Enhancing Web Search by Mining Search and Browse Logs

Daxin Jiang
Microsoft Research Asia

Jian Pei
Simon Fraser University

Hang Li
Microsoft Research Asia
Conventional Search System

Search System

Query-Document Matching

Query Representation
Query Understanding

Query

Doc Representation
Doc Understanding

Documents

Queries
Personalized/Context-Aware Search System

Search System

Query-Document Matching [Personalized/Context-aware]

Query Representation
Query Understanding

User/Context Profiles
User Understanding

Doc Representation
Doc Understanding

Queries
Users
Documents
Search and Browse Log Mining

Search System

Query-Document Matching [Personalized/Context-aware]

Monitor & Feedback

Search/Browse Logs

Enhance

Query Representation
Query Understanding

User/Context Profiles
User Understanding

Doc Representation
Doc Understanding
Agenda

• Section 1: Introduction (20’) [Hang Li]
• Section 2: Query understanding (40’) [Hang Li]
• Section 3: Document understanding (30’) [Jian Pei]
• Section 4: Query-document matching (30’) [Jian Pei]
• Section 5: User understanding (20’) [Daxin Jiang]
• Section 6: Monitoring & feedback (20’) [Daxin Jiang]
• Section 7: Challenges and future trends (20’) [Daxin Jiang]
Road Map

Search System

Query-Document Matching
[Personalized/Context-aware]

Sec 4

Enhance

Query Representation
Query Understanding

Sec 2

Enhance

User/Context Profiles
User Understanding

Sec 5

Enhance

Doc Representation
Doc Understanding

➢ Search & browse logs
➢ Log mining applications
➢ Frequently-used data summarizations

Sec 1

Introduction

Search/Browse Logs

Sec 6

Monitor & Feedback

Sec 7

Challenges and Future Trends
Search and Browse Logs
Different Types of Log Data: Search Logs

Search Logs
- Collected by search engine server
- Recording queries, clicks, as well as the search results by the search engine
Different Types of Log Data: Browse Logs

Browse Logs

- Collected by toolbar (browser plug-in) or ISP proxy
- Recording queries, clicks, and browsed URLs

White, R.W., et al. Studying the use of popular destinations to enhance web search interaction. SIGIR'07.
Different Types of Log Data: Other Logs

User Machine Logs
- Collected by user’s machine
- Stored on user’s machine
- Containing richer information, e.g., user’s every input in browser

Web Site Logs
- Web sites have their own server logs
- Recording how users visit the site
Putting Them Together

Query → Clicks

- Collected by Search Engine
- Collected by Toolbar or ISP Proxy

Browse

- User Machine Log
- Web Browser
- Web Site Log
- Browse Log

Search Log

Focus of this tutorial
Major Information in Search Logs

• Recorded by search engine at server side
• Four categories of information
  – User info: user ID & IP
  – Query info: query text, time stamp, location, search device, etc
  – Click info: URL, time stamp, etc
  – Results from search engine
    • Search results, Ads results, query suggestions, deep links, instant answers, etc.
Major Information in Browse Logs

• Recorded by toolbar at client side
• Major information
  – User ID & IP, query info, click info
  – Browse info: URL, time stamp
• Collection of browse logs follows strict privacy policy
  – Collecting data only when user permission is granted
  – Users can choose to opt-out at any time
Log Mining Applications
Log Mining Applications

• Log mining can [Silvestri09]
  – Enhance efficiency of search system
  – Enhance effectiveness of search system

• This tutorial focuses on effectiveness
Log Mining Application Examples

- **Query Understanding**
  - Query Transformation
  - Query Classification
  - Query Expansion
  - Query Suggestion

- **Document Understanding**
  - Metadata Extraction
  - Document Classification
  - Document Clustering
  - Importance Calculation

- **User Understanding**
  - User Segmentation
  - Personalized Search
  - Context-Aware Search

- **Query-Doc Matching**
  - Relevance Ranking
  - Search Results Clustering
  - Search Results Diversification

- **Monitor & Feedback**
  - Search Engine Metrics
  - User Satisfaction Evaluation

**Search System**
Frequently Used Data Summarizations
Summarizing Raw Log Data

• Raw log data is in plain text: unstructured data
• Can we convert raw log data structured data to facilitate different applications?
• Challenges: complex data structure and various applications
Data Structure of Search/Browse Logs

- Various types of data summarizations
- Complex relationship among data objects
  - Hierarchical relationship
  - Sequential relationship

How to describe the objects as well as their relationships?
Frequently-Used Data Types in Log Mining

90% of research papers on log mining use the four types of data summarization.

- Query: Srch clicks, Ads clicks
- Click-through bipartite
- Click patterns
- Session patterns
- Query histogram
Query Histogram

<table>
<thead>
<tr>
<th>Query</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>facebook</td>
<td>3,157 K</td>
</tr>
<tr>
<td>google</td>
<td>1,796 K</td>
</tr>
<tr>
<td>youtube</td>
<td>1,162 K</td>
</tr>
<tr>
<td>myspace</td>
<td>702 K</td>
</tr>
<tr>
<td>facebook.com</td>
<td>665 K</td>
</tr>
<tr>
<td>yahoo</td>
<td>658 K</td>
</tr>
<tr>
<td>yahoo.mail</td>
<td>486 K</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>486 K</td>
</tr>
<tr>
<td>ebay</td>
<td>486 K</td>
</tr>
<tr>
<td>facebook.login</td>
<td>445 K</td>
</tr>
</tbody>
</table>

Example applications:
• Query auto completion
• Query suggestion
• Query analysis: temporal changes of query frequency
Click-through Bipartite

- Example applications
  - Document (re-)ranking
  - Search results clustering
  - Query suggestion

![Click-through bipartite graph]

- Queries: \( q_1, q_2, q_3, q_4 \)
- URLs: \( u_1, u_2, u_3, u_4, u_5 \)

Weights:
- \( q_1 \to u_1: 30 \)
- \( q_1 \to u_2: 20 \)
- \( q_2 \to u_2: 100 \)
- \( q_2 \to u_3: 40 \)
- \( q_2 \to u_4: 1000 \)
- \( q_3 \to u_3: 120 \)
- \( q_4 \to u_4: 120 \)
- \( q_4 \to u_5: 10 \)
### Click Pattern

<table>
<thead>
<tr>
<th>Query</th>
<th>Doc 1</th>
<th>Doc 2</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>Doc N</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pattern 1 (count)</td>
</tr>
<tr>
<td>Doc 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Doc N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pattern 1 (count)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Doc 1</th>
<th>Doc 2</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Doc 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Doc 1</th>
<th>Doc 2</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Doc 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Doc 1</th>
<th>Doc 2</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>×</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Doc 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pattern 1 (count)</th>
<th>Pattern 2 (count)</th>
<th>Pattern n (count)</th>
</tr>
</thead>
<tbody>
<tr>
<td>× Doc 1</td>
<td>× Doc 2</td>
<td>× Doc 1</td>
</tr>
<tr>
<td>× Doc 2</td>
<td>× ...</td>
<td>× Doc 2</td>
</tr>
<tr>
<td>× ...</td>
<td>× ...</td>
<td>× ...</td>
</tr>
<tr>
<td>× ...</td>
<td>× ...</td>
<td>× ...</td>
</tr>
<tr>
<td>× ...</td>
<td>× ...</td>
<td>× ...</td>
</tr>
<tr>
<td>× ...</td>
<td>× ...</td>
<td>× ...</td>
</tr>
<tr>
<td>× ...</td>
<td>× ...</td>
<td>× ...</td>
</tr>
<tr>
<td>× ...</td>
<td>× ...</td>
<td>× ...</td>
</tr>
<tr>
<td>× ...</td>
<td>× ...</td>
<td>× Doc N</td>
</tr>
<tr>
<td>× Doc N</td>
<td></td>
<td>Doc N</td>
</tr>
</tbody>
</table>

### Example applications
- Estimate relevance of document to query
- Predict users’ satisfaction
- Query classification (informational vs navigational)
Session Pattern

Example applications
- Doc (re-)ranking
- Query suggestion
- User satisfaction prediction

User activities in a session
- Srch click: search click
- Ads click: advertisement click

Summary of Introduction

• Search & browse logs
  – Search logs: collected by search engine servers
  – Browse logs: collected by client-side toolbar

• Log mining applications
  – Query understanding, document understanding, user understanding, query-document matching, monitoring & feedback

