-art clickthrough methods. We generalize our approach to model user behavior beyond clickthrough, which results in higher preference prediction accuracy than models based on clickthrough information alone. We report results of a large-scale experimental evaluation that show substantial improvements over published implicit feedback interpretation methods.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process, relevance feedback.

General Terms

Algorithms, Measurement, Performance, Experimentation.

Keywords: Interpreting implicit relevance feedback, user behavior modeling, predicting relevance preferences

1. INTRODUCTION

Relevance measurement is crucial to web search and to information retrieval in general. Traditionally, search relevance is measured by using human assessors to judge the relevance of query-document pairs. However, explicit human ratings are expensive and difficult to obtain. At the same time, millions of people interact daily with web search engines, providing valuable *implicit* feedback through their interactions with the search results. If we could turn these interactions into relevance judgments, we could obtain large amounts of data for evaluating, maintaining, and improving information retrieval systems.

Recently, automatic or implicit relevance feedback has developed into an active area of research in the information retrieval community, at least in part due to an increase in available resources and to the rising popularity of web search. However, most traditional IR work was performed over controlled test collections and carefully-selected query sets and tasks. Therefore, it is not clear whether these techniques will work for general real-world web search. A significant distinction is that web search is not controlled. Individual users may behave irrationally or maliciously, or may not even be real users; all of this affects the data that can be gathered. But the amount of the user interaction data is orders of magnitude larger than anything available in a non-web-search setting. By using the *aggregated* behavior of large numbers of users (and not treating each user as an individual "expert") we can correct for the noise inherent in individual interactions, and generate relevance judgments that are more accurate than techniques not specifically designed for the web search setting.

Furthermore, observations and insights obtained in laboratory settings do not necessarily translate to real world usage. Hence, it is preferable to *automatically* induce feedback interpretation strategies from large amounts of user interactions. Automatically learning to interpret user behavior would allow systems to adapt to changing conditions, changing user behavior patterns, and different search settings. We present techniques to automatically interpret the collective behavior of users interacting with a web search engine to predict user preferences for search results. Our contributions include:

- A distributional model of user behavior, robust to noise within individual user sessions, that can recover relevance preferences from user interactions (Section 3).
- Extensions of existing clickthrough strategies to include richer browsing and interaction features (Section 4).
- A thorough evaluation of our user behavior models, as well as of previously published state-of-the-art techniques, over a large set of web search sessions (Sections 5 and 6).

We discuss our results and outline future directions and various applications of this work in Section 7, which concludes the paper.

2. BACKGROUND AND RELATED WORK

Ranking search results is a fundamental problem in information retrieval. The most common approaches in the context of the web use both the similarity of the query to the page content, and the overall quality of a page [3, 20]. A state-of-the-art search engine may use hundreds of features to describe a candidate page, employing sophisticated algorithms to rank pages based on these features. Current search engines are commonly tuned on human relevance judgments. Human annotators rate a set of pages for a query according to perceived relevance, creating the "gold standard" against which different ranking algorithms can be evaluated. Reducing the dependence on explicit human judgments by using *implicit relevance feedback* has been an active topic of research.

Several research groups have evaluated the relationship between implicit measures and user interest. In these studies, both reading

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

time and explicit ratings of interest are collected. Morita and Shinoda [14] studied the amount of time that users spent reading Usenet news articles and found that reading time could predict a user's interest levels. Konstan et al. [13] showed that reading time was a strong predictor of user interest in their GroupLens system. Oard and Kim [15] studied whether implicit feedback could substitute for explicit ratings in recommender systems. More recently, Oard and Kim [16] presented a framework for characterizing observable user behaviors using two dimensions—the underlying purpose of the observed behavior and the scope of the item being acted upon.

Goecks and Shavlik [8] approximated human labels by collecting a set of page activity measures while users browsed the World Wide Web. The authors hypothesized correlations between a high degree of page activity and a user's interest. While the results were promising, the sample size was small and the implicit measures were not tested against explicit judgments of user interest. Claypool et al. [6] studied how several implicit measures related to the interests of the user. They developed a custom browser called the *Curious Browser* to gather data, in a computer lab, about implicit interest indicators and to probe for explicit judgments of Web pages visited. Claypool et al. found that the time spent on a page, the amount of scrolling on a page, and the combination of time and scrolling have a strong positive relationship with explicit interest, while individual scrolling methods and mouse-clicks were not correlated with explicit interest. Fox et al. [7] explored the relationship between implicit and explicit measures in Web search. They built an instrumented browser to collect data and then developed Bayesian models to relate implicit measures and explicit relevance judgments for both individual queries and search sessions. They found that clickthrough was the most important individual variable but that predictive accuracy could be improved by using additional variables, notably dwell time on a page.

Joachims [9] developed valuable insights into the collection of implicit measures, introducing a technique based entirely on clickthrough data to learn ranking functions. More recently, Joachims et al. [10] presented an empirical evaluation of interpreting clickthrough evidence. By performing eye tracking studies and correlating predictions of their strategies with explicit ratings, the authors showed that it is possible to accurately interpret clickthrough events in a controlled, laboratory setting. A more comprehensive overview of studies of implicit measures is described in Kelly and Teevan [12].

Unfortunately, the extent to which existing research applies to real-world web search is unclear. In this paper, we build on previous research to develop robust user behavior interpretation models for the real web search setting.

