IIIT-H

Web Mining
Lecture 24: Query-Document Matching by Log Mining

Manish Gupta

6th Nov 2013

Slides borrowed (and modified) from
Daxin Jiang, Jian Pei, Hang Li. Web Search/Browse Log Mining: Challenges, Methods, and Applications. Tutorial at WWW 2010

Recap of Lecture 23: Document Understanding by Log Mining

- Query Expansion, Refinement, and Suggestion
- Temporal and Spatial Aspects of Queries
- Text Mining from Query Logs

Announcements

Today's Agenda

- Learning user preferences from logs
- Modeling and predicting clicks

Today's Agenda

- **Learning user preferences from logs**
- Modeling and predicting clicks

Clicks and Preferences

- 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 satisfy the user's information need
- User clickthrough data provides implicit feedback and hints about user preference on search results

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

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, …, d_n) and a set C of clicked results, extract a preference relation $d_i < d_j$ for $1 \leq j < i$, $i \in C$, and $j \notin C$
- 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, …, 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 $i + 1 \notin C$, $R(d_i, d_{i+1})$
- T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. SIGIR'05.

Kendall's τ

- 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) ≠ 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?

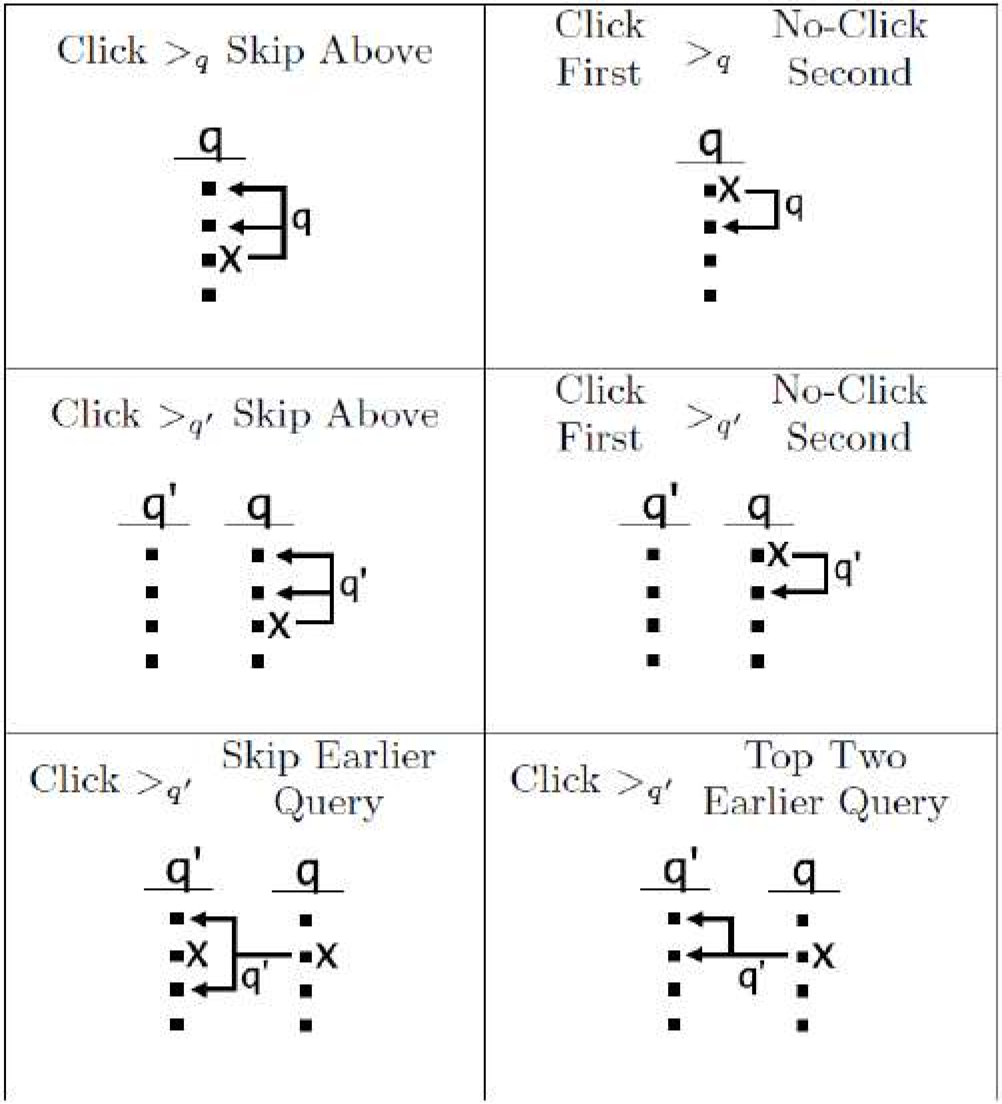
- For a preference relation R, the average precision of R is bounded by
$$AvgPrec(R) \geq \frac{1}{l}[Q + \binom{l+1}{2}]^{-1}\left(\sum_{i=1}^{l}\sqrt{i}\right)^2$$
where l is the number of relevant documents
- Learn a preference relation R maximizing
$$\int \tau(R_q, R^*) d\Pr(q, R^*)$$
- However, 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.