• Frequently-used data summarization
  – Query histogram, click-through bipartite, click patterns, session patterns
Road Map

Search System

- Query Model
  - Query Understanding
- User/Context Profiles
  - User Understanding
- Document Model
  - Doc Understanding

Query-Document Matching
[Personalized/Context-aware]

Sec 1
Introduction

Search/Browse Logs

Sec 2
- Query Transformation
- Query Classification

Sec 4
Enhance

Sec 5
Enhance

Sec 6
Monitor & Feedback

Sec 7
Challenges and Future Trends
Query Understanding Using Log Data

• Query transformation: represent queries by similar queries
  – Refined queries (e.g., spelling error correction)
  – Related queries (e.g., more specific queries, more general queries)

• Query classification: represent queries by categories
  – User goals (informational, navigational, transactional)
  – Topics (e.g., ODP taxonomy)
  – Time sensitivity
  – Location sensitivity
  – Other categories...
Query Transformation
Query Transformation

• Similar queries = queries with same or similar search intent
• Applications: query expansion/ refinement/ suggestion
  – Query expansion
    • Re-write query to increase search recall
    • Example: ‘ny times’ → `ny times new york’
  – Query refinement
    • Spelling error correction: e.g., ‘machin learning’ → ‘machine learning’
    • Stemming
    • Acronym expansion
  – Query suggestion
    • Specialization: e.g., ‘harry potter’ -> ‘harry potter books’
    • Generalization: e.g., ‘seattle employment rate’ -> ‘employment rate’
    • Association: e.g., ‘walmart’ -> ‘sears’
• Using log data for query transformation
  – Using click-through data
  – Using session data
Methods Using Click-Through Data

• Using click-through bipartite

• Measure similarity between queries
  – Overlap of clicked document [Beeferman00], [Wen01], [Cao08]
  – Similarity of category or content of clicked documents [Wen01], [Yates04]

• Cluster queries
  – Agglomerative hierarchical method [Beeferman00], DBScan [Wen01], K-means [Yates04]
Challenges of Clustering a Click-Through Bipartite

• A click-through bipartite can be huge
  – For example, 151 million unique queries in experiment data

• Data set is of extremely high dimensionality
  – For example, 114 million unique URLs in experiment data

• The number of clusters is unknown

• Search logs increase dynamically

Cao, H., et al. Context-aware query suggestion by mining click-through and session data. KDD'08.
Observations from Real Data

• Average degrees of query and URL nodes are low
  – E.g., average degree of query nodes is 3.1; and average degree of URL nodes is only 3.7.

• Given a query \( q \), only a few “co-click” queries
  – Average number of co-click queries is upper-bounded by 
    \[ 3.1 \cdot (3.7 - 1) = 8.37 < 9 \]
  – No need to consider non co-click queries

Cao, H., et al. Context-aware query suggestion by mining click-through and session data. KDD'08.
Query Stream Clustering Algorithm

• A BIRCH-like algorithm
• Major difference: dimension array instead of cluster feature tree
  – Each element corresponds to one URL
  – \( d_i \rightarrow C_j \) if \( \exists q_k \in C_j \) such that \( q_k \) is connected to URL \( u_i \)
• Only one scan of the data set (details in paper)

Cao, H., et al. Context-aware query suggestion by mining click-through and session data. KDD'08.
Methods Using Session Data

• Extract sessions from log data
• Count co-occurrence or adjacency in sessions
  – If two queries are often adjacent or co-occurring in the same session, they are similar to each other [Jensen06][Huang03][Jones06]
• Measure correlation between queries in sessions
  – Mutual information, weighted mutual information [Jensen06]
  – Jaccard similarity, dependency, cosine similarity [Huang03]
  – Log likelihood ratio [Jones06]
Counting Co-Session Frequency

• Given two queries Q_a and Q_b, let C(Q_a, Q_b) be the number of sessions where Q_a and Q_b co-occur or are adjacent
• If C(Q_a, Q_b)>Threshold, then Q_a and Q_b are similar
• Problem: false positives
  – Q_a and Q_b may not be closely related
  – C(Q_a, Q_b) is high simply because both Q_a and Q_b are hot queries
    • Example: ‘britney spears’ and ’walmart’
Measuring Query Correlation

• Measure correlation
  – Jaccard measure, dependence measure, and Cosine measure [Huang03]
    • e.g., \(Jaccard(Q_a, Q_b) = \frac{C(Q_a, Q_b)}{C(Q_a) + C(Q_b) - C(Q_a, Q_b)}\)
  – Mutual information and variants [Jensen06][Jones06]
    • e.g., \(Mutual(Q_a, Q_b) = \frac{P(Q_a, Q_b)}{P(Q_a) \cdot P(Q_b)}\)

• Problem: hard to set threshold
Identifying Significantly Correlated Queries

• Testing the hypothesis that seeing $Q_b$ is independent of seeing $Q_a$
  $H_1: P(Q_b | Q_a) = p = P(Q_b | \neg Q_a)$
  $H_2: P(Q_b | Q_a) = p_1 \neq p_2 = P(Q_b | \neg Q_a)$
  Log Likelihood Ratio: $LLR = -2 \log \frac{L(H_1)}{L(H_2)}$

• Suppose the data follows binomial distribution, then $LLR$ follows $\chi^2$ distribution
  If $LLR > 3.84$, then 95% confidence to reject the $H_1$ hypothesis

Jones, R. et al. Generating query substitutions. WWW’06.
Query Classification
User Goal Classification Using Click-through and Anchor Text Data [Lee05]

- Only two categories considered, i.e., navigational and informational

- Basic idea
  - Navi query $\iff$ click distribution is skewed
  - Navi query $\iff$ anchor text distribution is skewed

- Method
  - Using mean, median, skewness, and kurtosis to characterize distributions of clicks and anchor texts
  - Linear combination of features

- Accuracy: 90%

- Challenge: difficult for tail queries
Result of Single Feature

- 50 head queries labeled by 28 graduate students
- Each point represents query
- $i(q)$ is percentage of informational labels of query $q$
- A feature is effective if we can set horizontal bar (i.e., threshold) to separate navigational queries from informational queries
Result of Linear Combination

- Linear combination
  \[ f = w_1 \cdot f_1 + w_2 \cdot f_2 + \cdots + w_n \cdot f_n \]
- A simple combination shows a better accuracy
  - Combines two features
  - Equal weights
  - Accuracy reaches 90%

\[ f = (\text{median of click distribution}) + (\text{median of anchor distribution}) \]
Query Topic Classification Using Click-through Data [Fuxman07]

• View click-through bipartite as directed graph
• Define random walk model
• Probability on edge represents transition probability (calculated using click-through counts)
• Probability of node represents probability of belonging to class
• Propagate class labels on graph
Random Walk Algorithm

• Add ‘null’ node to the click through bipartite
  – Each node may walk to null node with probability $\alpha$

• Iteration between two processes
  – Estimate probability of query node
    \[ P(l_q = c) = (1 - \alpha) \sum_{(q,u)} P(q \rightarrow u) P(l_u = c) \]
  – Estimate probability of URL node
    \[ P(l_u = c) = (1 - \alpha) \sum_{(q,u)} P(u \rightarrow q) P(l_q = c) \]

• It is guaranteed to converge

• Analogy to electrical network.
Query Topic Classification Using Click-through Data [Li08]

• Given a set of labeled queries (representing same topic)
• Train classifier based on content of queries
• Propagate class labels through click-though bipartite
• Iteratively combining content-based classification and click-based classification
• Accuracy: F score = 0.74 to 0.88
Propagation through Click-Through Bipartite

Labeled seeds

After propagation
Content-based Classifier

• Maximum Entropy Classifier

\[
P_\lambda(y \mid x) = \frac{\exp\left(\sum_i \lambda_i \phi_i(x, y)\right)}{\sum_y \exp\left(\sum_i \lambda_i \phi_i(x, y)\right)}
\]

\(x\) denotes query, \(y\) denotes query topic class, \(\phi(x, y)\) denotes feature, \(\lambda\) denotes parameter

• Using n-grams of query or snippets of query as features
Click-based Classifier

• Let $W$ be $m \times n$ matrix where $W[i, j]$ is click count on URL $j$ for query $i$

• Let $F$ be $m \times 2$ matrix where $W[i, j]$ is non-negative, real number indicate likelihood that query $i$ belongs to class $y$