3. LEARNING USER BEHAVIOR MODELS

As we noted earlier, real web search user behavior can be "noisy" in the sense that user behaviors are only probabilistically related to explicit relevance judgments and preferences. Hence, instead of treating each user as a reliable "expert", we aggregate information from many unreliable user search session traces. Our main approach is to model user web search behavior as if it were generated by two components: a "relevance" component – query-specific behavior influenced by the apparent result relevance, and a "background" component – users clicking indiscriminately. Our general idea is to model the *deviations* from the *expected user behavior*. Hence, in addition to basic features, which we will describe in detail in Section 3.2, we compute *derived* features that measure the deviation of the observed feature value for a given search result from the expected values for a result, with no query-dependent information. We motivate our intuitions with a particularly important behavior feature, result clickthrough, analyzed next, and then introduce our general model of user behavior that incorporates other user actions (Section 3.2).

3.1 A Case Study in Click Distributions

As we discussed, we aggregate statistics across many user sessions. A click on a result may mean that some user found the result summary promising; it could also be caused by people clicking indiscriminately. In general, individual user behavior, clickthrough and otherwise, is noisy, and cannot be relied upon for accurate relevance judgments. The data set is described in more detail in Section 5.2. For the present it suffices to note that we focus on a random sample of 3,500 queries that were randomly sampled from query logs. For these queries we aggregate click data over more than 120,000 searches performed over a three week period. We also have explicit relevance judgments for the top 10 results for each query.

Figure 3.1 shows the relative clickthrough frequency as a function of result position. The aggregated click frequency at result position p is calculated by first computing the frequency of a click at p for each query (i.e., approximating the probability that a randomly chosen click for that query would land on position p). These frequencies are then averaged across queries and normalized so that relative frequency of a click at the top position is 1. The resulting distribution agrees with previous observations that users click more often on top-ranked results. This reflects the fact that search engines do a reasonable job of ranking results as well as biases to click top results and noise – we attempt to separate these components in the analysis that follows.

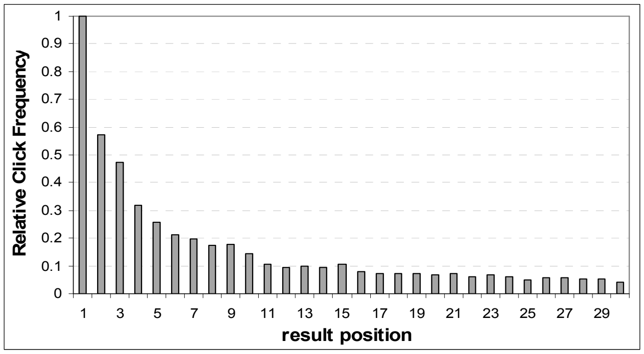

Figure 3.1: Relative click frequency for top 30 result positions over 3,500 queries and 120,000 searches.

First we consider the distribution of clicks for the relevant documents for these queries. Figure 3.2 reports the aggregated click distribution for queries with varying Position of Top Relevant document (PTR). While there are many clicks above the first relevant document for each distribution, there are clearly "peaks" in click frequency for the first relevant result. For example, for queries with top relevant result in position 2, the relative click frequency at that position (second bar) is higher than the click frequency at other positions for these queries. Nevertheless, many users still click on the non-relevant results in position 1 for such queries. This shows a stronger property of the bias in the click distribution towards top results – users click more often on results that are ranked higher, even when they are not relevant.

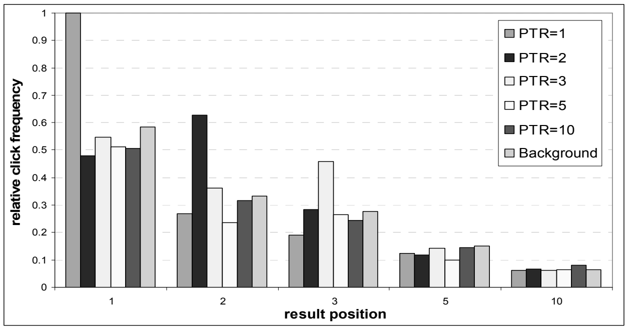

Figure 3.2: Relative click frequency for queries with varying PTR (Position of Top Relevant document).

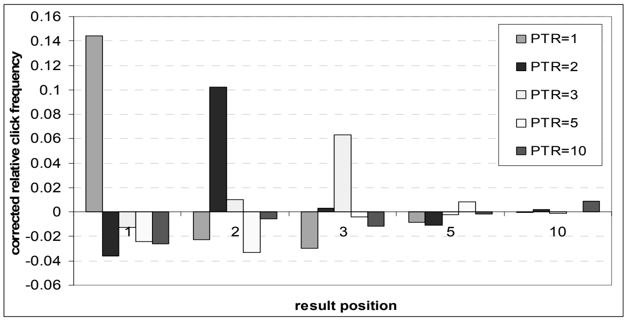

Figure 3.3: Relative corrected click frequency for relevant documents with varying PTR (Position of Top Relevant).

If we subtract the background distribution of Figure 3.1 from the "mixed" distribution of Figure 3.2, we obtain the distribution in Figure 3.3, where the remaining click frequency distribution can be interpreted as the relevance component of the results. Note that the *corrected* click distribution correlates closely with actual result relevance as explicitly rated by human judges.

