IR1 Sample Exam

1 Instructions

0.0 points · 1 question

(omitted)

2 Data structures for indexing

6.0 points · 3 questions

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

Description

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

2.0 points · Open question

+1 point

Vocabulary.

+1 point

This information is specific to the term only.

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

2.0 points · Open question

+1 point

Inverted lists.

+1 point

This information is about a term and a document.

c The number of incoming hyperlinks for a document.

2.0 points · Open question

+1 point

Page attribute file.

+1 point

This information is specific to the document only.

3 Click models

11.0 points · 6 questions

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

Description

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).

5.0 points · Open question

Grading description

NOTE for TAs: 1pt for each line of the following derivation. If 1--2 lines are missing, but the whole derivation and result are correct, give full points.

+1 point

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

+1 point

$=$
$(P(E_r=1 \mid E_r=1, C_r=1)\cdot P(C_r=1) + P(E_r=1 \mid E_r=1, C_r=0)\cdot P(C_r=0))\cdot$
$P(E_r=1)$

+1 point

$= ((1-\lambda_r)\cdot \alpha_{qd_r} + 1 - \alpha_{qd_r})\cdot P(E_r=1)$

+1 point

$= (1-\lambda_r\cdot\alpha_{qd_r})\cdot P(E_r=1)$

+1 point

$= \prod_{i=1}^{r}(1-\lambda_i\cdot\alpha_{qd_i})$

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).

2.0 points · Open question

+1 point

$P(C_r = 1) = P(C_r = 1 \mid E_r = 1) \cdot P(E_r = 1) + P(C_r = 1 \mid E_r = 0) \cdot P(E_r = 0)$

+1 point

$= \alpha_{qd_r} \cdot \prod_{i=1}^{r-1}(1 - \lambda_i \cdot \alpha_{qd_i})$

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.

Description

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?

1.0 point · Open question

+1 point

All documents up to (and including) the last clicked one were examined.

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?

1.0 point · Open question

+1 point

All clicked documents were attractive.

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.

1.0 point · Open question

+1 point

$$\alpha_{qd_r} = \frac{\#\text{ clicks on } d_r \text{ for } q}{\#\ d_r \text{ is shown for } q}$$

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.

1.0 point · Open question

+1 point

$$\lambda_r = \frac{\#\ r \text{ is the last clicked rank}}{\#\text{ there is a click on rank } r}$$

4 Counterfactual evaluation

3.0 points · 3 questions

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

Description

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).

1.0 point · Open question

+1 point

The observance probability is equal to the full examination probability, i.e., $P(o_i = 1 \mid R, d_i) = P(E_{rank(d_i)} = 1)$.

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).

1.0 point · Open question

+1 point

The attractiveness probability, i.e., $P(c_i = 1 \mid o_i = 1, y(d_i)) = P(A_{d_i} = 1)$.

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

1.0 point · Open question

+1 point

$P(c_i = 1 \mid o_i, y(d_i)) = P(E_{rank(d_i)} = 1) \cdot P(A_{d_i} = 1)$

5 Offline evaluation, metrics

7.0 points · 4 questions

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.

Description

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.

2.0 points · Open question

+1 point

$Metric_{DCM}$ can be used in standard search scenarios, where a few relevant documents are important and they need to be ranked as high as possible.

+1 point

For example, search for information about something, search for movies/products, planning a trip, etc.

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

2.0 points · Open question

+1 point

RBP, ERR

+1 point

because they are also based on user models/click models.

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

2.0 points · Open question

Grading description

NOTE for TAs: Any two of the following:

+1 point

Collect user clicks (click log) and check the discount of which metric is closer to the actual click-through rates (CTRs) in the click log.

+1 point

Use side-by-side comparison of search results and check which metric is closer to the outcomes of this comparison.

+1 point

Consider such characteristics of $Metric_{DCM}$ as discriminative power and informativeness.

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

1.0 point · Open question

+1 point

In LambdaRank we need to replace $\Delta NDCG$ with $\Delta Metric_{DCM}$.

6 Offline evaluation, test collections

2.0 points · 1 question

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

2.0 points · Open question

+1 point

There are many more non-relevant documents compared to relevant ones.
Thus, random sampling will mostly sample non-relevant documents and almost never relevant documents.

+1 point

So almost all relevance judgments will be 0 and, thus, offline evaluation cannot be used effectively.

7 Term-based retrieval

5.0 points · 3 questions

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

Description

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?

1.0 point · Open question

+1 point

In a long query, the same term may appear multiple times, so $tf(t,q)$ can be greater than one.

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

2.0 points · Open question

+1 point

Such terms are probably more important than others (inside the query) and so they should have more weight within BM25. The proposed modification does exactly that: gives more weight to terms that occur more often in the query.

+1 point

But it also accounts for cases where $tf(t,q)$ is too large (by having $k_3 + tf(t,q)$ in the denominator).

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

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

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

2.0 points · Open question

+1 point

If $tf(t_1, q) > tf(t_2, q)$, then $P(t_1 \mid q) > P(t_2 \mid q)$.

+1 point

So terms that occur more often in a query have more weight by default. Thus, no modification to KL-divergence is needed.

8 Semantic retrieval and evaluation

4.0 points · 2 questions

Consider the pLSA topic model:

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

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 $P(w \mid \phi_z)$ and $P(z \mid \theta_d)$ 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.

Description

a There is only one topic for the whole collection.

2.0 points · Open question

+1 point

Since there is only one topic, the probability of that topic in each document is 1, i.e., $P(z \mid \theta_d) = 1$.

+1 point

Also, since the topic is one for the whole collection, the probability of a word in that topic is equal to the probability of that word in the collection,
i.e., $P(w \mid \phi_z) = \frac{cf(w)}{cl}$, where $cf(w)$ is the collection frequency of word w and cl is the collection length in the number of words.

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

2.0 points · Open question

+1 point

Since there is a topic for each word, the probability of a word in a topic is likely to be 1, i.e., $P(w \mid \phi_z) = 1$.

+1 point

The importance of each topic in a document is likely to be equal to the importance of the corresponding word in that document, i.e., $P(z \mid \theta_d) = \frac{tf(w_z,d)}{dl(d)}$, where w_z is the word corresponding to topic z.

9 Content-based recommendation and semantic retrieval

3.0 points · 1 question

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

3.0 points · Open question

+1 point

First, the item-term matrix is decomposed into three matrices using singular-value decomposition (SVD).

+1 point

Second, a low rank approximation is computed.

+1 point

Third, a semantic representation of an item is computed
based on its sparse representation and the low rank approximation above.

10 Online evaluation

5.0 points · 3 questions

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.

Description

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?

1.0 point · Open question

+1 point

Since both rankers are equally likely to place a document at each position of the interleaved ranking, there will be no preference between them in the case of random clicks. So Modified TDI is the same as the original TDI here.

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?

2.0 points · Open question

+1 point

No matter how rankers A and B are combined, document 3 (the relevant one) can be picked from one ranker only (because if it is picked, for example, from ranker A, it is removed from ranker B).

+1 point

Thus, the number of times document 3 is picked from ranker A is the same as the number of times it is picked from ranker B. Thus, there will still be no preference between A and B in this example, so Modified TDI is again the same as the original TDI.

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

2.0 points · Open question

+1 point

The behavior of Modified TDI is the same as the original TDI (see the previous two points).

+1 point

However, Modified TDI produces many more interleaved rankings compared to the original TDI. There is no point in producing more interleaved rankings and getting the same result, so the original TDI should be preferred over Modified TDI, because is produces fewer interleaved rankings.