• Random walk converges to

$$F^* = (1 - \alpha)(1 - \alpha A)^{-1} F^0$$

where $A = D^{1/2} W W^T D^{-1/2}$, $D$ is diagonal matrix in which element $d_{i,i}$ equals sum of elements in row $i$ of $W W^T$
Combining Classifiers

• Step 1: initialize $F^*$ by labeled seeds, initialize $\lambda$ as random

• Step 2: repeat
  – Train $\lambda^*$ of content-based classifier using classification results by current $F^*$
  – Train $F^*$ of click-based classifier use classification results by current $\lambda^*$
  until converge
Classification of Location Intent
[Welch et al 08]

• An empirical study
  – Significant fraction of queries are localizable
  – Roughly 30%, but users only explicitly localize them about half of the time
  – Users exhibit consensus on which queries are localizable

• Approach
  – Identify candidate localizable queries
  – Select relevant features
  – Train and evaluate classifier
Identify Candidate Localizable Queries

- Use U.S. Census Bureau data as an address book
- For each query $Q$, look up the address book
  - If a match is found, the matched part is $Q_l$, the remaining part is $Q_b$
  - $Q$ is a localized query of $Q_b$
- Aggregate all $Q_l$ for each $Q_b$
  - The set of $Q_l$ for each $Q_b$ is denoted by $L(Q_b)$
  - $Q_b$ is candidate localizable query if it often localized
Major Features of Classifier

• Localization ratio
  – How often query is localized

• Location distribution
  – Query should be localized with evenly distributed locations
Summary for Query Understanding

• Using log data to enhance query representation
• Query transformation
  – Using click-through data and session data
• Query classification
  – Using query histogram, click-patterns, and click-through bipartite
Road Map

Search System

Query-Document Matching [Personalized/Context-aware]

Query Model
- Query Understanding

User/Context Profiles
- User Understanding

Document Model
- Doc Understanding

Search/Browse Logs
- Enhance

Motivation
- Enhancing content modeling by query annotation
- Modeling page importance using browse behavior

Monitor & feedback

Challenges and Future Trends

Sec 1
- Introduction

Sec 2
Sec 3
Sec 4
Sec 5
Sec 6
Sec 7
Modeling Documents

• Often, a document is modeled as a bag of words

• Vector model
  – \( V = \{v_1, \ldots, v_n\} \), the set of terms
  – A document \( d = (w_1, \ldots, w_n) \), where \( w_i \) is the importance of term \( v_i \) in \( d \)
  – Importance can be measured by, for example, TFIDF
    • \( TF(v, d) = \# \text{ of times term } v \text{ appears in } d \)
    • \( IDF(v) = \log \left( \frac{N}{\# \text{ of documents in the corpus containing } v} \right) \)
    • \( TFIDF(v, d) = TF(v, d) \times IDF(v) \)

• A vector model tries to capture what the author of a document wants to express using the terms in the document
Web Documents and Links

• A Web page may be referred (pointed to) by some other Web pages
  – A link to the target page
  – Anchor text: a short annotation on the intension of reference

• A page having many incoming links tends to be important (well explored by link-based ranking methods, e.g., PageRank)

• What does anchor text tell us? – what others on the Web think about the target page
Anchor Text

• Anchor text may not be consistent with the vector model of the target page
  – Example: Anchor text “my publications” → DBLP bibliography server

• Anchor text can be used to complement the vector model of a target Web page
  – What the author writes + what some others read

• Anchor text tends to be bias and static
  – Old pages may receive more references, and thus more anchor text annotations
  – Once a link is made, more often than not, it will not be updated (at least in a long time)
Search Logs as Annotations from Users

• User queries $\rightarrow$ search results $\rightarrow$ user clickthroughs
  – If a user asks a query $Q$ and clicks a page $P$, likely $P$ is related to $Q$ – $Q$ can be used as an annotation of why a user wants to read $P$

• User clickthroughs can be used as dynamic, continuously updated, more accurate (after aggregation) annotation of Web pages

• User browsing trajectories can be counted as votes for popular pages
## Log Data & Document Understanding

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Text</th>
<th>Hyperlinks</th>
<th>Log data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling content</td>
<td>Bag of words</td>
<td>Anchor text</td>
<td>Query annotations</td>
</tr>
<tr>
<td>Importance of documents</td>
<td>Authorities and hubs indicated by the link structure of the Web</td>
<td>Users voting for Web page importance by browsing and clicking Web pages</td>
<td></td>
</tr>
</tbody>
</table>
Road Map

- **Query Model**
- **User/Context Profiles**
- **Document Model**

**Search System**

- **Query-Document Matching**
  - [Personalized/Context-aware]

**Sections**

- **Sec 1**: Introduction
- **Sec 2**: Enhance
- **Sec 3**: Monitor & feedback
- **Sec 4**: Challenges and Future Trends

**Motivation**

- Enhancing content modeling by query annotation
- Modeling page importance using browse behavior

**Documents**
Using Queries as Features

• Queries can be used as features to model documents
• Two documents are similar if they are clicked in the same set of queries

Using queries as “bridges”, similar documents \( d_2 \) and \( d_3 \) can be captured

Challenges

• How to model “query annotations”?  
• Search log data is sparse, how to handle documents that have very few or even no clicks?
  – A small number of queries are frequently asked, many queries are rarely asked
  – A small number of Web pages are heavily clicked, many Web pages have very few or even no clicks
• How to use “query annotations”?  

Two-Way Annotation

- Can we use documents as features of queries?

Query-Document Model

- Let $V = \{t_1, \ldots, t_m\}$ be the vocabulary of all queries in the access log $L$, where $t_1, \ldots, t_m$ are the terms in $V$
- Let $Q(d)$ be the set of all queries in $L$ from which users clicked at least one time on $d$
- Let the frequency of $t$ in $Q(d)$ be the total number of times that queries that contained $t$ were used to visit $d$

\[
\vec{d} = \langle C_1, \ldots, C_m \rangle
\]

- Where $C_i = \text{TFIDF}(t_i, Q(d))$

- [Barbara Poblete, Ricardo Baeza-Yates, Query-sets: using implicit feedback and query patterns to organize web documents. WWW'08.]
Example

Barbara Poblete, Ricardo Baeza-Yates, Query-sets: using implicit feedback and query patterns to organize web documents. WWW'08.
Query-Set Document Model

• Query-document model considers terms in queries independently even if some of them co-occur frequently
  – “Apple” and “Apple phone” carry very different meanings
• Query-set document model includes frequent term combinations as features for documents

• [Barbara Poblete, Ricardo Baeza-Yates, Query-sets: using implicit feedback and query patterns to organize web documents. WWW'08.]
Example

Barbara Poblete, Ricardo Baeza-Yates, Query-sets: using implicit feedback and query patterns to organize web documents. WWW'08.
### Case Study

<table>
<thead>
<tr>
<th>DocId</th>
<th>Vector Space</th>
<th>Query</th>
<th>Query-Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
<td>download, test, file, 2007, guide, publication</td>
<td>official, test, social, publication, module, science, guides</td>
<td>physics, geometry, physics topics, topics, admission topics</td>
</tr>
<tr>
<td>74</td>
<td>able, Europe, world, kingdom, MBA, Asia, library</td>
<td>degree, search, graduate, certificate, advanced, diploma, simulation</td>
<td>university scholarship, universities, university ranking, best universities</td>
</tr>
<tr>
<td>47</td>
<td>scholarship, application, loan, benefit, fill, form</td>
<td>dates, free, vocational, on-line, scholarship, loan</td>
<td>loan scholarship loan cosigner loan application</td>
</tr>
<tr>
<td>80</td>
<td>vitae, curriculum, presentation, job, letter, interview, experience, highlight</td>
<td>CV, letter, resume, recommendation, presentation, example</td>
<td>CV, write CV, curriculum vitae, CV example, write curriculum vitae</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Quality</th>
<th>Dimensions</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector-Space</td>
<td>40%</td>
<td>8,910</td>
<td>69%</td>
</tr>
<tr>
<td>Query</td>
<td>57%</td>
<td>7,718</td>
<td>67%</td>
</tr>
<tr>
<td>Query-Set</td>
<td>77%</td>
<td>564</td>
<td>81%</td>
</tr>
</tbody>
</table>

Barbara Poblete, Ricardo Baeza-Yates, Query-sets: using implicit feedback and query patterns to organize web documents. WWW'08.
Tackling Query Sparsity