3.2 Robust User Behavior Model

Clicks on search results comprise only a small fraction of the post-search activities typically performed by users. We now introduce our techniques for going beyond the clickthrough statistics and explicitly modeling *post-search* user behavior.

Although clickthrough distributions are heavily biased towards top results, we have just shown how the 'relevance-driven' click distribution can be recovered by correcting for the prior, background distribution. We conjecture that other aspects of user behavior (e.g., page dwell time) are similarly distorted. Our general model includes two feature types for describing user behavior: *direct* and *deviational* where the former is the directly measured values, and latter is deviation from the expected values estimated from the *overall* (query-independent) distributions for the corresponding directly observed features.

More formally, we postulate that the observed value o of a feature f for a query q and result r can be expressed as a mixture of two components:

$$o(q,r,f) = C(f) + rel(q,r,f) \tag{1}$$

where $C(f)$ is the prior "background" distribution for values of f aggregated across all queries, and *rel(q,r,f)* is the component of the behavior influenced by the relevance of the result *r*. As illustrated above with the clickthrough feature, if we subtract the background distribution (i.e., the expected clickthrough for a result at a given position) from the observed clickthrough frequency at a given position, we can approximate the relevance component of the clickthrough value[1]. In order to reduce the effect of individual user variations in behavior, we average observed feature values across all users and search sessions for each query-URL pair. This aggregation gives additional robustness of not relying on individual "noisy" user interactions.

In summary, the user behavior for a query-URL pair is represented by a feature vector that includes both the directly observed features and the derived, "corrected" feature values. We now describe the actual features we use to represent user behavior.

3.3 Features for Representing User Behavior

Our goal is to devise a sufficiently rich set of features that allow us to characterize when a user will be satisfied with a *web search result*. Once the user has submitted a query, they perform many different *actions* (reading snippets, clicking results, navigating, refining their query) which we capture and summarize. This information was obtained via opt-in client-side instrumentation from users of a major web search engine.

This rich representation of user behavior is similar in many respects to the recent work by Fox et al. [7]. An important difference is that many of our features are (by design) *query specific* whereas theirs was (by design) a general, query-independent model of user behavior. Furthermore, we include derived, distributional features computed as described above.

The features we use to represent user search interactions are summarized in Table 3.1. For clarity, we organize the features into the groups *Query-text*, *Clickthrough*, and *Browsing*.

Query-text features: Users decide which results to examine in more detail by looking at the result title, URL, and summary – in some cases, looking at the original document is not even necessary. To model this aspect of user experience we defined features to characterize the nature of the query and its relation to the snippet text. These include features such as overlap between the words in title and in query (TitleOverlap), the fraction of words shared by the query and the result summary (SummaryOverlap), etc.

Browsing features: Simple aspects of the user web page interactions can be captured and quantified. These features are used to characterize interactions with pages beyond the results page. For example, we compute how long users dwell on a page (TimeOnPage) or domain (TimeOnDomain), and the deviation of dwell time from expected page dwell time for a query. These features allows us to model intra-query diversity of page browsing behavior (e.g., navigational queries, on average, are likely to have shorter page dwell time than transactional or informational queries). We include both the direct features and the derived features described above.

Clickthrough features: Clicks are a special case of user interaction with the search engine. We include all the features necessary to "learn" the clickthrough-based strategies described in Sections 4.1 and 4.4. For example, for a query-URL pair we provide the number of clicks for the result (ClickFrequency), as well as whether there was a click on result below or above the current URL (IsClickBelow, IsClickAbove). The derived feature values such as ClickRelativeFrequency and ClickDeviation are computed as described in Equation 1.

[1] Of course, this is just a rough estimate, as the observed background distribution also includes the relevance component.

Query-text features	
TitleOverlap	Fraction of shared words between query and title
SummaryOverlap	Fraction of shared words between query and summary
QueryURLOverlap	Fraction of shared words between query and URL
QueryDomainOverlap	Fraction of shared words between query and domain
QueryLength	Number of tokens in query
QueryNextOverlap	Average fraction of words shared with next query
Browsing features	
TimeOnPage	Page dwell time
CumulativeTimeOnPage	Cumulative time for all subsequent pages after search
TimeOnDomain	Cumulative dwell time for this domain
TimeOnShortUrl	Cumulative time on URL prefix, dropping parameters
IsFollowedLink	1 if followed link to result, 0 otherwise
IsExactUrlMatch	0 if aggressive normalization used, 1 otherwise
IsRedirected	1 if initial URL same as final URL, 0 otherwise
IsPathFromSearch	1 if only followed links after query, 0 otherwise
ClicksFromSearch	Number of hops to reach page from query
AverageDwellTime	Average time on page for this query
DwellTimeDeviation	Deviation from overall average dwell time on page
CumulativeDeviation	Deviation from average cumulative time on page
DomainDeviation	Deviation from average time on domain
ShortURLDeviation	Deviation from average time on short URL
Clickthrough features	
Position	Position of the URL in Current ranking
ClickFrequency	Number of clicks for this query, URL pair
ClickRelativeFrequency	Relative frequency of a click for this query and URL
ClickDeviation	Deviation from expected click frequency
IsNextClicked	1 if there is a click on next position, 0 otherwise
IsPreviousClicked	1 if there is a click on previous position, 0 otherwise
IsClickAbove	1 if there is a click above, 0 otherwise
IsClickBelow	1 if there is click below, 0 otherwise

Table 3.1: Features used to represent post-search interactions for a given query and search result URL

3.4 Learning a Predictive Behavior Model

Having described our features, we now turn to the actual method of mapping the features to user preferences. We attempt to *learn* a general implicit feedback interpretation strategy *automatically* instead of relying on heuristics or insights. We consider this approach to be preferable to heuristic strategies, because we can always mine more data instead of relying (only) on our intuition and limited laboratory evidence. Our general approach is to train a classifier to induce weights for the user behavior features, and consequently derive a predictive model of user preferences. The training is done by comparing a wide range of implicit behavior measures with explicit user judgments for a set of queries.

