CS600.475: Proposal Version 1.1

Machine Learning Systems: Personalized Web Search

Biman Gujral (bgujral1, bgujral1@jhu.edu) Rujuta Deshpande (rdeshpa3,rdeshpa3@jhu.edu)

November 19, 2014

1 Abstract

This project aims to improve web search results through personalization. It involves re-ranking of results on the basis of user profiles comprising their past searches and actions.
The features used are based upon the short term and long term user search history. Short term history is the user's search queries and actions in the same search session. Long term refers to a user's search behavior over multiple sessions.This problem lies in the Learning to Rank class of problems.

2 Methods

2.1 Learning and Prediction

The problem of re-ranking web search results for personalization falls under Learning to Rank which involves algorithms that rank new retrieval results based on rankings in training data. We will do a feature analysis by comparing performances with different feature combinations and Learning to Rank algorithms. These are discussed in more detail in Milestones.

2.2 Evaluation

The evaluation metric used is NDCG (Normalized Discounted Cumulative Gain). It is a metric between 0 and 1 that evaluates the ranking order. It is given by:

$$NDCG_k = \frac{DCG_k}{IDCG_k}$$

where k denotes documents uptil rank k and DCG is Discounted Cumulative Gain, given by:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

where rel is the actual relevance of the document provided by labels and i is the rank given by the algorithm. Thus, a highly relevant document ranked later in the list results in penalization and a low DCG. The Ideal DCG or IDCG gives the best ranking in accordance with the relevance values. Therefore, NDCG = 1 when the ideal ranking is obtained.

3 Resources

3.1 Data

Our project idea is obtained from a Kaggle competition which provides the data. It consists of web search logs of Yandex, a Russian search engine, collected over 30 days. The first 27 days form the train data and the remaining 3 days form the test data.
Each log line in train data includes a *SESSION ID*, a type of record- indicating information about query(Q), a click(C) or meta data(M). Based on the type of record, there are fields like *UserID, QueryID and terms, URLID-DomainID and SERPID*. The field *TimePassed*, indicates the time that has passed since sessions start until the click was made. We use this data to generate feature files per user wherein each line contains the relevance score (0,1,2), query id and feature-value pairs for that query.

3.2 Tools

We plan to use Python for generation of feature files. We also use RankLib library for the Learning to Rank algorithms and scikit-learn for any machine learning libraries we may need. We may use a database to work with the large datasets and store our learned parameters - SQLite.

4 Milestones

4.1 Must achieve

We musts achieve a sorted URL list in order of decreasing relevance for a particular user. We will analyze features (see next subsection) to obtain the most optimum feature combination. We are using RankLib library for running Learning to Rank algorithms (in detail in the next subsection) to obtain NDCG score on these different feature combinations.

4.2 Expected to achieve

The Kaggle competition winners have experimented with various features - both user specific and global. Some of them are the non-personalized rank of the URL, the frequency of query, the position of the query within the user's session, whether the URL has been missed/skipped/clicked, the relevance of the snippet displayed in search results, the time a user spends on a particular URL and others. Many features are obtained as conditional probabilities given a predicate which is a function of variables on user, session, query and url returned.
Our aim is to experiment with different feature subsets and compare their NDCG score to obtain the most relevant combination. We will use Random Forests (pointwise), RankNet (pairwise), AdaRank (listwise) and LambdaMART (listwise) algorithms from the RankLib library for our comparative analysis.

4.3 Would like to achieve

One of the papers we read, speaks about whether personalization is necessary for every query. We would like to improve upon the search results by only personalizing when needed which depends on the type of query issued. This will add to the performance of the ranking system.

5 Final Writeup

The final writeup will disuss in detail the algorithms and feature analysis done based on the results obtained in each setting. Some broad headers will be:

- Abstract and General Approach
- Related work and our approach in comparison to them
- Explanation of Feature Engineering, Algorithm, Evaluation metrics and Result

6 Bibliography

1. David Sontag, Kevyn Collins-Thompson, Paul N. Bennett, Ryen W. White, Susan Dumais, and Bodo Billerbeck. Probabilistic models for personalizing web search. In Proceedings of the fifth ACM international conference on Web search and data mining (WSDM '12).
2. Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In Proceedings of the 16th international conference on World Wide Web (WWW '07).
3. Learning to Rank `http://en.wikipedia.org/wiki/Learning_to_rank`
4. Dataiku's winning Solution to the problem `http://research.microsoft.com/en-us/um/people/nickcr/wscd2014/papers/wscdchallenge2014dataiku.pdf`