• Many queries are rarely asked
• Idea: clustering similar queries to identify groups of user information needs of significant sizes → reliable annotations on Web pages clicked
• A two phase algorithm
  – Preprocessing phase
  – Online searching phase

Preprocessing Phase

• At periodical and regular intervals
• Extract queries and clicked URLs from the Web log, and cluster them using the text of all the clicked URLs (by k-means)
• For each cluster $C_i$, compute and store
  – A list $Q_i$ containing queries in the cluster
  – A list $U_i$ containing the k-most popular URLs along with their popularity

Online Searching Phase

• Input: a query q
• If q appears in the stored clusters, find the corresponding cluster $C_i$ containing q, use $U_i$ to boost the search engine ranking algorithm by

$$New\text{Rank}(u) = \beta \times Orig\text{Rank}(u) + (1 - \beta) \times \text{Rank}(u)$$

Examples & Effectiveness

<table>
<thead>
<tr>
<th>Query</th>
<th>Other Queries in Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>dress bride</td>
<td>house of bride</td>
</tr>
<tr>
<td></td>
<td>dress wedding</td>
</tr>
<tr>
<td></td>
<td>dress bridegroom</td>
</tr>
<tr>
<td></td>
<td>wedding cake</td>
</tr>
<tr>
<td></td>
<td>wedding rings</td>
</tr>
<tr>
<td>free internet</td>
<td>phone company</td>
</tr>
<tr>
<td></td>
<td>free internet connection</td>
</tr>
<tr>
<td></td>
<td>free ads</td>
</tr>
<tr>
<td></td>
<td>cibercafe santiago</td>
</tr>
<tr>
<td></td>
<td>free text messages</td>
</tr>
<tr>
<td></td>
<td>free email</td>
</tr>
<tr>
<td>yoga</td>
<td>tai chi exercises</td>
</tr>
<tr>
<td></td>
<td>astral letter</td>
</tr>
<tr>
<td></td>
<td>reiki</td>
</tr>
<tr>
<td></td>
<td>birth register</td>
</tr>
<tr>
<td>soccer leagues</td>
<td>ivan zamorano</td>
</tr>
<tr>
<td></td>
<td>soccer leagues chile</td>
</tr>
<tr>
<td></td>
<td>soccer teams chile</td>
</tr>
<tr>
<td></td>
<td>marcelo salas</td>
</tr>
</tbody>
</table>

Documents Not Clicked

• Many documents may have very few or even no clicks
  – 75% of a sample of 2.62 million Web pages do not have any click in a real case study
• Idea: use smoothing techniques
  – Random walk
  – Discounting

• [Gao, J., et al. Smoothing clickthrough data for web search ranking. SIGIR'09.]
Random Walk

Construct matrix $A_{ij} = P(d_i | q_j)$ and matrix $B_{ij} = P(q_i | d_j)$

Random walk using the probabilities

Before expansion, document $d_3$ has a clickthrough stream of $q_2$ only; after a random walk expansion, the click-through stream is augmented with query $q_1$, which has a similar click pattern as $q_2$

Good-Turing Estimator

• Let $N$ be the size of a sample text, $n_r$ be the number of words which occur in the text exactly $r$ times: $N = \sum_r rn_r$.

• Estimate $P_{GT}$ for a probability of a word that occurred in the sample $r$ times as $P_{GT} = \frac{r^*}{N}$, where $r^* = (r+1) \frac{n_{r+1}}{n_r}$.

• Heuristic: not discounting high values of counts, i.e., for $r > k$ (typically $k = 5$), $r^* = r$. 
Discounting

• Applying Good-Turing estimate on raw clickthrough data does not work – all not-clicked words take the same free ride
  – Those features are meaningless

• Idea: discounting the clickthrough feature values

• The discounting method works very well in the empirical studies
Browse Logs and Search Trails

• Browse logs may contain more information than search log
  – Search trails record other browsing activities in addition to queries
  – Queries and documents are linked through terms in queries

Mikhail Bilenko, Ryen W. White. Mining the search trails of surfing crowds: identifying relevant websites from user activity WWW'08.
Road Map

- **Query Model**
  - Query Understanding
- **User/Context Profiles**
  - User Understanding
- **Document Model**
  - Doc Understanding

**Search System**

- **Query-Document Matching**
  - [Personalized/Context-aware]

**Motivation**
- Enhancing content modeling by query annotation
- Modeling page importance using browse behavior

**Search/Browse Logs**

**Documents**

- Sec 1: Introduction
- Sec 2: Enhancement
- Sec 3: Challenges and Future Trends
Modeling Importance of Web Pages

- An important task in document modeling
- PageRank
  - A link from one page to another is regarded as an endorsement of the linking page
  - The more links pointed to a page, the more likely the page is important
  - The importance of pages can be propagated in the graph
- HITS
  - A hub page links to many pages
  - An authority page is pointed by many pages
  - Good hubs tend to link to good authorities, and vice versa
- No user feedback is considered
Using User Browsing Behavior

• User browsing graph
  – Vertices representing pages
  – Directed edges representing transitions between pages in browsing history
  – Lengths of staying time are included

• Using the continuous-time Markov process
  – The stationary probability distribution of the process is the importance of a page

• [Yuting Liu, Bin Gao, Tie-Yan Liu, Ying Zhang, Zhiming Ma, Shuyuan He, Hang Li. BrowseRank: letting web users vote for page importance. SIGIR'08.]
Example: Spam Fighting

• BrowseRank can push many spam websites to the tail buckets
• The number of spam Websites in the top buckets in BrowseRank is smaller than that in PageRank and in TrustRank
• Users browse meaningful pages more significantly than spam pages
• BrowseRank consistently outperforms PageRank in Web search
ClickRank

• A session is modeled as a logical sequence of hops through the Web graph according to the user’s retrieval intension
  – Temporal attributes (e.g., dwell time) reflects user’s interest on a page
• For a session $s$, the local ClickRank defines a random variable associated with all pages on the Web graph reflecting how important a page is to the user’s retrieval intension in this session

• [Zhu, G, Mishne, G. Mining rich session context to improve web search. KDD'09.]
Summary

• Search logs and browse logs can be used to improve document modeling
  – Verified by user studies
• Enriched models of documents considering log data
  – Central idea: using query terms and segments as features
• Tackling sparsity of log data
  – Clustering similar queries
  – Smoothing
Challenges

• From document modeling to document cluster/site modeling
  – Several Web pages are often visited together

• Modeling temporal characteristics of search activities
  – Detecting bursts of new interests

• Many applications can be improved by using search/browse log data
Query Document Matching

Search System

Query-Document Matching

Output

Query Model
Query Understanding

Document Model
Doc Understanding

Queries

A ranked list of documents in the descending order of degrees matching the query

Documents
Why Do Users Click?

• A user asks a query, a search engine shows a list of results

• Why does a user click on a result?
  – The result looks interesting, probably hinted by the snippet information

• Why does a user click on another result?
  – Possibly, the previous result clicked does not (completely) satisfy the user’s information need
Clickthrough as User Feedback

• User clickthrough data provides implicit feedback and hints about user preference on search results
• Using user clickthrough data, how can we improve ranking?
• A naïve method: promoting clicked answers, demoting not-clicked answers
  – Position bias: answers at higher positions may get more clicks
  – What about documents never been shown?
Learning Preferences from Clicks

• Pair-wise versus list-wise preferences
  – Pair-wise: between pages a and b, which one is more preferable?
  – List-wise: given a set of Web pages, sort them in preference order

• Clickthrough information used in learning
  – What does a click tell us?
  – What do a series of clicks tell us?
  – What do a series queries and the corresponding clickthrough information tell us?

• Preference functions: binary, scoring function, categorical/discrete

• Applications: organic search and sponsored search
Major Methods

- Learning pair-wise preferences
- Sequential click models
A Naïve Method

• A clicked answer is more preferable than a non-clicked answer ranked at a lower place

• For a ranking of results \((d_1, \ldots, d_n)\) and a set \(C\) of clicked results, extract a preference relation

\[ d_i < d_j \]

for \(1 \leq j < i, i \in C, \text{ and } j \not\in C\)

• Drawbacks: much information has not been used
  – No comparison between clicked answers
  – No comparison between non-clicked answers
What Do User Clicks Mean?

