Homework Assignment - Part B

Information Retrieval 1 [2018/2019]

Deadline: Wednesday, January 23nd, 17:00
Submit: An ***IPython Notebook*** with the necessary (a) **implementation**, (b) **code documentation**, and (c) **analysis** of the results. Note: You can run your experiments outside the notebook, so that they run faster. At the end however please copy your code, in an elegant manner, to a notebook to submit.
Filename: <id1>_<id2>_<id3>_code.ipynb

The homework will cover the topics covered in Lecture 1 – 3.

Commercial search engines typically use a funnel approach in evaluating a new search algorithm: they first use an offline test collection to compare the production algorithm (P) with the new experimental algorithm (E); if *E* outperforms *P* with respect to the evaluation measure of their interest, the two algorithms are then compared online through an interleaving experiment.

In an interleaving experiment the ranked results of *P* and *E* (against a user query) are interleaved in a single ranked list which is presented to a user. The user then clicks on the results and the algorithm that receives most of the clicks wins the comparison. The experiment is repeated for a number of times (impressions) and the total wins for *P* and *E* are computed.

A Sign/Binomial Test is then run to examine whether the difference in wins between the two algorithms is statistically significant (or due to chance). Alternatively one can calculate the proportion of times the *E* wins and test whether this proportion, *p*, is greater than $p_0=0.5$. This is called an 1-sample 1-sided proportion test.

The offline test collection is static (i.e. fixed), however when it comes to the online experiment, one would like to know for how long the experiment should run, or in other words how many impressions are necessary so that a statistically significant difference between *E* and *P* can be detected. E.g. if out of 100 impressions *E* wins 55% of them, can this still be due to random chance?

Unfortunately we cannot know a priori (i.e. before actually running the experiment) the proportion of *E* wins against *P*. If we knew we could perform a power analysis of the proportion test and find the necessary number of impressions. What we can know however is by what margin *E* outperformed *P* in the offline experiments.

For the purpose of this homework we assume that: (a) documents in the ranked list are being judged on a binary scale, i.e. {0, 1}, and (b) the evaluation measure of interest is the Expected Reciprocal Rank (ERR) at cut-off rank 3.

Further, for the purpose of this homework we assume that, (a) there are two interleaving algorithm we are interested in, the Team-Draft and the Probabilistic Interleaving, and (b) users click on documents on the basis of two models, the Random Model and the Position-Based Model. We will use the Random Click Model for sanity check of the experiment.

In this homework we will determine the sample size (i.e. the number of interleaved impressions) required for a Probabilistic Interleaving experiment to identify statistical significant differences, when it is known that in offline evaluation *E* outperformed *P* by a certain margin measured by the Expected Reciprocal Rank. In particular, we are looking at filling in the following table.

ΔERR	Number of Impressions Interleaving Method/Click Model		
	Min	Median	Max
[0.05, 0.1)			
[0.1, 0.2)			
[0.2, 0.3)			
...			
[0.8, 0.9)			
[0.9, 0.95]			

Below is a step-by-step design of an experiment that can answer the aforementioned question.

Step 1: Simulate Rankings of Relevance for *E* and *P*

In the first step generate pairs of rankings, for the production *P* and experimental *E*, respectively. Assume a binary relevance. Make no assumption regarding the documents returned by the two algorithms (they can be distinct but they may also overlap). Further, assume that the algorithms are used on mobiles, so we are interested only in rankings of length 3.

Step 2: Calculate the *Δmeasure*

Implement the aforementioned measure, ERR.

For all *P* and *E* ranking pairs constructed above calculate the difference: *Δ*measure = measure_E-measure_P. Consider only those pairs for which *E* outperforms *P* and group them such that group 1 contains all pairs for which $0.05 < \Delta measure \leq 0.1$, group 2 all pairs for which $0.1 < \Delta measure \leq 0.2$, etc.

Step 3: Implement Team-Draft Interleaving (5pts) and Probabilistic Interleaving (35 points)
Implement Team-Draft and Probabilistic Interleaving, with methods that interleave two rankings, and given the users clicks on the interleaved ranking assign credit to the algorithms that produced the rankings.

Step 4: Simulate User Clicks (40 points)
Having interleaved all the ranking pairs in each group (for each measure) an online experiment could be ran. However, given that we do not have any real users (and the entire homework is a big simulation) we will simulate user clicks.

Consider a click model, namely the Position Based Model (PBM). The parameters of PBM can be estimated based on the Expectation-Maximization (EM) method. Implement PBM so that (a) there is a method that learns the parameters of the model given a set of training data, (b) there is a method that predicts the click probability given a ranked list of relevance labels, (c) there is a method that decides - stochastically - whether a document is clicked based on these probabilities.

Having implemented the PBM click models, estimate its parameters the Yandex Click Log [file].

After training PBM, use the learnt parameters γ_r , while instead of the α_{uq} learnt use ϵ for the non-relevant documents (for a small value of ϵ) and 1-ϵ for the relevant documents.

Further consider and implement a Random Click Model, which will be used for sanity check.

Step 5: Simulate Interleaving Experiment
Having implemented the click model, it is time to run the simulated experiment.

For each of interleaving experiment run k simulations for each one of the two click models implemented and measure the proportion *p* of wins for E. Group these proportions in the respective group the interleaved ranking came from. The larger the k the better, but also the larger the k the longer it takes to run the experiment; so make a reasonable choice.

Step 6: Compute Sample Size

Use each one of the afore-computed proportions to compute the sample size needed to detect such a proportion in a statistically significant manner. Allow a chance of falsely rejecting the null hypothesis (i.e. concluding that E is better than P, when it is not) of 5% and a chance of falsely not rejecting the null hypothesis (i.e. not concluding that E is better than P, when it is) of 10%. Use the values above for a power analysis of the proportion test, for the 1-sided case.

Step 7: Analysis (20 points)

- Report the aforementioned tables for the Random Click Model and PBM and for the two interleaving methods [5pts];
- Analyze the results and report your conclusions by observing the results of the experiment [10pts];
- Based on the literature, suggest possible improvements to the experimental design and how would you implement them [5pts].

Yandex Click Log File:

The dataset includes user sessions extracted from Yandex logs, with queries, URL rankings and clicks. To allay privacy concerns the user data is fully anonymized. So, only meaningless numeric IDs of queries, sessions, and URLs are released. The queries are grouped only by sessions and no user IDs are provided. The dataset consists of several parts. Logs represent a set of rows, where each row represents one of the possible user actions: query or click.

In the case of a Query:
SessionID TimePassed TypeOfAction QueryID RegionID ListOfURLs

In the case of a Click:
SessionID TimePassed TypeOfAction URLID

SessionID - the unique identifier of the user session.
TimePassed - the time elapsed since the beginning of the current session in standard time units.
TypeOfAction - type of user action. This may be either a query (Q), or a click (C).
QueryID - the unique identifier of the request.
RegionID - the unique identifier of the country from which a given query. This identifier may take four values.
URLID - the unique identifier of the document.
ListOfURLs - the list of documents otranzhirovanny from left to right as they have been shown to users on the page extradition Yandex (top to bottom).