Feedback Strategies



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

Example

q1	q2
d1	d4 x
d2 x	d5
d3	d6

$$\boxed{\begin{array}{lll} d_2 >_{q1} d_1 & d_4 >_{q2} d_5 & d_4 >_{q1} d_5 \\ d_4 >_{q1} d_1 & d_4 >_{q1} d_3 & \end{array}}$$

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

Using Aggregated Clickthrough Data

- The preferences learned from individual use 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 $cdif(q, d_i, d_j) > 0$, $d_i >_q d_j$
- Z. Dou, R. Song, X. Yuan, J-R Wen. Are click-through data adequate for learning web search rankings? CIKM'08.

Presentation Bias

- A user is more likely to click on documents presented higher in the result set irrespective of relevance
  - T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. SIGIR'05.
- 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 Radlinkski and Thorsten Joachims. Minimally invasive randomization for collecting unbiased preferences from click-through data. AAAI'08.

Why FairPairs Works?

- Let c_{ij} be the number of times a user clicks on di 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.

Clickthrough for Sponsored Search

- Preference learning problem also exists for sponsored search
  - Which ads are more likely to be clicked by a user with respect to a query?
- Machine learning approaches can be used
  - How to use click data for training and evaluation?
  - Which learning framework is more suitable for the task?
  - Which features are useful for existing methods?
- Ciaramita, M., et al. Online learning from click data for sponsored search. In WWW'08, 2008.