• For a ranking of results \((d_1, \ldots, d_n)\) and a set \(C\) of the clicked results

• (Click > Skip above) for all pairs \(1 \leq j < i, i \in C,\) and \(j \notin C, R(d_i, d_j)\)
  - (Last click > Skip above) let \(i \in C\) be the rank of the link that was clicked temporally last, for all pairs \(1 \leq j < i, j \notin C, R(d_i, d_j)\) [more accurate empirically]

• (Last click > No-click next) for all pairs \(i \in C\) and \(i + 1 \notin C, R(d_i, d_{i+1})\)

Kendall’s $\tau$

- How can we compare two rankings of a set of $m$ documents?
- For two preference relations $R$ and $R'$, let $P$ be the number of concordant pairs $(a, b)$ such that $R(a, b) = R'(a, b)$, and $Q$ be the number of discordant pairs $(a, b)$ such that $R(a, b) \neq R'(a, b)$

$$\tau(R, R') = \frac{P - Q}{P + Q} = 1 - \frac{2Q}{\binom{m}{2}}$$

- $P + Q = m$
How Good Is a Preference Relation?

• For a preference relation $R$, the average precision of $R$ is bounded by

$$\text{Avg Prec}(R) \geq \frac{1}{l} \left[ Q + \left( \frac{l+1}{2} \right) \right]^{-1} \left( \sum_{i=1}^{l} \sqrt{i} \right)^2$$

where $l$ is the number of relevant documents.

• Learn a preference relation $R$ maximizing

$$\int \tau(R_q, R^*)d \Pr(q, R^*)$$

• However, the ideal preference is unknown ...
  – An SVM algorithm

• T. Joachims, Optimizing search engines using clickthrough data. KDD '02.
Query Chains

• Users often reformulate their queries to approach a good representation of their information needs (for the target search engine)
  – “Lexis Nexis” → “Lexis Nexus”
• Query chain: a sequence of reformulated queries asked by a user
  – How can we use query chains to learn preferences?

• Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback. KDD'05.
Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback. KDD'05.
Example

Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback. KDD'05.
Using Aggregated Clickthrough Data

• The preferences learned from individual user clickthrough data may not be highly reliable

• Using intelligence of crowd – aggregating clickthrough data from many users
  – Let click(q, d) be the corresponding aggregate click frequency of document d with respect to query q
  – Let cdif(q, d_i, d_j) = click(q, d_i) – click(q, d_j)

• If cdif(q, d_i, d_j) > 0, d_i >_q d_j

Presentation Bias

• A user is more likely to click on documents presented higher in the result set irrespective of relevance

• A simple FairPairs algorithm
  – Let $R = (d_1, \ldots, d_n)$ be the results for some query
  – Randomly choose $k \in \{0, 1\}$ with uniform probability
  – If $k = 0$ ($k = 1$), for all odd (even) numbers $i$, swap $d_i$ and $d_{i+1}$ with probability 0.5
  – Present $R$ to the user, recording clicks on results
  – Every time the lower result in a pair that was considered for flipping is clicked, record this as a preference for that result over the one above it

• Filip Radlinski and Thorsten Joachims. Minimally invasive randomization for collecting unbiased preferences from click-through data. AAAI'08.
Why Does FairPairs Work?

• Let $c_{ij}$ be the number of times a user clicks on $d_i$ when $d_j$ is presented just above $d_i$

• FairPairs designs the experiment such that $c_{ij}$ is the number of votes for $(d_i > d_j)$ and $c_{ji}$ is the number of votes for $(d_j > d_i)$
  – The votes are counted only if the results are presented in equivalent ways

• Both sets of votes are affected by presentation bias in the same way

• Filip Radlinkski and Thorsten Joachims. Minimally invasive randomization for collecting unbiased preferences from click-through data. AAAI'08.
Passive Learning

- A user often considers only the top-ranked answers, and rarely evaluates results beyond the first page
  - The clickthrough data collected passively is strongly biased toward documents already ranked highly
- Highly relevant results not initially ranked highly may never be observed and evaluated
  
  - F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. KDD'07.
Active Exploration for Learning

• Idea: presenting to users a ranking optimized to obtain useful feedback

• A naïve method: intentionally present unevaluated results in the top few positions
  – May hurt user satisfaction

• A principled approach: using a Bayesian approach

• F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. KDD'07.
Major Methods

• Learning pair-wise preferences
• Sequential click models
Why Sequential Click Model?

• Pair-wise preferences are easy to learn, but may not generate a ranked list
  – Given $a > b$, $b > c$, and $c > a$, no ranking can be generated

• Sequential click model: for a given query, produce a ranking of documents
  – Using a sequential click model, a search engine can retrieve relevant documents that have not yet been clicked for that query, and rank those documents effectively
Click Bias on Presentation Order

• The probability of click is influenced by the position of a document in the results page

• Click bias modeling: how probability of click depends on positions
  – Probability $P(c \mid r, u, q)$ that a document $u$ presented at position $r$ is clicked by a user who issued a query $q$
Baseline/Examination Hypotheses

• Baseline: no bias associated to the document positions
  – \( P(c|r, u, q) = P(a|u, q) \), where \( P(a|u, q) \) is the attractiveness of document \( u \) as a result of query \( q \)

• The examination/separatability hypothesis: users are less likely to look at results at lower ranks – each rank has a certain probability \( P(e|r) \) of being examined
  – \( P(c|r, u, q) = P(e|r)P(a|u, q) \)
  – When \( P(e|r) = 1 \), we obtain the baseline

• Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparision of click position-bias models. WSDM'08.
The Cascade Model

• Users view search results from top to bottom, deciding whether to click each result before moving to the next
  – Each document is either clicked with a probability \( P(a \mid u, q) \) or skipped with a probability \( 1 - P(a \mid u, q) \)
  – A user clicks never comes back; a user skips always continues
    \[
    P(c \mid r, u, q) = P(a \mid u, q) \prod_{i=1}^{r-1} (1 - P(a \mid u_i, q))
    \]

• Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. WSDM'08.
Empirical Study

• The cascade model performs significantly better than the other models for clicks at higher ranks, but slightly worse than the other models for clicks at lower ranks.

• What does the cascade model capture?
  – Users examine all documents sequentially until they find a relevant document and then abandon the search.

• Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. WSDM'08.
What Are Not Modeled Yet?

• What is the possibility that a user skips a document without examining it

• In informational queries, a user may examine documents after the first click – what is the possibility?
  – In navigational queries, a user tends to stop after the first relevant document is obtained

• We need a user browsing model

• Georges E. Dupret, Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. SIGIR'08.
The Single Browsing Model

• The probability that a user examines a document depends on the distance from the document to the last click
  – Rationale: a user tends to abandon the search after seeing a long sequence of unattractive snippets

• Assuming both attractiveness and examination be Bernoulli variables

• Georges E. Dupret, Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. SIGIR'08.
The Single Browsing Model

• Assuming both attractiveness and examination be Bernoulli variables
  - \( P(a|u, q) = \alpha_{uq}^a (1 - \alpha_{uq})^{1-a} \)
  - \( P(e|r, d) = \gamma_{rd}^e(1 - \gamma_{rd})^{1-e} \)
  - \( \alpha_{uq} \) is the probability of attractiveness of snippet \( u \) if presented to a user who issued query \( q \)
  - \( \gamma_{rd} \) is the probability of examination at distance \( d \) and position \( r \)

• The full model: \( P(c, a, e|u, q, d, r) = P(c|a, e)P(e|d, r) P(a|u, q) = P(c|a, e) \gamma_{rd}^e(1 - \gamma_{rd})^{1-e} \alpha_{uq}^a (1 - \alpha_{uq})^{1-a} \)

• Georges E. Dupret, Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. SIGIR'08.
Multiple Browsing Model

• Navigational versus informational queries
  – In general, there may be a variety of many kinds of user behaviors

• Build a mixture of single browsing models, and use a latent variable \( m \) to indicate which is used for a particular query \( q \)
  – \( P(e | r, d, m) = \gamma_{rdm}^e (1 - \gamma_{rdm})^{1-e} \)

• Georges E. Dupret, Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. SIGIR'08.
Logistic Model

• Model the logarithm of the odds of a click
  – Odds = P(c=1|r, d, u, q)/(1 – P(c=1|r, d, u, q))

• The logarithms of the odds are regressed against the explanatory variable
  – ln odds = β_{uq} + β_{rd}
  – Odds = \exp(β_{uq}) + \exp(β_{rd})

• Georges E. Dupret, Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. SIGIR'08.
A Dynamic Bayesian Network Model

For a given position $i$, $C_i$ is the only observed variable indicating whether there was a click or not at this position. $E_i$, $A_i$, and $S_i$ are hidden binary variables modeling whether the user examined the URL, the user was attracted by the URL, and the user was satisfied by the landing page, respectively.

