Learning User Interaction Models for Predicting Web Search Result Preferences

Eugene Agichtein, Eric Brill, Susan Dumais, Robert Ragno

A presentation by Michail Kazimianec

The Goal of the Paper

- To present a real-world study of modeling the behavior of web search users to predict web search result preferences
- To improve robustness of interpreting implicit feedback (to model query-dependent deviations from the expected user behavior)

Problem

- Relevance measurement is crucial to web search and to information retrieval
- Traditionally, search relevance is measured by using human assessors to judge relevance of query-document pairs
- Explicit human ratings are expensive and difficult to obtain

"Implicit" Feedback Is the Key Aspect

- People interact with web search engines
- People provide valuable *implicit* feedback through their interactions
- Turning interactions into relevance judgments provides a possibility to obtain lots of data for:
  a. evaluating
  b. maintaining
  c. improving information retrieval systems

Problems of Traditional IR Works

- Works were performed over:
  a. controlled test collections
  b. carefully-selected query sets
  c. carefully-selected tasks
- Observations and insights obtained in laboratory settings may not translate to real world usage

Problems for Evaluating Preferences

- Web search is not controlled
- Individual users may behave irrationally or maliciously
- Users may not be real users

Contributions

- A distributional model of user behavior, robust to noise within individual user sessions, that recovers relevance preferences from user interactions
- Extensions of existing clickthrough strategies to include richer browsing and interaction features
- An evaluation of proposed behavior models and other state-of-the-art techniques, over a large set of web search sessions

Current techniques focuse on...

- Ranking of search results
- Using both the similarity of the query to the page content, and overall quality of a page
- On human relevance judgment

User Behavior Models

- To aggregate information from many user session traces
- To model user web search behavior on basis of:

  a. relevance component – query-specific behavior influenced by apparent result relevance

  b. background component – users clicking disorderly
- To model the *deviations* from expected user behavior based on *derived* features

Example

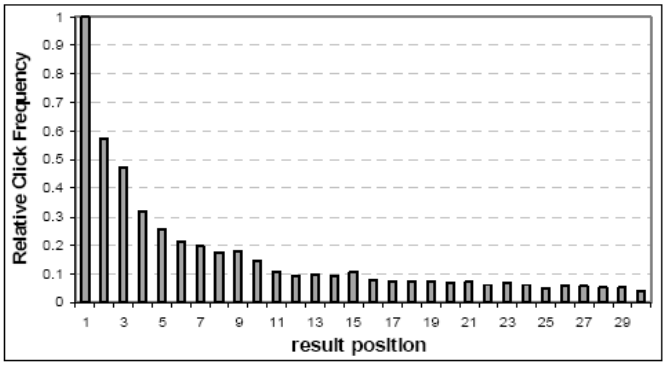

Figure 3.1: Relative click frequency for top 30 result positions over 3,500 queries and 120,000 searches.

Relative Corrected Click Frequency

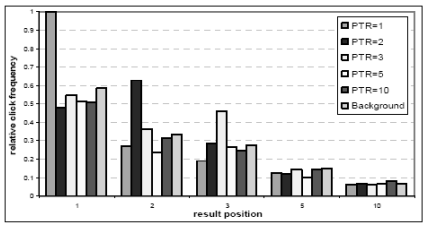

Figure 3.2: Relative click frequency for queries with varying PTR (Position of Top Relevant document).

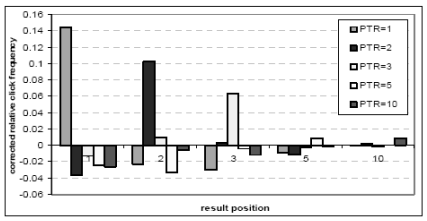

Figure 3.3: Relative corrected click frequency for relevant documents with varying PTR (Position of Top Relevant).

Robust User Behavior Model

- General model includes two feature types for describing user behavior:

  direct (directly measured values)

  deviational (deviation from the expected values estimated from the overall distributions for the corresponding directly observed features)

Robust User Behavior Model

- The observed value *o* of a feature *f* for a query *q* and result *r* can be expressed as a mixture of two components

$$o(q,r,f) = C(f) + rel(q,r,f)$$

C(f) - the prior "background" distribution for values of *f* aggregated across all queries

rel(q,r,f) - the component of the behavior influenced by the relevance of the result *r*

Features for Representing User Behavior

- Query-text features
  - Characterize the nature of the query and its relation to the snippet text (TitleOverlap, SummaryOverlap, etc.)
- Browsing features
  - Characterize interactions with pages beyond the results page (TimeOnPage, TimeOnDomain)
- Clickthrough features
  - (ClickFrequency, IsClickBelow, IsClickAbove)

Features for Representing User Behavior

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

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

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

Learning a Predictive Behavior Model with RankNet

- The general approach is
  - To train a classifier to induce weights for the user behavior features,
  - To derive consequently a predictive model of user preferences.
  - To compare a wide range of implicit behavior measures with explicit user judgments for a set of queries.
  - To use a scalable implementation of neural networks, RankNet, to learn the mapping from features to relevance preferences.