For this, we use a large random sample of queries in the search query log of a popular web search engine, the sets of results (identified by URLs) returned for each of the queries, and any explicit relevance judgments available for each query/result pair. We can then analyze the user behavior for all the instances where these queries were submitted to the search engine.

To learn the mapping from features to relevance preferences, we use a scalable implementation of neural networks, RankNet [4], capable of learning to *rank* a set of given items. More specifically, for each judged query we check if a result link has been judged. If so, the label is assigned to the query/URL pair and to the corresponding feature vector for that search result. These vectors of feature values corresponding to URLs judged relevant or non-relevant by human annotators become our training set. RankNet has demonstrated excellent performance in learning to rank objects in a supervised setting, hence we use RankNet for our experiments.

4. PREDICTING USER PREFERENCES

In our experiments, we explore several models for predicting user preferences. These models range from using no implicit user feedback to using all available implicit user feedback.

Ranking search results to predict user preferences is a fundamental problem in information retrieval. Most traditional IR and web search approaches use a combination of page and link features to rank search results, and a representative state-of-the-art ranking system will be used as our baseline ranker (Section 4.1). At the same time, user interactions with a search engine provide a wealth of information. A commonly considered type of interaction is user clicks on search results. Previous work [9], as described above, also examined which results were skipped (e.g., 'skip above' and 'skip next') and other related strategies to induce preference judgments from the users' skipping over results and not clicking on following results. We have also added refinements of these strategies to take into account the variability observed in realistic web scenarios.. We describe these strategies in Section 4.2.

As clickthroughs are just one aspect of user interaction, we extend the relevance estimation by introducing a machine learning model that incorporates clicks as well as other aspects of user behavior, such as follow-up queries and page dwell time (Section 4.3). We conclude this section by briefly describing our "baseline" – a state-of-the-art ranking algorithm used by an operational web search engine.

4.1 Baseline Model

A key question is whether browsing behavior can provide information absent from existing explicit judgments used to train an existing ranker. For our baseline system we use a state-of-the-art page ranking system currently used by a major web search engine. Hence, we will call this system *Current* for the subsequent discussion. While the specific algorithms used by the search engine are beyond the scope of this paper, the algorithm ranks results based on hundreds of features such as query to document similarity, query to anchor text similarity, and intrinsic page quality. The Current web search engine rankings provide a strong system for comparison and experiments of the next two sections.

4.2 Clickthrough Model

If we assume that every user click was motivated by a rational process that selected the most promising result summary, we can then interpret each click as described in Joachims et al.[10]. By studying eye tracking and comparing clicks with explicit judgments, they identified a few basic strategies. We discuss the two strategies that performed best in their experiments, Skip Above and Skip Next.

> **Strategy *SA* (Skip Above):** For a set of results for a query and a clicked result at position *p*, all *unclicked* results ranked above *p* are predicted to be less relevant than the result at *p*.

In addition to information about results *above* the clicked result, we also have information about the result immediately following the clicked one. Eye tracking study performed by Joachims et al. [10] showed that users usually consider the result immediately following the clicked result in current ranking. Their "Skip Next" strategy uses this observation to predict that a result following the clicked result at *p* is less relevant than the clicked result, with accuracy comparable to the *SA* strategy above. For better coverage, we combine the *SA* strategy with this extension to derive the *Skip Above + Skip Next* strategy:

> **Strategy *SA+N* (Skip Above + Skip Next):** This strategy predicts all un-clicked results *immediately following* a clicked result as less relevant than the clicked result, and combines these predictions with those of the SA strategy above.

We experimented with variations of these strategies, and found that SA+N outperformed both SA and the original Skip Next strategy, so we will consider the SA and SA+N strategies in the rest of the paper. These strategies are motivated and empirically tested for individual users in a laboratory setting. As we will show, these strategies do not work as well in real web search setting due to inherent inconsistency and noisiness of individual users' behavior.

The general approach for using our clickthrough models directly is to filter clicks to those that reflect higher-than-chance click frequency. We then use the same SA and SA+N strategies, but *only for clicks that have higher-than-expected frequency* according to our model. For this, we estimate the relevance component *rel(q,r,f)* of the observed clickthrough feature *f* as the deviation from the expected (background) clickthrough distribution $C(f)$.

> **Strategy *CD (deviation d)*:** For a given query, compute the observed click frequency distribution *o(r, p)* for all results *r* in positions *p*. The click deviation for a result *r* in position *p*, *dev(r, p)* is computed as:
>
> $$dev(r,p) = o(r,p) - C(p)$$
>
> where *C(p)* is the expected clickthrough at position *p*. If *dev(r,p)>**d**,* retain the click as input to the SA+N strategy above, and apply SA+N strategy over the filtered set of click events.