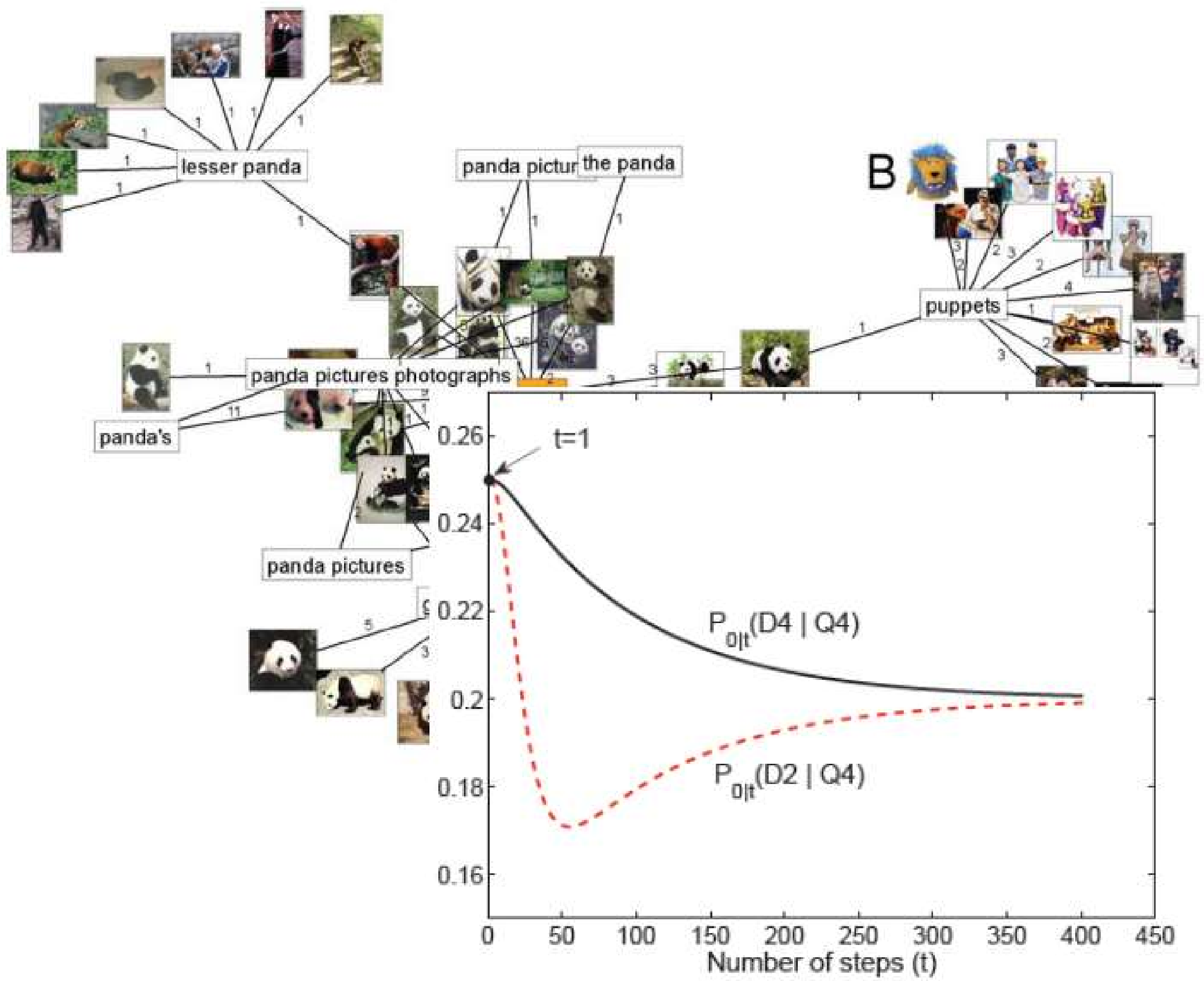
Learning Listwise Preferences

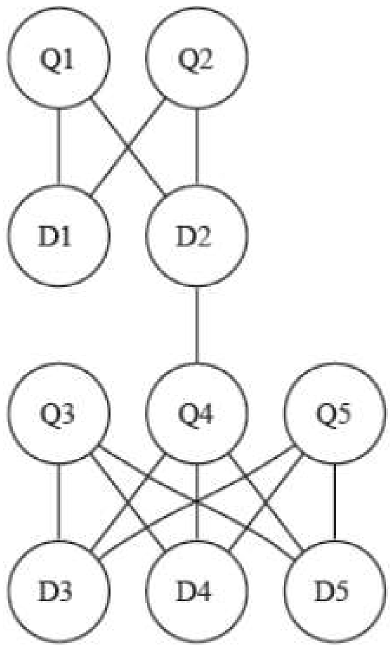
- Pairwise preferences are easy to learn, but may not generate a ranked list
  - Given a > b, b > c, and c > a, no ranking can be generated
- Learning listwise preferences: for a given query, produce a ranking of documents
  - Using listwise preferences a search engine can retrieve relevant documents that have not yet been clicked for that query, and rank those documents effectively

A Markov Random Walk Method

- Query-document bipartite graph
- The random walk process
  - A user imagines a single document to represent the user's information need, and thinks of a query associated with the document, and issues the query
  - Alternatively, the query makes the user imagine another document, and that document makes the user imagine another query
- The model produces a probabilistic ranking of documents for a query
- N. Craswell and M. Szummer. Random walks on the click graph. SIGIR'07.