Learning a Predictive Behavior Model with RankNet

- Ranknet *rank* a set of given items
- Training set:
  - For each judged query it is checked if a result link has been judged.
  - If so, the label is assigned to the query/URL pair and for corresponding feature vector for that search result.
  - These vectors of feature values corresponding to URLs judged relevant or non-relevant by human annotators become a training set

Machine Learning Model

- Extend the relevance estimation by introducing a machine learning model that incorporates:
  - Clicks
  - Follow-up queries
  - Page dwell time

Clickthrough Model

- 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* predicted to be less relevant than the result at *p*
- Strategy SA+N (Skip above + Skip Next)
  - All unclicked results that *immediately follow* a clicked result are predicted to be less relevant than the clicked result.
  - This strategy combines predictions with the predictions of the SA strategy.

Clickthrough Model

- Use SA and SA+N strategies only for clicks that have higher-than-expected frequency
- For this estimate the relevance component rel(q,r) of the observed clickthrough feature f as the deviation from the expected clickthrough distributions C(f).

Clickthrough Model

- 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

    $$dev(r,p) = o(r,p) - C(p)$$

    C(p) – expected clickthrough at position *p*.
  - If $dev(r,p) > d$,
    - retain the click as input to the SA+N strategy,
    - apply SA+N strategy over the filtered set of click events.

    The choice of *d* selects the tradeoff between recall and precision.

Clickthrough Model

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

    The choice of *m* selects the tradeoff between recall and precision.
- Strategy CD + CDiff (deviation *d*, margin *m*)
  - Union of CD and CDiff predictions

General User Behavior Model

- The User Behavior Strategy
  - For a given query, each result is represented with the Query-text, Browsing and Clickthrough features. Relative user preferences are estimated using learned predictive behavior model.

  This strategy models user interaction with search engine, allowing it to benefit from the wisdom of crowds interacting with the results and the pages beyond.

Datasets

- 3500 queries
- Each query – the top 10 returned search results
- Results manually rated on a 6-point scale by trained judges
- User interaction data for more than 120000 instances of the queries (21 days)

Datasets

- Subsets of the 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)

Methods Compared

- SA – the “Skip Above” strategy
- SA+N – an extension of SA
- CD – the refinement of SA+N that takes advantage of the mixture model of clickthrough distribution to select “trusted” clicks for interpretation
- CDiff – the generalization of the CD strategy that uses the relevance component of clickthrough probabilities to induce preferences between search results
- CD+CDiff – the union of CD and CDiff
- UserBehavior – ordering of predictions based on decreasing highest score of any page
- Current – Current search engine ranking

Results

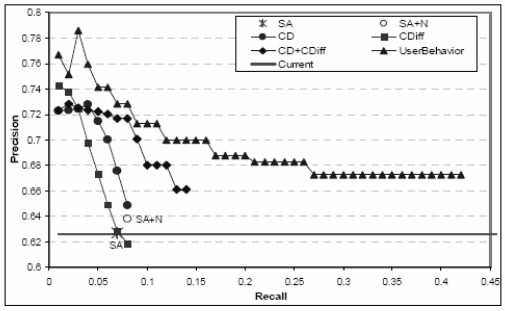

Figure 6.1: Precision vs. Recall of SA, SA+N, CD, CDiff, CD+CDiff, UserBehavior, and Current relevance prediction methods over the Q1 dataset.

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

Results

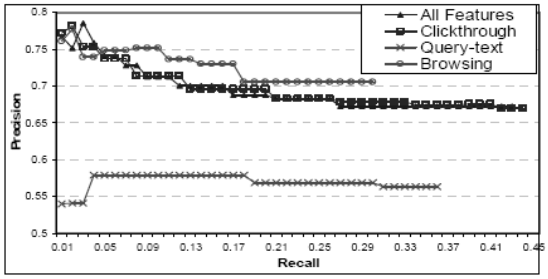

Figure 6.2: Precision vs. recall for predicting relevance with each group of features individually.

Results

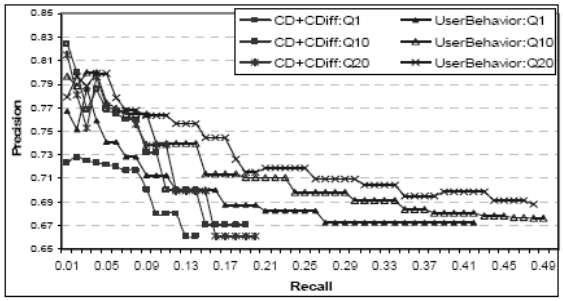

Figure 6.3: Recall vs. Precision of CD+CDiff and UserBehavior for query sets Q1, Q10, and Q20 (queries with at least 1, at least 10, and at least 20 clicks respectively).

Results

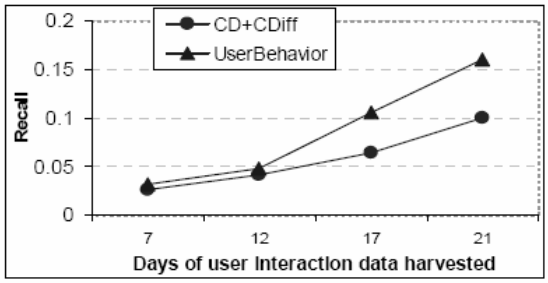

Figure 6.4: Recall of CD+CDiff and UserBehavior strategies at fixed minimum precision 0.7 for varying amounts of user activity data (7, 12, 17, 21 days).

Conclusions

- Interpret post-search user behavior to estimate user preferences in a real web search setting
- Robust models result is in higher prediction accuracy than previously published approaches
- Automatically *learning* to interpret user behavior results is in substantially better performance than human-designed ad-hoc clickthrough interpretation strategies
- Models can be used to detect anomalies in user behavior