

2 Data structures for indexing

In what data structure (e.g., inverted lists, web graph, direct index, etc.) the following quantities should be stored? Explain your answer.

- 2.0p a The part of speech of a term (e.g., noun, adverb, etc).

- 2.0p b The language model probability of a term given a document, i.e., $p(t \mid d)$.

- 2.0p c The number of incoming hyperlinks for a document.

3 Click models

Consider the following cascade-based click model (it is called Dependent Click Model or DCM for short):

$$P(C_r = 1) = P(A_{d_r} = 1) \cdot P(E_r = 1)$$

$$P(A_{d_r} = 1) = \alpha_{qd_r}$$

$$P(E_1 = 1) = 1$$

$$P(E_{r+1} = 1 \mid E_r = 0) = 0$$

$$P(E_{r+1} = 1 \mid C_r = 1) = 1 - \lambda_r$$

$$P(E_{r+1} = 1 \mid E_r = 1, C_r = 0) = 1.$$

Here, C_r is a binary random variable representing a click at rank r , A_{d_r} is a binary random variable showing whether a document at rank r is attractive, E_r is a binary random variable showing whether rank r is examined, α_{qd_r} is the attractiveness parameter for query q and document d_r , and λ_r is a parameter that depends on rank r , i.e., there are as many parameters λ_r as there are ranks.

The DCM click model is similar to the cascade model, but it allows more than one click. Essentially, DCM says that even after a user clicked on some document, i.e., $C_r = 1$, she may continue examining other document below with probability $P(E_{r+1} = 1 \mid C_r = 1) = 1 - \lambda_r$.

- 5.0p a Represent the full examination probability $P(E_{r+1} = 1)$ using the DCM attractiveness parameters $\{\alpha_{qd}\}$ and examination parameters $\{\lambda_r\}$. In other words, represent the examination probability in the following form:

$$P(E_{r+1} = 1) = [\text{only parameters here, no probabilities}]$$

Present a complete derivation (not only the end result).

- 2.0p b Represent the full click probability $P(C_r = 1)$ using the DCM attractiveness parameters $\{\alpha_{qd}\}$ and examination parameters $\{\lambda_r\}$. In other words, represent the click probability in the following form:

$$P(C_r = 1) = [\text{only parameters here, no probabilities}]$$

Present a complete derivation (not only the end result).

You have a click log with submitted queries, returned search results, and clicks on these results (**NO** other information is available in the log). Assume that when a user examines search results from top to bottom according to the DCM model, she stops after she makes the last click. For example, a user was presented with 10 search results and she clicked as follows: [1, 1, 0, 0, 1, 0, 0, 0, 0, 0]. The above assumption means that the user stops examining results after clicking on the 5th result.

- 1.0p c Given the above click log (containing queries, search results and clicks), the DCM model, and the additional assumption above, how can you understand which documents in the log were examined by users?

- 1.0p d Given the above click log (containing queries, search results and clicks), the DCM model, and the additional assumption above, how can you understand which documents in the log were attractive to users?

- 1.0p e Propose a formula that calculates the parameters α_{qd_r} based on the above click log. The formula should be fully computable given the log, i.e., it should give a number for each parameter α_{qd_r} based on the queries, search results and clicks in the log.

- 1.0p f Propose a formula that calculates the parameters λ_r based on the above click log. The formula should be fully computable given the log, i.e., it should give a number for each parameter λ_r based on the queries, search results and clicks in the log.

4 Counterfactual evaluation

Consider the DCM click model from the previous question (the exact definition of DCM is not important here). Consider also counterfactual evaluation.

- 1.0p a How should DCM be used to calculate the observance probability $P(o_i = 1 \mid R, d_i)$? (Here, R denotes a ranking and d_i denotes a document).

- 1.0p b What part of the DCM model corresponds to the probability of click given observance $P(c_i = 1 \mid o_i = 1, y(d_i))$? (Here, $y(d_i)$ denotes the true relevance of document d_i).

- 1.0p c How should DCM be used to calculate the click probability $P(c_i = 1 \mid o_i, y(d_i))$?

5 Offline evaluation, metrics

Consider the following offline evaluation metric based on the DCM click model from two previous questions (the exact definition of DCM is not important here):

$$Metric_{DCM} = \sum_{r=1}^n P(C_r = 1) \cdot R_{d_r},$$

where n is the number of documents in a result list and R_{d_r} is the relevance of document d_r .

- 2.0p a Come up with one application/search scenario, where the above $Metric_{DCM}$ is most suitable. Explain, why it is most suitable in your application.