Clustering Effect



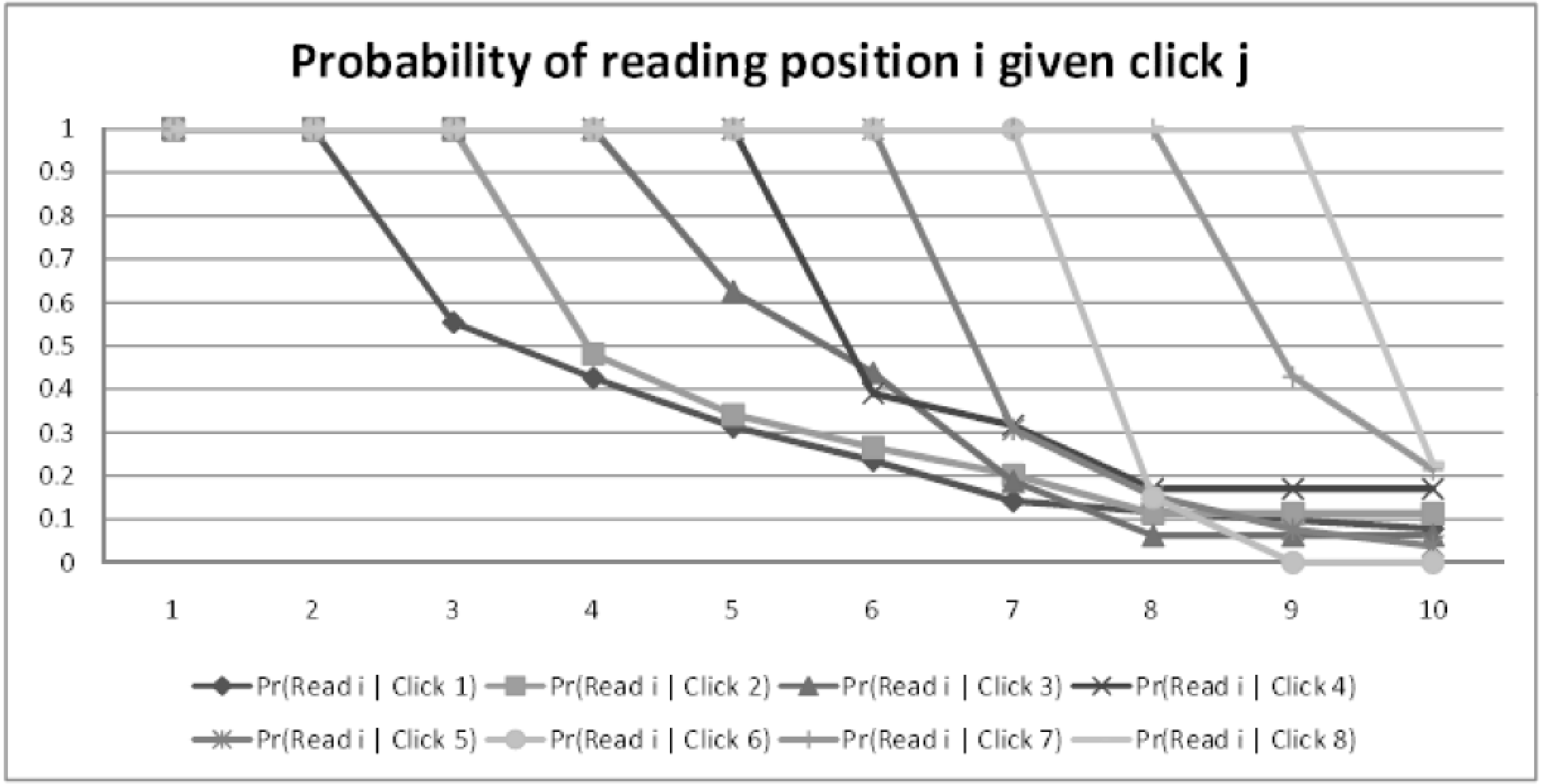


- N. Craswell and M. Szummer. Random walks on the click graph. SIGIR'07.

Learning from Labeled Data

- In search engines, a ranking function is learned from labeled training data
  - A training example is a (query, URL) pair labeled by a human judge who assigns a score of “perfect”, “excellent”, etc.
- Clickthrough data can be used to generate good labels automatically
  - Generate preferences between URLs for a given query with probability proportional to the probability a user reads position i given that the user clicks on position j
  - Create a per query preference graph: vertices are URLs, and a directed edge $u \rightarrow v$ indicates the number of users who read u and v, clicked u and skipped v
- R. Agrawal et al. Generating labels from clicks. WSDM'09.

Click-Read Probability



- The probability a user reads position i given that the user clicks on position j
- R. Agrawal et al. Generating labels from clicks. WSDM'09.

Computing Labels

- Using pairwise preferences
- Given a directed graph G(V, E), and an ordered set A of K labels, find a labeling L such that the net agreement weight is maximized
- $A_G(L) = \sum_{u \to v} w_{u \to v} - \sum_{u \to v} w_{v \to u}$
  - NP-hard in general
  - Can be solved in time O(|E|) when K = 2
- R. Agrawal et al. Generating labels from clicks. WSDM'09.