The choice of ***d*** selects the tradeoff between recall and precision. While the above strategy extends SA and SA+N, it still assumes that a (filtered) clicked result is preferred over all unclicked results presented to the user above a clicked position. However, for informational queries, multiple results may be clicked, with varying frequency. Hence, it is preferable to individually compare results for a query by considering the *difference* between the estimated relevance components of the click distribution of the corresponding query results. We now define a generalization of the previous clickthrough interpretation strategy:

> **Strategy *CDiff (margin m)*:** Compute deviation *dev(r,p)* for each result $r_1 ... r_n$ in position p. For each pair of results r_i and r_j, predict preference of r_i over r_j iff $dev(r_i,p_i)$-$dev(r_i,p_j)$>***m***.

As in CD, the choice of ***m*** selects the tradeoff between recall and precision. The pairs may be preferred in the original order or in reverse of it. Given the margin, two results might be effectively indistinguishable, but only one can possibly be preferred over the other. Intuitively, CDiff generalizes the skip idea above to include cases where the user "skipped" (i.e., clicked less than expected) on u_j and "preferred" (i.e., clicked more than expected) on u_i. Furthermore, this strategy allows for differentiation within the set of clicked results, making it more appropriate to noisy user behavior.

CDiff and CD are complimentary. CDiff is a generalization of the clickthrough frequency model of CD, but it ignores the positional information used in CD. Hence, combining the two strategies to improve coverage is a natural approach:

> **Strategy *CD+CDiff (deviation d, margin m)*:** Union of CD and CDiff predictions.

Other variations of the above strategies were considered, but these five methods cover the range of observed performance.

4.3 General User Behavior Model

The strategies described in the previous section generate orderings based solely on observed clickthrough frequencies. As we discussed, clickthrough is just one, albeit important, aspect of user interactions with web search engine results. We now present our general strategy that relies on the *automatically derived* predictive user behavior models (Section 3).

> The ***UserBehavior*** **Strategy:** For a given query, each result is represented with the features in Table 3.1. Relative user preferences are then estimated using the learned user behavior model described in Section 3.4.

Recall that to learn a predictive behavior model we used the features from Table 3.1 along with explicit relevance judgments as input to RankNet which learns an optimal weighting of features to predict preferences.

This strategy models user interaction with the search engine, allowing it to benefit from the wisdom of crowds interacting with the results and the pages beyond. As our experiments in the subsequent sections demonstrate, modeling a richer set of user interactions beyond clickthroughs results in more accurate predictions of user preferences.

5. EXPERIMENTAL SETUP

We now describe our experimental setup. We first describe the methodology used, including our evaluation metrics (Section 5.1). Then we describe the datasets (Section 5.2) and the methods we compared in this study (Section 5.3).

5.1 Evaluation Methodology and Metrics

Our evaluation focuses on the *pairwise* agreement between preferences for results. This allows us to compare to previous work [9,10]. Furthermore, for many applications such as tuning ranking functions, pairwise preference can be used directly for training [1,4,9]. The evaluation is based on comparing preferences predicted by various models to the "correct" preferences derived from the explicit user relevance judgments. We discuss other applications of our models beyond web search ranking in Section 7.

To create our set of "test" pairs we take each query and compute the cross-product between all search results, returning preferences for pairs according to the order of the associated relevance labels. To avoid ambiguity in evaluation, we discard all ties (i.e., pairs with equal label).

In order to compute the accuracy of our preference predictions with respect to the correct preferences, we adapt the standard Recall and Precision measures [20]. While our task of computing pairwise agreement is different from the absolute relevance ranking task, the metrics are used in the similar way. Specifically, we report the average *query* recall and precision. For our task, Query Precision and Query Recall for a query *q* are defined as:

- Query Precision: Fraction of predicted preferences for results for *q* that agree with preferences obtained from explicit human judgment.
- Query Recall: Fraction of preferences obtained from explicit human judgment for *q* that were correctly predicted.

The overall Recall and Precision are computed as the average of Query Recall and Query Precision, respectively. A drawback of this evaluation measure is that some preferences may be more valuable than others, which pairwise agreement does not capture. We discuss this issue further when we consider extensions to the current work in Section 7.

5.2 Datasets

For evaluation we used 3,500 queries that were randomly sampled from query logs(for a major web search engine. For each query the top 10 returned search results were manually rated on a 6-point scale by trained judges as part of ongoing relevance improvement effort. In addition for these queries we also had user interaction data for more than 120,000 instances of these queries.

The user interactions were harvested from anonymous browsing traces that immediately followed a query submitted to the web search engine. This data collection was part of voluntary opt-in feedback submitted by users from October 11 through October 31. These three weeks (21 days) of user interaction data was filtered to include only the users in the English-U.S. market.

In order to better understand the effect of the amount of user interaction data available for a query on accuracy, we created subsets of our data (Q1, Q10, and Q20) that contain different amounts of interaction data:

- **Q1:** Human-rated queries with at least 1 click on results recorded (**3500** queries, 28,093 query-URL pairs)
- **Q10:** Queries in Q1 with at least 10 clicks (**1300** queries, 18,728 query-URL pairs).
- **Q20**: Queries in Q1 with at least 20 clicks (**1000** queries total, 12,922 query-URL pairs).