- 2.0p b What other offline evaluation metrics are similar to $Metric_{DCM}$ and why?

- 2.0p c Propose two ways to do meta-evaluation of $Metric_{DCM}$, i.e., to measure how good this metric is.

- 1.0p d You would like to use $Metric_{DCM}$ in LambdaRank. How should it be used?

6 Offline evaluation, test collections

- 2.0p Explain, why random sampling is **NOT** a feasible strategy to select documents for relevance assessment?

7 Term-based retrieval

Consider a situation where a whole document is used as a query and, thus, the query is very long.

- 1.0p a The modification of BM25 for long queries is the following, the additional part is highlighted in red:

$$BM25_d = \sum_{t \in q} \log \left[\frac{N}{df(t)} \right] \cdot \frac{(k_1 + 1) \cdot tf(t, d)}{k_1 \cdot \left[(1 - b) + b \cdot \frac{dl(d)}{dl_{ave}} \right] + tf(t, d)} \cdot \frac{(k_3 + 1)tf(t, q)}{k_3 + tf(t, q)}$$

Why/for what reason is the red part added to standard BM25 when long queries are used?

- 2.0p b Explain in detail, how this added part handles long queries.

2.0p c Consider KL-divergence for ranking documents given a long query:

$$KL(d||q) = \sum_{t \in V} P(t | q) \log \frac{P(t | q)}{P(t | d)}$$

where V is the vocabulary of terms. Explain, why this method does **NOT** need any modification to account for long queries?

8 Semantic retrieval and evaluation

Consider the pLSA topic model:

$$p(w | d) = \sum_z P(w | \phi_z) \cdot P(z | \theta_d),$$

where ϕ_z is the distribution of words in topic z and θ_d is the distribution of topics in document d .

Give the values of probabilities $P(w | \phi_z)$ and $P(z | \theta_d)$ for the two cases below. The values should either be numbers or use quantities that can be computed directly from the collection of documents (e.g., collection length, term frequency, etc). Explain your answers.

2.0p a There is only one topic for the whole collection.

2.0p b The number of topics is equal to the size of the vocabulary (i.e., the total number of unique words in the collection).

9 Content-based recommendation and semantic retrieval

- 3.0p You would like to use LSI for content-based recommendation. Explain in detail, how you represent items as vectors using LSI.

10 Online evaluation

The main steps of team-draft interleaving (TDI) are the following:

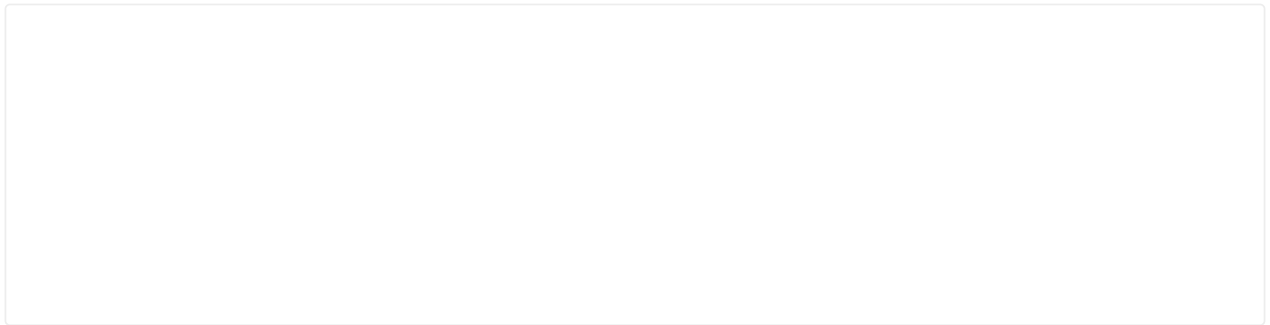
1. Randomly choose ranker A or B.
2. Let chosen ranker place its next unplaced document.
3. Let other ranker place its next unplaced document.

These steps are repeated until a complete interleaved ranking is formed.

Consider a modification of TDI, where the step 3 is removed (let's call it Modified TDI). For each position in the interleaved ranking, Modified TDI randomly chooses between rankers A and B and picks the next unplaced documents from the chosen ranker.

- 1.0p a If users click randomly, the original TDI does not distinguish between rankers A and B, i.e., none of these rankers wins under random clicks. Explain, what happens to this property for Modified TDI and why?

- 2.0p b For the original TDI, there are cases, where a better ranker does not win (see an example below). Explain, what happens in this situation to Modified TDI and why?



2.0p c Explain, why the original TDI should be preferred over Modified TDI.