Chapelle, O. and Zhang, Y. A Dynamic Bayesian Network Click Model for Web Search Ranking. WWW'09.
Further Development

• Handling huge amounts of data
  – Chao Liu, Fan Guo, Christos Faloutsos. BBM: bayesian browsing model from petabyte-scale data. KDD'09.

• Click chain model
Summary

• User clickthrough data provides implicit feedback and hints about user preference on search results

• Clickthrough information used in learning
  – A click $\rightarrow$ a series of clicks $\rightarrow$ a series queries and the corresponding clickthrough

• Preference functions: binary, scoring function, categorical/discrete

• Applications: organic search and sponsored search
Challenges

• There are still many problems remained open
• How to learn preferences effectively about rare queries and documents?
• Context-aware preference learning
  – Query “digital camera”
  – About Cannon versus Nikon, different users may have different preferences – how can we detect the preferences?
• Temporal and burst sensitive preferences
  – Query “Obama”
  – More recent events may be more preferable
  – Some milestone events (e.g., healthcare plan) may be more preferable
  – How to model, learn, and apply such preferences?
Road Map

Search System

Query Representation
Query Understanding

User/Context Profiles
User Understanding

Doc Representation
Doc Understanding

Query-Document Matching [Personalized/Context-aware]

Sec 4
Enhance

Sec 2
Enhance

Sec 6
Monitor & feedback

Sec 5

- Personalization: creates a user profile for each individual user
- Contextualization: creates a context profile for each search activity

Search/Browse Logs

Challenges and Future Trends
Motivation for User Understanding

- Different users may have different intents behind the same query
  - Example “GMC”
  - Cars? Medical Council?
Personalization and Contextualization

• Personalization: creates a user profile for each user to describe user preference
  – If the user profile shows that the user often raises medical-related queries, “GMC” is more likely to refer to General Medical Council

• Contextualization: creates a context profile for each search activity to characterize the search environment
  – If the user inputs query “Honda” and “Nissan” before “GMC” in the same session, it is more likely the user is searching for GMC cars
Methods for Personalization

Personalization: creates user profiles
- Click-based profiles, e.g., [Dou07]
- Term-based profiles, e.g., [Teevan05][Tan06]
- Topic-based profiles, e.g., [Pretschner99][Speretta05][Qiu06]
Click-Based Personalization

- Users tend to click on the results they clicked before
- Clicked-based user profiles
  - For each user $U_k$, $C_{kij}$ is the number of times user clicked on document $D_j$ as an answer to query $Q_i$
<table>
<thead>
<tr>
<th>Query</th>
<th>Document</th>
<th>Clicks</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_1$</td>
<td>$D_1$</td>
<td>$C_{k11}$</td>
</tr>
<tr>
<td>$Q_1$</td>
<td>$D_2$</td>
<td>$C_{k12}$</td>
</tr>
<tr>
<td>$Q_2$</td>
<td>$D_1$</td>
<td>$C_{k21}$</td>
</tr>
<tr>
<td>$Q_2$</td>
<td>$D_3$</td>
<td>$C_{k23}$</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Applying click-based user profiles
  - Personalized score
    $$ S(D_j|Q_i, U_k) = \frac{C_{kij}}{\sum_j C_{kij} + \beta} $$
  - A user’s history is often sparse

Term-Based Personalization

• **BM25 model**

  
  \[
  \text{Score}(q, d) = \sum_{t_i \in q} \frac{tf_i(k_1 + 1)}{k_1 + tf_i} w_i
  \]

  \(tf_i\): frequency of term \(t_i\) in \(d\)

  \(k_1\) is a constant

  \(w_i = \log \frac{(r_i + 0.5)(N - n_i - R + r_i + 0.5)}{(n_i - r_i + 0.5)(R - r_i + 0.5)}\)

  \(N\): number of documents in the corpus

  \(n_i\): number of documents containing \(t_i\)

  \(R\): number of relevant documents

  \(r_i\): number of relevant documents containing \(t_i\)

  \(r_i\): number of browsed documents containing \(t_i\)

Using the documents browsed by the user as “relevance feedbacks”

Teevan et al. Personalizing search via automated analysis of interests and activities. SIGIR’05.
Topic-Based Personalization

• Create a topic vector $\pi^u$ for each user $u$

<table>
<thead>
<tr>
<th>0.1</th>
<th>0</th>
<th>0.5</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>...</th>
<th>0</th>
<th>0.2</th>
</tr>
</thead>
</table>

  – Each element $\pi_u[c_i]$ represents the probability that $u$ is interested in a topic $c_i$
  – The probabilities are estimated from all search/browse history of $u$

• Applying the topic-based profile

$Score(q, d_j, \pi^u) = s(q, d_j) \cdot (0.5 + \frac{1}{4} \sum_{i=1}^{4} \pi^u[c_i] \cdot \gamma(d_j, c_i))$

  – $s(q, d_j)$ is the original score, $\gamma(d_j, c_i)$ is the probability that document $d_j$ belongs to topic $c_i$
  – The more the topic distribution of $d_j$ matches the user’s topic profile, the higher personalized score $d_j$ receives

Road Map

- **Sec 1**: Introduction
- **Sec 2**: Enhance
- **Sec 3**: Enhance
- **Sec 4**: Enhance
- **Sec 5**: Monitor & feedback
- **Sec 6**: Challenges and Future Trends

**Search System**

- **Query Representation**
- **Query Understanding**
- **User/Context Profiles**
- **Doc Representation**
- **Doc Understanding**
- **Sec 4**

**Query-Document Matching**

- **Personalization**: creates a user profile for each individual user
- **Contextualization**: creates a context profile for each search activity

- **Search/Browse Logs**

- **Challenges and Future Trends**
Methods for Contextualization

- Option 1: Directly creating a context-profile from the context information
- Similar to personalization
  - The major difference is that contextualization usually considers a users’ short history within a session, while personalization usually consider a user’s long history over several days or even longer time
  - Click-based profiles, e.g., [Dou07]; Term-based profiles, e.g., [Shen05]; Topic-based profiles, e.g., [White09]
- Problem: context is very sparse
Option 2: Summarizing context profiles from log data

- To address context sparsity
- A context profile may be summarized from many users’ histories
- A Hidden Markov Model approach [Cao09]
A HMM for User Sessions

- Behind each search, a user bears a search intent in mind (hidden state)
- The user formulates queries and clicks search results (observations)
- Given a query $q_t$ and its context info $O_{1...t-1}$, we can infer user’s search intent at time $t$ and rank the search results
Learning Large Scale HMM from Logs

• How to define hidden states?
  – Using click-through bipartite
  – A hidden state is represented by a cluster of queries and URLs

• How to learn a very large HMM from log data?
  – Decompose the EM algorithm into Map-Reduce structure and deploy it in a distributed system
Training Large Scale HMM using Map-Reduce

The training on 1.8 billion queries, 2.6 billion clicks, 840 million sessions finished in several hours.
Summarization for User Understanding

• Personalization and contextualization
  – Both to address ambiguity
  – Personalization creates a user profile for each user, while contextualization creates a context profile for each search activity

• Methods for personalization and contextualization
  – Clicked-based methods, term-based methods, topic-based methods
  – Summarizing context profiles from log data using a Hidden Markov Model
Road Map

- **Sec 1** Introduction
- **Sec 2** Enhance
- **Sec 3** Enhance
- **Sec 4** Enhance
- **Sec 5** Enhance
- **Sec 6** Monitor & feedback
- **Sec 7** Challenges and Future Trends

**Search System**
- Query Representation
- Query Understanding

**Query-Document Matching** [Personalized/Context-aware]

**Documents**

**User/Context Profiles**
- User Understanding

**Search/Browse Logs**

**Monitor & feedback**
Monitoring Search Engine Using Logs

• Calculating a series of metrics from log data
  – Query traffic
  – Number of users
  – Average click-through rate
  – Etc.

