Homework Assignments

Information Retrieval 1 [2015/2016]

Module 1: Evaluation (2pts)

Deadline: Monday, January 11th, midnight
Submit: An IPython Notebook with the necessary (a) **implementation**, (b) **explanations**, (c) **comments**, and (d) **analysis**. Code quality, informative comments, detailed explanations of what each block in the notebook does and convincing analysis of the results will be considered when grading.
Filename: <your id>-hw1.ipynb

The homework will cover the following three topics covered in Lecture 1 and 2:

- Evaluation measures;
- Interleaving;
- Power analysis.

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.

For the purpose of this homework we will assume that the evaluation measures of interest are:

1. Discounted Cumulative Gain at rank 5 (DCG@5),
2. Rank Biased Precision with persistence parameter $\theta=0.8$ (RBP), and
3. Expected Reciprocal Rank (ERR).

Further, for the purpose of this homework we will assume that interleaving algorithm used is the Team-Draft Interleaving.

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 if 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.

In this homework we will determine the sample size (i.e. the number of interleaved impressions) required for a Team-Draft Interleaving experiment to identify statistical significant differences, when it is known that in offline evaluation *E* outperformed the *P* by a certain margin measured by an evaluation measure (e.g. $\Delta DCG@5 = 0.1$).

Step 1: Simulate Rankings of Relevance for *E* and *P*
In the first step you will generate pairs of rankings of relevance, for the production *P* and experimental *E*, respectively. Assume a 5-graded relevance, i.e. {0, 1, 2, 3, 4}. Construct all possible *P* and *E* ranking pairs of length 5.

Example:
P: {0 0 0 0 0}
E: {0 0 0 0 0}

P: {0 0 0 0 0}
E: {0 0 0 0 1}
…
P: {4 4 4 4 3}
E: {4 4 4 4 4}

P: {4 4 4 4 4}
E: {4 4 4 4 4}

Step 2: Calculate the $\Delta measure$
Implement the three aforementioned measures DCG@5, RBP, ERR.

For the three measures and all *P* and *E* ranking pairs constructed above calculate the difference: Δmeasure = 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 $0 < \Delta measure \leq 0.1$, group 2 all pairs for which $0.1 < \Delta measure \leq 0.2$, etc.

For each measure (DCG@5, RBP, and ERR) there will be 10 groups of ranking pairs. Each group may contain more than one ranking pair based on your calculations above.

Example:

(A) For DCG@5:

(a) Group 1 ($0 < \Delta DCG@5 \leq 0.1$):

- P: {0 0 0 0 0}
- E: {0 0 0 0 1}
- …

(b) Group 2 ($0.1 < \Delta DCG@5 \leq 0.2$):

- …

(c) …

(B) For RBP: ...

(C) For ERR: ...

Step 3: Implement Team-Draft Interleaving

Implement Team-Draft 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.

For each measure and group above: Consider all the ranking pairs and generate an interleaved ranking for each pair. (Each interleaved rankings will be of length 10.) Note here that as opposed to a normal interleaving experiment the rankings consist of relevance labels and not urls or docid, hence in this case (a) we will assume that *E* and *P* return different documents, (b) the interleaved ranking will also be a ranking of labels.

Step 4: Simulate User Clicks

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 users (and the entire homework is a big simulation) we will simulate user clicks.

Consider two different click models, (a) the Random Click Model (RCM), and (b) the Simple Dependent Click Model (SDCM). The parameters of these three models can be estimated based on the Maximum Likelihood Estimation (MLE) method. Implement the two models 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.

For the RCM the labels play no role in determining the click on each rank. However, for the SDCM model the labels can replace the attractiveness parameter a_{uq} by setting a_{uq} for a document u in the ranked list equal to: (label of u)/4.

Having implemented the two click models, estimate the parameter p (RCM), and λ_r (SDCM) using the Yandex Click Log [file].

Step 5: Simulate Interleaving Experiment

Having implemented the two click models, it is time to run the simulated experiment.

For each of interleaved ranking run 50 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.

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, as described here.

For each measure, group of *Δ*measure, and click model report the min, max, 5% and 95% quantile and the median sample size.

Step 7: Analysis

- Report your conclusions by observing the results of the experiment;
- Describe any pitfalls or drawbacks of the method;
- Describe other possible methods (beyond the simulated experiment described here) one could use to identify for how long to run an interleaving experiment

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).