These datasets were collected as part of normal user experience and hence have different characteristics than previously reported datasets collected in laboratory settings. Furthermore, the data size is order of magnitude larger than any study reported in the literature.

5.3 Methods Compared

We considered a number of methods for comparison. We compared our UserBehavior model (Section 4.3) to previously published implicit feedback interpretation techniques and some variants of these approaches (Section 4.2), and to the current search engine ranking based on query and page features alone (Section 4.1). Specifically, we compare the following strategies:

- **SA**: The "Skip Above" clickthrough strategy (Section 4.2)
- **SA+N**: A more comprehensive extension of SA that takes better advantage of current search engine ranking.
- **CD**: Our refinement of SA+N that takes advantage of our mixture model of clickthrough distribution to select "trusted" clicks for interpretation (Section 4.2).
- **CDiff**: Our generalization of the CD strategy that explicitly uses the relevance component of clickthrough probabilities to induce preferences between search results (Section 4.2).
- **CD+CDiff**: The strategy combining CD and CDiff as the union of predicted preferences from both (Section 4.2).
- **UserBehavior**: We order predictions based on decreasing highest score of any page. In our preliminary experiments we observed that higher ranker scores indicate higher "confidence" in the predictions. This heuristic allows us to do graceful recall-precision tradeoff using the score of the highest ranked result to threshold the queries (Section 4.3)
- **Current**: Current search engine ranking (section 4.1). Note that the Current ranker implementation was trained over a superset of the rated query/URL pairs in our datasets, but using the same "truth" labels as we do for our evaluation.

Training/Test Split: The only strategy for which splitting the datasets into training and test was required was the UserBehavior method. To evaluate UserBehavior we train and validate on 75% of labeled queries, and test on the remaining 25%. The sampling was done per query (i.e., all results for a chosen query were included in the respective dataset, and there was no overlap in queries between training and test sets).

It is worth noting that both the ad-hoc SA and SA+N, as well as the distribution-based strategies (CD, CDiff, and CD+CDiff), do not require a separate training and test set, since they are based on heuristics for detecting "anomalous" click frequencies for results. Hence, all strategies except for UserBehavior were tested on the full set of queries and associated relevance preferences, while UserBehavior was tested on a randomly chosen hold-out subset of the queries as described above. To make sure we are not favoring UserBehavior, we also tested all other strategies on the same hold-out test sets, resulting in the same accuracy results as testing over the complete datasets.

6. RESULTS

We now turn to experimental evaluation of predicting relevance preference of web search results. Figure 6.1 shows the recall-precision results over the **Q1** query set (Section 5.2). The results indicate that previous click interpretation strategies, SA and SA+N perform suboptimally in this setting, exhibiting precision 0.627 and 0.638 respectively. Furthermore, there is no mechanism to do recall-precision trade-off with SA and SA+N, as they do not provide prediction confidence. In contrast, our clickthrough distribution-based techniques CD and CD+CDiff exhibit somewhat higher precision than SA and SA+N (0.648 and 0.717 at Recall of 0.08, maximum achieved by SA or SA+N).

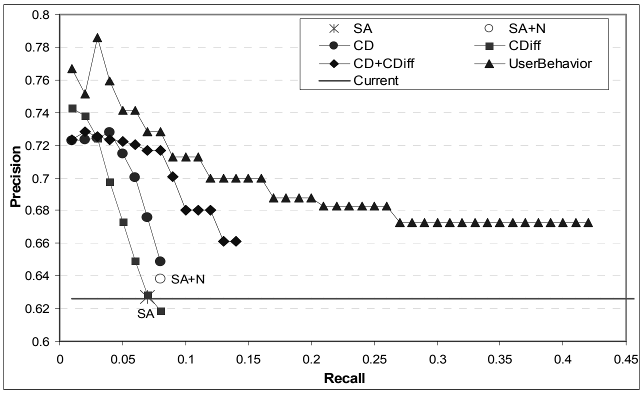

Figure 6.1: Precision vs. Recall of SA, SA+N, CD, CDiff, CD+CDiff, UserBehavior, and Current relevance prediction methods over the Q1 dataset.

Interestingly, CDiff alone exhibits precision equal to SA (0.627) at the same recall at 0.08. In contrast, by combining CD and CDiff strategies (CD+CDiff method) we achieve the best performance of all clickthrough-based strategies, exhibiting precision of above 0.66 for recall values up to 0.14, and higher at lower recall levels. Clearly, aggregating and intelligently interpreting clickthroughs, results in significant gain for realistic web search, than previously described strategies. However, even the CD+CDiff clickthrough interpretation strategy can be improved upon by *automatically* learning to interpret the aggregated clickthrough evidence.

But first, we consider the best performing strategy, UserBehavior. Incorporating post-search navigation history in addition to clickthroughs (Browsing features) results in the highest recall and precision among all methods compared. Browse exhibits precision of above 0.7 at recall of 0.16, significantly outperforming our Baseline and clickthrough-only strategies. Furthermore, Browse is able to achieve high recall (as high as 0.43) while maintaining precision (0.67) significantly higher than the baseline ranking.