• An excellent tutorial on search engine metrics by Ali Dasdan, Kostas Tsioutsiouliklis, Emre Velipasaoglu in WWW’10
Evaluating User Satisfaction Using Logs

• Evaluation at session level
  – Sequential patterns
  – Markov chains
  – Layered Bayesian model
An Example of Sequential Behavioral Patterns

- Define a vocabulary of five letters to describe sessions
- Explore correlations between sequential patterns and user satisfaction

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Freq.</th>
<th>%SAT</th>
<th>%PSAT</th>
<th>%DSAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SqLrZ</td>
<td>509</td>
<td>81</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>SqLrLZ</td>
<td>117</td>
<td>75</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>SqLrLrZ</td>
<td>82</td>
<td>73</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>SqLrqLr*</td>
<td>70</td>
<td>64</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>SqLrLrLrZ</td>
<td>61</td>
<td>57</td>
<td>22</td>
<td>19</td>
</tr>
<tr>
<td>SqLrLr*</td>
<td>362</td>
<td>23</td>
<td>39</td>
<td>36</td>
</tr>
<tr>
<td>SqLrLrLr*</td>
<td>129</td>
<td>20</td>
<td>37</td>
<td>42</td>
</tr>
<tr>
<td>SqLrLrLrLr*</td>
<td>114</td>
<td>13</td>
<td>35</td>
<td>51</td>
</tr>
</tbody>
</table>

- Session starts (S)
- Submits a query (q)
- Result list returned (L)
- Clicks a result (r)
- Exits session (Z)

Limitations of Sequential Patterns

• The previous sequential pattern model has several limitations
  – Only “query” and “clicking a result” modeled
    • May not capture user behavior well
  – No aggregation on patterns
    • May harm the generalization power on new sessions
  – Hard to incorporate features which describe user activities into the sequential patterns
    • E.g., dwell time, whether first click in the session, etc
A Markov Chain Model

• Model rich types of user clicks with a Markov chain [Hassan10]
  – START: the user starts a new goal
  – A query (Q)
  – A click of any of the following types:
    • Algorithmic Search Click (SR)
    • Sponsored Search Click (AD)
    • Related Search Click (RL)
    • Spelling Suggestion Click (SP)
    • Shortcut Click (SC)
    • Any Other Click (OTH), such as a click on one of the tabs
  – END: the user ends the search goal

Hassan, A. et al. Beyond DCG: user behavior as a predictor of a successful search, WSDM'10.
User Activity Markov Model

- The Markov model is defined as $G = (V, E, w)$
  - $V = \{Q, SR, AD, RL, SP, SC, OTH\}$ is the set of possible user actions during the session
  - $E \subseteq V \times V$ is the set of possible transitions between any two actions
  - $w : E \rightarrow [0..1]$ is the transition probability from state $s_i$ to state $s_j$

$$w(s_i, s_j) = \frac{N_{s_i,s_j}}{N_{s_i}}$$

Hassan, A. et al. Beyond DCG: user behavior as a predictor of a successful search, WSDM'10.
Predicting User Satisfaction

• Offline: label training sessions and train two Markov models
  – $M_s$ model: trained by success sessions
  – $M_f$ model: trained by failure sessions

• Online: given a session $S=(s_1,...,s_n)$
  – Calculate the likelihood from both $M_s$ and $M_f$
    \[
    LL_M(S) = \sum_{i=2}^{n} W(S_{i-1}, S_i)
    \]
    
    
    \[
    \text{Pred}(S) = \begin{cases} 
      1 & \text{if } \frac{LL_{M_s}(S)}{LL_{M_f}(S)} > \tau \\
      0 & \text{otherwise.}
    \end{cases}
    \]

Hassan, A. et al. Beyond DCG: user behavior as a predictor of a successful search, WSDM'10.
Adding Time Features to Model

• Time distributions of state transitions are different in successful and unsuccessful sessions.
• Assume the transition time follows a gamma distribution.
• Time distributions incorporated into the transition probabilities of the Markov models.

Hassan, A. et al. Beyond DCG: user behavior as a predictor of a successful search, WSDM’10.
Advantages of the Markov Chain Model

• A richer vocabulary
  – Capture more user behaviors

• Summarize user behaviors through transition probabilities between states
  – Predict user satisfaction for new sessions not in the training data

• A probabilistic model
  – Flexible to incorporate other features beyond the sequence of activities
Piwowarski, B., et al. Mining user web search activity with layered Bayesian networks or how to capture a click in its context. WSDM'09.
A Bayesian Network Model

• Four hidden variables
  – Chain, search, page, click
  – Each hidden variable has a predefined number of states

• Each hidden variable is associated with some observed features
  – Chain: # searches
  – Search: # pages requested
  – Page: # clicks
  – Click: a) dwell time; b) whether “re-click”; c) relevance judgement if available

Piwowarski, B., et al. Mining user web search activity with layered Bayesian networks or how to capture a click in its context. WSDM'09.
Example of BN

Piwowarski, B., et al. Mining user web search activity with layered Bayesian networks or how to capture a click in its context. WSDM'09.
Using BN to Predict User Satisfaction

• In the offline stage
  – Learning model parameters

• In the online stage
  – Given the observed values of a user session, infer the distributions of states of the hidden variables
  – Extract BN features
    • Distribution of the states for each hidden variable
    • Maximal likelihood of the BN
  – Apply BN features to predict satisfaction

Piwowarski, B., et al. Mining user web search activity with layered Bayesian networks or how to capture a click in its context. WSDM'09.
Summary of Monitoring and Feedback

• An important function of log data is to monitor the status of the search engine and collect user feedbacks

• Evaluating user satisfaction at session level
  – Sequential patterns
  – Markov chains
  – Bayesian network
Road Map

Search System

Query-Document Matching [Personalized/Context-aware]

Search/Browse Logs

Sec 1 Introduction

Sec 2 Enhance

Sec 3 Enhance

Sec 4 Enhance

Sec 5 Enhance

Sec 6 Monitor & feedback

Sec 7 Challenges and Future Trends

Query Representation
Query Understanding

User/Context Profiles
User Understanding

Doc Representation
Doc Understanding

Challenges and Future Trends
Challenges and Trends

Log Mining Applications
- Example: Interpreting user’s search intent

Log Mining Infrastructure
- Example: Specially designed programming languages for log mining

Search and Browse Log Data
- Example: Privacy preserving release of log data
Why Intent Interpretation

• In traditional IR, queries are represented by their terms
• In Web search, queries are often short and ambiguous
• Accurately interpreting user intent from non-perfect user queries is key to search engines
• Trend: from term representation to intent representation
Previous Approaches to User Intent Representation

• By query classification

• By existing knowledge base

• By queries and clicked documents from log data
Representing Intent by Knowledge Base

• An intent is represented by a collection of Wikipedia concepts and categories [Hu09]

• Accurate and systematic, but may not adapt well to new search intents emerging from the users

Wikipedia nodes to represent the travel intent

Mining Intents from Log Data

• Mining click-through bipartite
  – Each intent is represented by a collection of similar queries as well as the clicked URLs [Cao09]

• Automatically adapt to emerging user interests, but results not as good as human edited knowledge base

Cao, H., et al. Towards context-aware search by learning a very large variable length hidden markov model from search logs. WWW’09.
Future Work on Intent Representation

• Towards a systematic framework to represent intents
  – Including the multiple dimensions mentioned before
  – Hierarchical granularity
    • Example: domain → topic → concept
  – Automatically adapting to emerging intents

• Joint efforts of log mining, Web page mining, semantic Web, and natural language processing

For more information, please refer to WWW’10 tutorial “Recent Progress on Inferring Web Searcher Intent” by Eugene Agichtein.
Challenges and Trends

- Log Mining Applications
  - Example: Specially designed programming language for log mining

- Log Mining Infrastructure

- Search and Browse Log Data
  - Example: Privacy preserving release of log data
Map-Reduce Programming Model

- It is impossible to mine huge amounts of log data on a single machine
- Map-reduce programming model

Dean, J. and Ghemawat, S. MapReduce: simplified data processing on large clusters. OSDI’04.
Challenges for Map-Reduce Model

• Some applications are hard to be expressed in the map-reduce model
  – E.g., multiplication of large-scale matrices

• Difficult for system to automatically optimize execution plans
  – Some complex applications may involve multiple steps of map and reduce
  – Implementation in C++ or C#
The SCOPE Language

• An example: find the queries which have been requested for at least 1,000 times

```
SELECT query, COUNT(*) AS count
FROM "search.log" USING LogExtractor
GROUP BY query
HAVING count > 1000
ORDER BY count DESC;
```

• Similar to SQL
• No need to decompose a job into map and reduce
• Optimization rules borrowed from database community

Chaiken, R., et al. SCOPE: easy and efficient parallel processing of massive data sets. VLDB'08
Challenge: Global Optimization