Summary

- User clickthrough data provides implicit feedback and hints about user preference on search results
- Pair-wise versus list-wise preferences
- Clickthrough information used in learning
  - A click → a series of clicks → 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., medical insurance bill) may be more preferable
  - How to model, learn, and apply such preferences?

Today's Agenda

- Learning user preferences from logs
- **Modeling and predicting clicks**

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 $P(c|r, u, q)$ that a document u presented at position r is clicked by a user who issued a query q
- A related problem CTR modeling/prediction
  - CTR: number of clicks per display
  - CTR can be used to select the best document in some applications such as the Today module on Yahoo! Front page

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|u, q) or skipped with a probability 1 – P(a|u, q)
  - A user clicks never comes back; a user skips always continues
    - $P(c|r,u,q) = P(a|u,q)\prod_{i=1}^{r-1}(1 - P(a|u_i,q))$
- Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparision 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
  - $P(a|u, q) = \alpha_{uq}^{a} (1 - \alpha_{uq})^{1-a}$
  - $P(e|r, d) = \gamma_{rd}^{e}(1 - \gamma_{rd})^{1-e}$
  - α_{uq} is the probability of attractiveness of snippet u if presented to a user who issued query q
  - γ_{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.

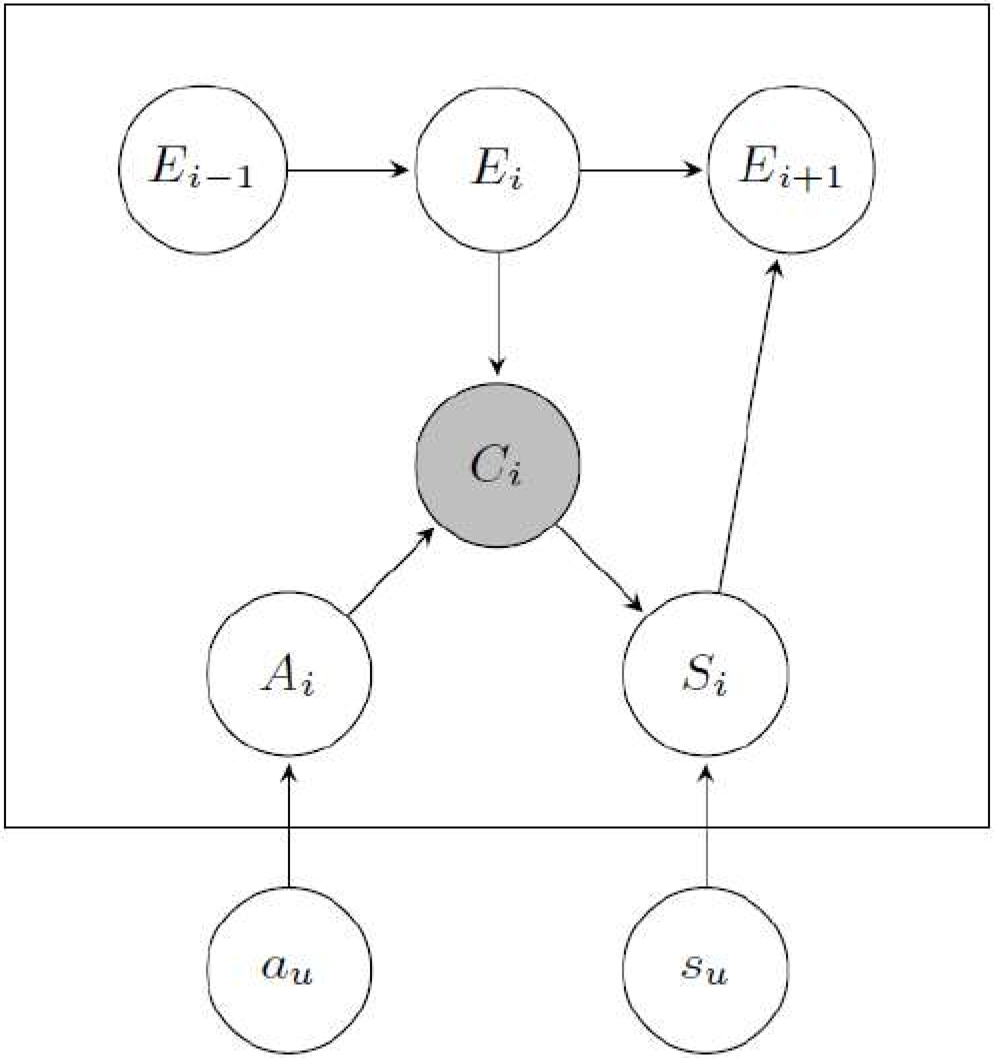
Multiple Browsing Model

- 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 = $\beta_{uq} + \beta_{rd}$
  - 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



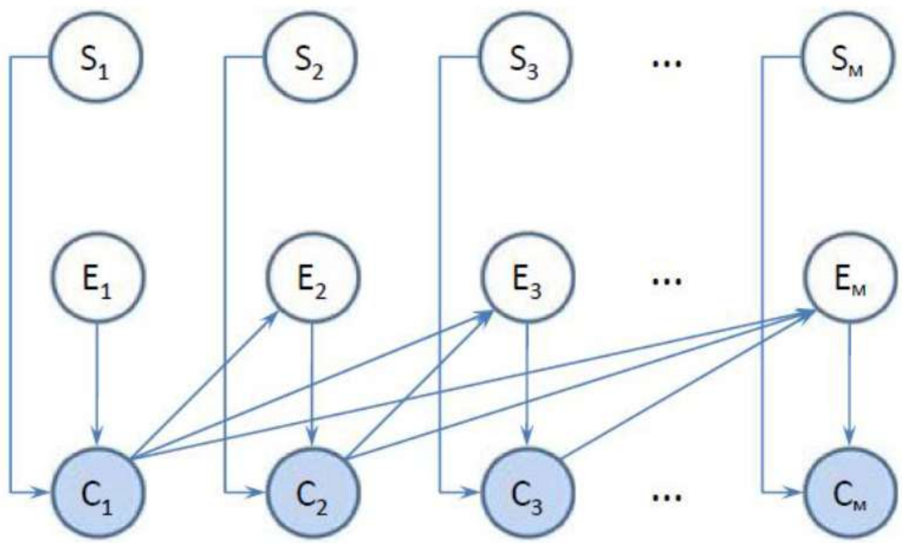
- For a given position i, C_i is the only observed variable indicating whether there was a click or not at this position. E_i, A_i, and S_i are hidden binary variables modeling whether the user examined the URL, the user was attracted by the URL, and the user was satisfied by the landing page, respectively
- Chapelle, O. and Zhang, Y. A Dynamic Bayesian Network Click Model for Web Search Ranking. WWW'09.

Handling Huge Amounts of Data

- Scalability: how to handle terabyte- or even petabyte-scale data
- Parallelizability: can a model be implemented in a parallelizable way?
- Incremental updatability: can it be single-pass computable?
- Chao Liu, Fan Guo, Christos Faloutsos. BBM: bayesian browsing model from petabyte-scale data. KDD'09.

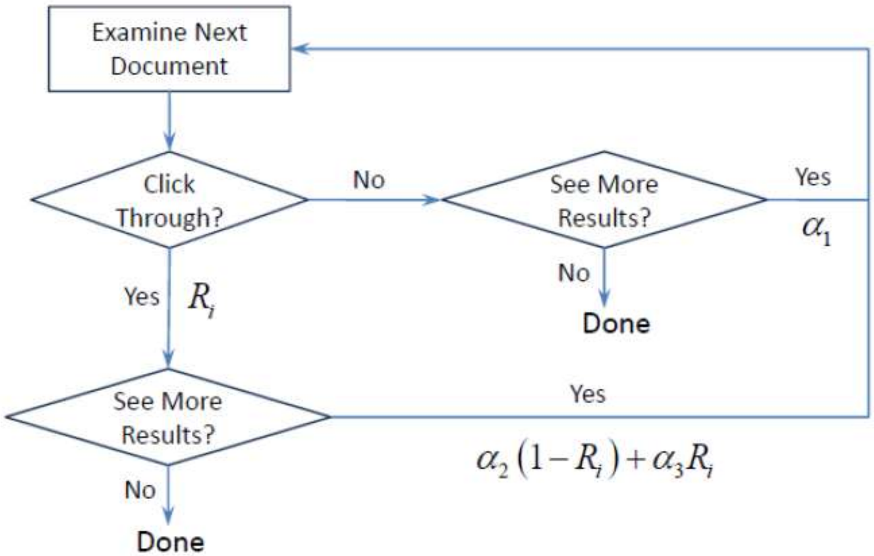
BBM: A Bayesian Browsing Model

- A single pass suffices for computing global parameters and inferring document relevance
- Exact posterior for document relevance can be derived in closed form