To further analyze the value of different dimensions of implicit feedback modeled by the UserBehavior strategy, we consider each group of features in isolation. Figure 6.2 reports Precision vs. Recall for each feature group. Interestingly, Query-text alone has low accuracy (only marginally better than Random). Furthermore, Browsing features alone have higher precision (with lower maximum recall achieved) than considering all of the features in our UserBehavior model. Applying different machine learning methods for combining classifier predictions may increase performance of using all features for all recall values.

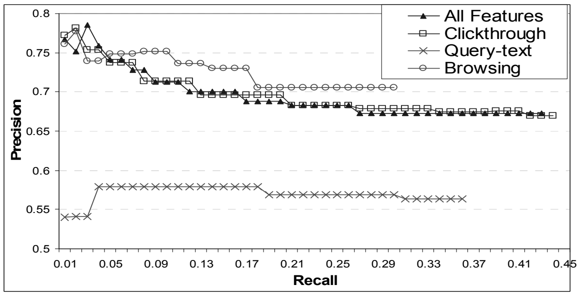

Figure 6.2: Precision vs. recall for predicting relevance with each group of features individually.

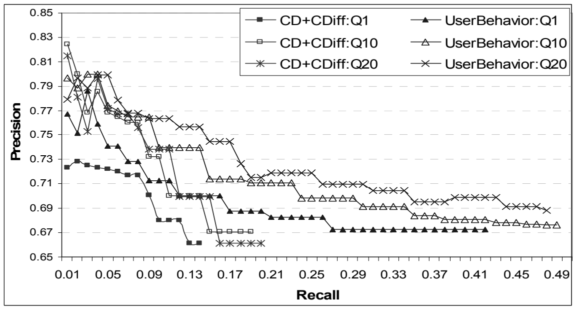

Figure 6.3: Recall vs. Precision of CD+CDiff and UserBehavior for query sets Q1, Q10, and Q20 (queries with at least 1, at least 10, and at least 20 clicks respectively).

Interestingly, the ranker trained over Clickthrough-only features achieves substantially higher recall and precision than human-designed clickthrough-interpretation strategies described earlier. For example, the clickthrough-trained classifier achieves 0.67 precision at 0.42 Recall vs. the maximum recall of 0.14 achieved by the CD+CDiff strategy.

Our clickthrough and user behavior interpretation strategies rely on extensive user interaction data. We consider the effects of having sufficient interaction data available for a query before proposing a re-ranking of results for that query. Figure 6.3 reports recall-precision curves for the CD+CDiff and UserBehavior methods for different test query sets with at least 1 click (Q1), 10 clicks (Q10) and 20 clicks (Q20) available per query. Not surprisingly, CD+CDiff improves with more clicks. This indicates that accuracy will improve as more user interaction histories become available, and more queries from the Q1 set will have comprehensive interaction histories. Similarly, the UserBehavior strategy performs better for queries with 10 and 20 clicks, although the improvement is less dramatic than for CD+CDiff. For queries with sufficient clicks, CD+CDiff exhibits precision comparable with Browse at lower recall.

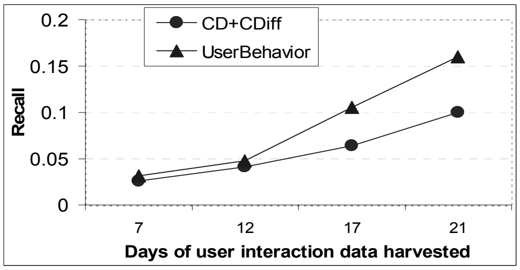

Figure 6.4: Recall of CD+CDiff and UserBehavior strategies at fixed minimum precision 0.7 for varying amounts of user activity data (7, 12, 17, 21 days).

Our techniques often do not make relevance predictions for search results (i.e., if no interaction data is available for the lower-ranked results), consequently maintaining higher precision at the expense of recall. In contrast, the current search engine *always* makes a prediction for every result for a given query. As a consequence, the recall of Current is high (0.627) at the expense of lower precision As another dimension of acquiring training data we consider the learning curve with respect to amount (days) of training data available. Figure 6.4 reports the Recall of CD+CDiff and UserBehavior strategies for varying amounts of training data collected over time. We fixed minimum precision for both strategies at 0.7 as a point substantially higher than the baseline (0.625). As expected, Recall of both strategies improves quickly with more days of interaction data examined.

We now briefly summarize our experimental results. We showed that by intelligently aggregating user clickthroughs across queries and users, we can achieve higher accuracy on predicting user preferences than previous strategies. Because of the skewed distribution of user clicks, our clickthrough-only strategies have high precision, but low recall (i.e., do not attempt to predict relevance of many search results). Nevertheless, our CD+CDiff clickthrough strategy outperforms most recent state-of-the-art results by a large margin (0.72 precision for CD+CDiff vs. 0.64 for SA+N) at the highest recall level of SA+N.

Furthermore, by considering the comprehensive UserBehavior features that model user interactions *after* the search and beyond the initial click, we can achieve substantially higher precision and recall than considering clickthrough alone. Our UserBehavior strategy achieves recall of over 0.43 with precision of over 0.67 (with much higher precision at lower recall levels), substantially outperforming the current search engine preference ranking and all other implicit feedback interpretation methods.

7. CONCLUSIONS AND FUTURE WORK

Our paper is the first, to our knowledge, to interpret post-search user behavior to estimate user preferences in a real web search setting. We showed that our robust models result in higher prediction accuracy than previously published techniques.