- The log mining system may receive hundreds of jobs for each day
  - Many jobs may consume the same data and share similar computation steps
  - The optimization rules borrowed from database system target at optimizing a single job

Global optimization needed
Challenges and Trends

Log Mining Applications

Log Mining Infrastructure

Search and Browse Log Data

Example: Privacy preserving release of log data
The AOL Data Release

• AOL data release, 2006
  – 650K users, 20 million Web search queries
  – Users anonymized by hash functions, IP omitted
  – User No. 4417749 was identified through her query history by a newspaper journalist
    • CTO resigned, 2 employees fired
    • Class action law suit pending
    • CNN Money: “101 dumbest moments in business”

Private Information in Log Data

• Direct identity information
  – Social security number, credit card number, driver’s license number, address, phone number, email address, etc

• Indirect identity information
  – DOB, zip code, gender, age, etc
  – May identify a person when joined with other data sources

• Potentially sensitive subjects
  – Health condition, financial condition, political affiliation, religious affiliation, etc
  – Dependency on identity information

Privacy-Preserving Log Release

• Typical steps in breaching privacy
  – Step 1: link all query terms related to an individual
  – Step 2: find private information from the linked queries

• Approaches to privacy-preserving log releasing
  – Breaking the linkage of queries, e.g., [Korolova09]
  – Scrubbing private information from queries, e.g., [Kumar07][Adar07][Xiong07]
Future Directions (1)

• Metrics for privacy and utility
  – Any privacy-preserving technique is a tradeoff between privacy and utility
    • Example: deleting user IDs => losing sessions
  – Each previous work targeted at particular applications and had its own definitions of privacy and utility
  – Need explicit and general metrics for privacy and utility

Xiong L. and Agichtein, E. Towards privacy-preserving query log publishing. Workshop at WWW’07.
Future Directions (2)

• Approaches from other communities
  – Database community
  – Network community

Xiong L. and Agichtein, E. Towards privacy-preserving query log publishing. Workshop at WWW’07.
Future Directions (3)

• Technical solutions plus policy-based protections
  – Privacy laws
  – Privacy policies
  – Confidentiality and licensing agreements
  – Institutional review boards

Cooper, A. A survey of query log privacy – enhancing techniques from a policy perspective. ACM Transactions on the Web. 2008
Summary for Challenges and Future Trends

• A log mining system consists of three layers
  – Application layer
    • Example: from text matching to intent matching
  – Infrastructure layer
    • Example: Specially designed programming languages for log mining
  – Data layer
    • Example: Privacy preserving release of log data
Summary of Tutorial

• Section 1: introduction
  – Search and browse logs, overview of log mining applications, four frequently used summarizations on log data

• Sections 2-5
  – How log data enhance query understanding, document understanding, query-document matching, user understanding

• Section 6
  – Another important function of log data: monitor and feedback

• Section 7
  – Challenges and future trends
References

• [Agichtein06] Agichtein, E, et al. Learning user interaction models for predicting web search result preferences. SIGIR'06
• [Arrington06] Arrington, M. AOL proudly releases massive amounts of user search data. TechCrunch. 2006
• [Beeferman00] Beeferman, D. and Berger, A.L. Agglomerative clustering of a search engine query log. KDD'00
• [Bilenko08] Bilenko, M. and White, R. W. Mining the search trails of surfing crowds: identifying relevant websites from user activity. WWW’08
• [Broder02] Broder, A. A Taxonomy of Web Search. SIGIR'02
• [Chaiken08] Chaiken, R., et al. SCOPE: easy and efficient parallel processing of massive data sets. VLDB'08
• [Chapelle09] Chapelle, O. and Zhang, Y. A Dynamic Bayesian Network Click Model for Web Search Ranking. WWW'09
References

• [Chien05] Chien, S. and Immorlica, N. Semantic similarity between search engine queries using temporal correlation. WWW’05
• [Ciaramita08] Ciaramita, M., et al. Online learning from click data for sponsored search. In WWW’08
• [Cid06] Cid, A., et al. Automatic maintenance of web directories using click-through data. ICDEW ’06
• [Craswell07] Craswell, N. and Szummer, M. Random walks on the click graph. SIGIR’07
• [Craswell08] Craswell, N., et al. An experimental comparison of click position-bias models. WSDM’08
• [Cui02] Cui, H., et al. Probabilistic query expansion using query logs. WWW'02
• [Dean04] Dean, J. and Ghemawat, S. MapReduce: simplified data processing on large clusters. OSDI’04.
• [Dou07] Dou, Z. et al. A large-scale evaluation and analysis of personalized search strategies. WWW’07.
• [Dupret08] Dupret, G. E. and Piwowarski, B. A user browsing model to predict search engine click data from past observations. SIGIR’08
• [Fonseca05] Fonseca, B. M., et al. Concept-based interactive query expansion. CIKM’05
• [Fox05] Fox et al. Evaluating implicit measures to improve web search. TOIS’05.
• [Fuxman08] Fuxman, A., et al. Using the wisdom of the crowds for keyword generation. WWW’08
References

• [Jansen06] Jansen, B.J. and Spink, A. How are we searching the world wide web? a comparison of nine search engine transaction logs. Information processing & management 2006
• [Joachims02] Joachims, T. Optimizing search engines using clickthrough data. KDD ’02
• [Joachims05] Joachims, T., et al. Accurately interpreting clickthrough data as implicit feedback. SIGIR’05
• [Jones06] Jones, R., et al. Generating query substitutions. WWW'06
• [Li08] Li, X., et al. Learning query intent from regularized click graphs. SIGIR’08.
• [Liu02] Liu, F., Yu, C., and Meng, W. Personalized web search by mapping user queries to categories. CIKM’02.
• [Liu08] Liu, Y., et al. BrowseRank: letting web users vote for page importance. SIGIR'08
• [Liu09] Liu, C., et al. BBM: bayesian browsing model from petabyte-scale data. KDD’09
• [Pasca07] Pasca, M. Organizing and searching the world wide web of facts--step two: harnessing the wisdom of the crowds. WWW'07
References

- [Pasca07a] Pasca, M. and Durme, B. V. What you seek is what you get: extraction of class attributes from query logs. IJCAI’07
- [Piwowarski09] Piwowarski, B., et al. Mining user web search activity with layered Bayesian networks or how to capture a click in its context. WSDM’09.
- [Poblete08] Poblete, B. and Yates, R. B. Query-sets: using implicit feedback and query patterns to organize web documents. WWW’08
- [Radlinski07] Radlinski, F. and Joachims, T. Active exploration for learning rankings from clickthrough data. KDD’07
- [Radlinski08] Radlinski, F. and Joachims, T. Minimally invasive randomization for collecting unbiased preferences from click-through data. AAAI’08
- [Radlinski05] Radlinski, F and Joachims. T Query chains: learning to rank from implicit feedback. KDD’05
- [Richardson07] Richardson, M., et al. Predicting clicks: estimating the click-through rate for new Ads. WWW’07
- [Speretta05] Speretta, M. and Gauch, S. Personalized Search Based on User Search Histories. WI’05.
- [Sun05] Sun et al. CubeSVD: a novel approach to personalized web search. WWW’05.
- [Sun05a] Sun, J.-T., et al. Web-page summarization using clickthrough data.SIGIR ’05
References

- [Teevan05] Teevan, J., et al. Personalizing search via automated analysis of interests and activities. SIGIR’05
- [Teevan08] Teevan, J. et al. To personalize or not to personalize: modeling queries with variation in user intent. SIGIR’08.
- [Vlachos04] Vlachos, M., et al. Identifying similarities, periodicities and bursts for online search queries. SIGMOD’04
- [Wang07] X. Wang and C. Zhai. Learn from web search logs to organize search results. SIGIR’07
- [White07] White, R. W., et al. Studying the use of popular destinations to enhance web search interaction. SIGIR’07
- [Xiong07] Xiong L. and Agichtein, E. Towards privacy-preserving query log publishing. Workshop at WWW’07.
- [Xue04] Xue, G. R., et al. Optimizing web search using web click-through data. CIKM'04
- [Yi09] Yi, X., et al. Discovering users' specific geo intention in Web search. WWW'09
- [Zhao06] Zhao, M., et al. Adapting document ranking to users' preferences using click-through data. AIRS’06
- [Zhu09] Zhu, G. and Mishne, G. Mining rich session context to improve web search. KDD'09