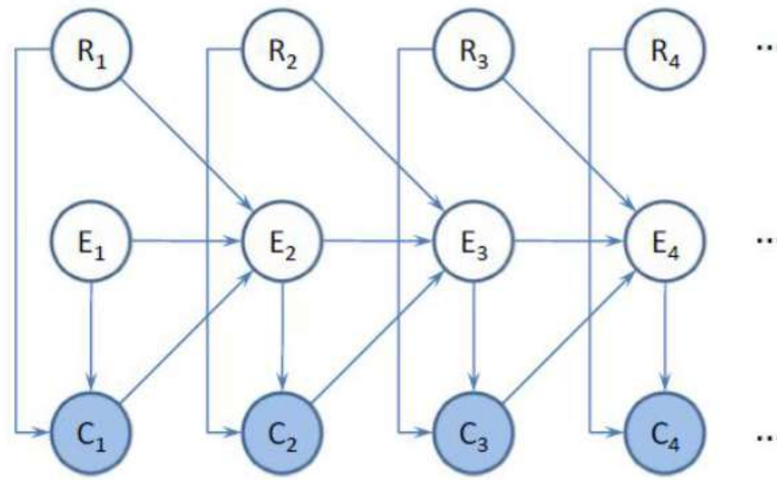


- Chao Liu, Fan Guo, Christos Faloutsos. BBM: bayesian browsing model from petabyte-scale data. KDD'09.

Click Chain Model



The generative process



The graphical model representation

- R_i is the relevance variable of d_i at position i, and α's form the set of user behavior parameters
- Fan Guo, et al. Click chain model in web search. WWW'09.

CTR Modeling/Prediction for Ads

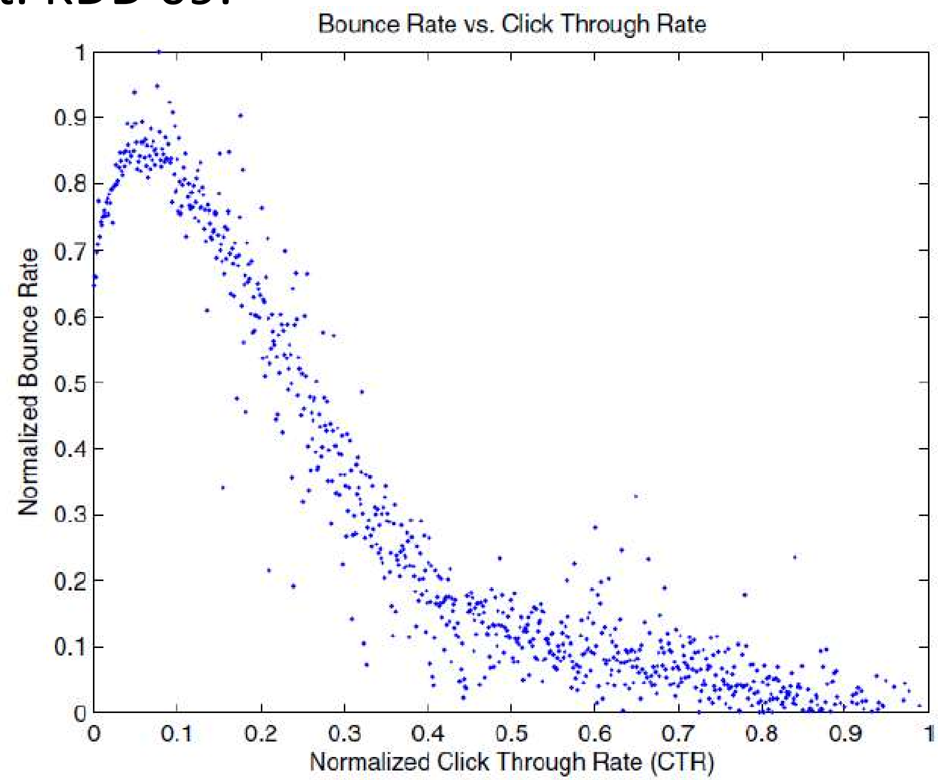
- CTR = P(click|ad, pos) = P(click|ad, seen)P(seen|pos)
- Using logistic regression, we have
  - $CTR = \frac{1}{1+e^{-\sum_i w_i f_i(ad)}}$
  - $f_i(ad)$ is the value of the i-th feature for the ad, and w_i is the learned weight for that feature
- Features
  - Term CTR: the CTR of other ads that have the same bid terms
  - Related term CTR: the CTR of the ads bidding on “buy red shoes” is related to the CTR of the ads bidding on “red shoes”
  - …
- Mattew Richardson, Ewa Dominowska, Robert Ragno. Predicting clicks: estimating the click-through rate for new Ads. WWW'07.