We introduced new, robust, probabilistic techniques for interpreting clickthrough evidence by aggregating across users and queries. Our methods result in clickthrough interpretation substantially more accurate than previously published results not specifically designed for web search scenarios. Our methods' predictions of relevance preferences are substantially more accurate than the current state-of-the-art search result ranking that does not consider user interactions. We also presented a general model for interpreting post-search user behavior that incorporates clickthrough, browsing, and query features. By considering the complete search experience *after* the initial query and click, we demonstrated prediction accuracy far exceeding that of interpreting only the limited clickthrough information.

Furthermore, we showed that automatically *learning* to interpret user behavior results in substantially better performance than the human-designed ad-hoc clickthrough interpretation strategies. Another benefit of automatically learning to interpret user behavior is that such methods can adapt to changing conditions and changing user profiles. For example, the user behavior model on intranet search may be different from the web search behavior. Our general UserBehavior method would be able to adapt to these changes by automatically learning to map new behavior patterns to explicit relevance ratings.

A natural application of our preference prediction models is to improve web search ranking [1]. In addition, our work has many potential applications including click spam detection, search abuse detection, personalization, and domain-specific ranking. For example, our automatically derived behavior models could be trained on examples of search abuse or click spam behavior instead of relevance labels. Alternatively, our models could be used directly to detect anomalies in user behavior – either due to abuse or to operational problems with the search engine.

While our techniques perform well *on average,* our assumptions about clickthrough distributions (and learning the user behavior models) may not hold equally well for all queries. For example, queries with divergent access patterns (e.g., for ambiguous queries with multiple meanings) may result in behavior inconsistent with the model learned for all queries. Hence, clustering queries and learning different predictive models for each query type is a promising research direction. Query distributions also change over time, and it would be productive to investigate how that affects the predictive ability of these models. Furthermore, some predicted preferences may be more valuable than others, and we plan to investigate different metrics to capture the utility of the predicted preferences.

As we showed in this paper, using the "wisdom of crowds" can give us an accurate interpretation of user interactions even in the inherently noisy web search setting. Our techniques allow us to automatically predict relevance preferences for web search results with accuracy greater than the previously published methods. The predicted relevance preferences can be used for automatic relevance evaluation and tuning, for deploying search in new settings, and ultimately for improving the overall web search experience.

8. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais, Improving Web Search Ranking by Incorporating User Behavior, in *Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR),* 2006

[2] J. Allan. HARD Track Overview in TREC 2003: High Accuracy Retrieval from Documents. In *Proceedings of TREC 2003*, 24-37, 2004.

[3] S. Brin and L. Page, The Anatomy of a Large-scale Hypertextual Web Search Engine,. In *Proceedings of WWW7*, 107-117, 1998.

[4] C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, Learning to Rank using Gradient Descent, in *Proceedings of the International Conference on Machine Learning (ICML)*, 2005

[5] D.M. Chickering, The WinMine Toolkit, *Microsoft Technical Report MSR-TR-2002-103*, 2002

[6] M. Claypool, D. Brown, P. Lee and M. Waseda. Inferring user interest, in *IEEE Internet Computing*. 2001

[7] S. Fox, K. Karnawat, M. Mydland, S. T. Dumais and T. White. Evaluating implicit measures to improve the search experience. In *ACM Transactions on Information Systems*, 2005

[8] J. Goecks and J. Shavlick. Learning users' interests by unobtrusively observing their normal behavior. In *Proceedings of the IJCAI Workshop on Machine Learning for Information Filtering*. 1999.

[9] T. Joachims, Optimizing Search Engines Using Clickthrough Data, in *Proceedings of the ACM Conference on Knowledge Discovery and Datamining (SIGKDD)*, 2002

[10] T. Joachims, L. Granka, B. Pang, H. Hembrooke and G. Gay, Accurately Interpreting Clickthrough Data as Implicit Feedback, in *Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR)*, 2005

[11] T. Joachims, Making Large-Scale SVM Learning Practical. Advances in Kernel Methods, in *Support Vector Learning*, MIT Press, 1999

[12] D. Kelly and J. Teevan, Implicit feedback for inferring user preference: A bibliography. In *SIGIR Forum*, *2003*

[13] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon and J. Riedl. GroupLens: Applying collaborative filtering to usenet news. In *Communications of ACM*, 1997.

[14] M. Morita, and Y. Shinoda, Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR)*, 1994

[15] D. Oard and J. Kim. Implicit feedback for recommender systems. in *Proceedings of AAAI Workshop on Recommender Systems*. 1998

[16] D. Oard and J. Kim. Modeling information content using observable behavior. In *Proceedings of the 64th Annual Meeting of the American Society for Information Science and Technology*. 2001

[17] P. Pirolli, The Use of Proximal Information Scent to Forage for Distal Content on the World Wide Web. In *Working with Technology in Mind: Brunswikian. Resources for Cognitive Science and Engineering*, Oxford University Press, 2004

[18] F. Radlinski and T. Joachims, Query Chains: Learning to Rank from Implicit Feedback, in *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD),* ACM, 2005

[19] F. Radlinski and T. Joachims, Evaluating the Robustness of Learning from Implicit Feedback, in the *ICML Workshop on Learning in Web Search*, 2005

[20] G. Salton and M. McGill. Introduction to modern information retrieval. McGraw-Hill, 1983

[21] E.M. Voorhees, D. Harman, Overview of TREC, 2001