Spatio-Temporal Models

- For a fixed location over time, use a dynamic Gamma-Poisson model
- Combine information from correlated locations through dynamic linear regressions
- Deepak Agarwal, Bee-Chung Chen, Pradheep Elango. Spatio-temporal models for estimating click-through rate. WWW'09.

Bounce Rates

- For an ad, the bounce rate is the fraction of users who click on the ad but almost immediately move on to other tasks
  - A poor bounce rate leads to poor advertiser return on investment and poor search engine user experience following the click
- Sculley, D., et al. Predicting bounce rates in sponsored search advertisement. KDD'09.



Bounce Rate Prediction

- Features
  - Parsed terms, extracted from content, scored using TF-IDF
  - Related terms, derived from the parsed terms using a transformation similar to term expansion via latent semantic analysis (LSA)
  - Cluster/category membership, the strength of similarity of a given piece of content to a set of topical clusters and a semi-automatically constructed hierarchical taxonomy
  - Shannon Redundancy, how focused a piece of content is
  - Binary cosine similarity between content groups
  - Binary KL-divergence for term-based relevance
- Using a logistic regression approach
- Sculley, D., et al. Predicting bounce rates in sponsored search advertisement. KDD'09.

Summary

- Click-bias on presentation order
  - Click (bias) modeling and CTR prediction
- Click models
  - Examination hypothesis
  - Cascade model
  - Single/multiple browsing models, logistic model
  - Dynamic Bayesian Network model
  - BBM/Click chain model: scalability, exact inference-ability, and parallelizability
- CTR prediction
  - Spatio-temporal models
  - Bounce rate prediction

Take-away Messages

- Logs can be very useful for improving the query-document matching task
- We studied two ways of using logs for this purpose
  - Learning user preferences from logs
  - Modeling and predicting clicks

Further Reading

- Daxin Jiang, Jian Pei, Hang Li. Web Search/Browse Log Mining: Challenges, Methods, and Applications. Tutorial at WWW 2010
- Daxin Jiang, Jian Pei, Hang Li. Mining Search and Browse Logs for Web Search: A Survey. ACM Transactions on Computational Logic, Vol. V, No. N, February 2013, Pages 1–42.
- Maristella Agosti, Franco Crivellari, Giorgio Maria Di Nunzio. Web log analysis: a review of a decade of studies about information acquisition, inspection and interpretation of user interaction. Data Min Knowl Disc (2012) 24:663–696
- Fabrizio Silvestri. Mining Query Logs: Turning Search Usage Data into Knowledge. Foundations and Trends in Information Retrieval. Vol. 4, Nos. 1–2 (2010) 1–174
- Marius Pasca. Tutorial. Web Search Queries as a Corpus. ACL 2011
- Ricardo Baeza-Yates, Fabrizio Silvestri. Query Log Mining.

Preview of Lecture 25: User Understanding by Log Mining

- Personalized search
- User behavior modeling
- Privacy in Web Search Query Log

Disclaimers

- This course represents opinions of the instructor only. It does not reflect views of Microsoft or any other entity (except of authors from whom the slides have been borrowed).
- Algorithms, techniques, features, etc mentioned here might or might not be in use by Microsoft or any other company.
- Lot of material covered in this course is borrowed from slides across many universities and conference tutorials. These are gratefully acknowledged.

Thanks!