Project proposal - Personalized E-commerce Search

Freeze Francis (MT2015044)
Mohammed Haroon D(MT2015065)
Prachi Prakash Mahapatro(MT2015076)

September 29, 2016

1 Aim

To use the historical search logs and purchase history of the user to personalize top-N product search rankings for a set of test users for an e-commerce website.

2 Introduction

Personalized search ranking has recently been receiving a lot of attention from the IR community especially in e-commerce websites where the goal is to rank the products higher for a query if it is more probable that the user will buy the product. The traditional one ranking-for-all approach to search often fails for ambiguous queries that can refer to multiple products like the query "mouse" refers to different products for a biologist and an engineer. For such queries, non-personalized search engines typically try to retrieve a diverse set of results covering as many possible query interpretations as possible. This can result in highly suboptimal search sessions, where actual products that the user is looking for are very low in the returned ranking. In many such cases previous user search history can help resolve the ambiguity and personalize (re-rank) returned results to user-specific information needs. Search logs and user purchase history can be useful to understand the user purchase pattern. This information is collected by the e-commerce websites by default. To encourage research in this area Diginetica has organized the Personalized Web Search Challenge on the www.codalab.com platform. It has released a large scale search log dataset. The challenge is to predict relevance labels and re-rank products returned by an e-commerce search engine on the search engine result page (SERP) using (1) search, browsing, and transaction histories for all users and specifically the user interacting with the search engine in the current session; (2) product meta-data

3 Dataset Description

The dataset includes user sessions extracted from an e-commerce search engine logs, with anonymized user ids, hashed queries, hashed query terms, hashed product descriptions and meta-data, log-scaled prices, clicks, and purchases. There are eight different files described below. The files have the following metadata.

1. **train-queries.csv and test-queries.csv**
   queryId, sessionId, userId, timeframe (time since the first query in a session), duration (page dwell time), eventdate, searchstring.token (empty if it is a query-less case), categoryId (empty if it is a query-full session), items (productIDs returned by the default ranking algorithm on the SERP; this IDs must be re-ranked), is.test (TRUE/FALSE; TRUE if it is a test query), regionId (geographical region of a query; serial).

2. **products.csv**
   productID, priceLog2, product.name.tokens (comma separated hashed product name tokens)

3. **product-categories.csv**
   productCategoryID, productID, categoryID

4. **train-purchases.csv**
   sessionId, timeframe (time since the first query in a session, eventdate, ordernumber (if a user bought several products, there are several records sharing the same ordernumber), itemId (purchased product), userId

5. **train-item-views.csv**
   sessionId, userId, itemId, timeframe, eventdate

6. **train-clicks.csv**
   queryId, timeframe, itemId

3.1 Dataset Statistics

The number of training unique users: 140388
The number of unique query IDs: 636160
The number of sessions: 573,935
The number of products: 134,319,529
The number of products viewed from search (including browsing after SERP): 2,451,565
The average number of products viewed per search session (including browsing after SERP): 4.271
The number of SERP clicks on products: 1,877,542
The average number of SERP click per search session: 3.271

The number of products purchased from search: 18,026
The average number of products purchased from search session: 0.119

4 Evaluation Metric

The goal of this competition is to predict relevance labels and re-rank products returned by an e-commerce search engine on the search engine result page (SERP) using
1. Search, browsing, and transaction histories for all users and specifically the user interacting with the search engine in the current session
2. Product meta-data

Dataset includes both
1. **Query-full:** SERPs returned in response to a query
2. **Query-less:** SERPs returned in response to the user click on some product category. In this case, only products from that category are returned) sessions.
In both cases, action by a user leading to the SERP is referred to as a query. The only difference is that in the "query-less" sessions the query string is empty and only the product category is provided.

Evaluation is done using NDCG (Normalized Discounted Cumulative Gain) measure, which will be calculated using the ranking of products returned by the re-ranking algorithm for each query, and then averaged over all test queries. NDCG is first calculated for each test query. Then, NDCG scores across all queries are averaged. The weight for the "Query-less" case is 0.8 and the weight for the "Query-full" is 0.2 (such weights are used because the category-based query-less search is more important according to DIGINETICA's prior data analysis). The following DCG formula is used:

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{1}$$

The products are labeled using 3 grades of relevance. The labeling is done automatically, based on user-specific actions with the products on the SERP and detailed product pages (click sometime during the session, click on the SERP, purchase):

- **0 : Irrelevant** Grade corresponds to the products with no SERP clicks.
- **1 : Somewhat relevant** Grade corresponds to the products, which were shown on the SERP and clicked by the user.
- **2 : Relevant** Grade corresponds to the products, which were shown on the SERP, clicked by the user, and purchased. If a product was purchased several times (e.g. three items of the same kind), we still use 2 as a relevance grade.

5 Algorithmic Approach

The problem statement has mentioned that there are 'query-less' and 'query-full' sessions. The 'query-less' sessions are the ones where the user have clicked on a category tab on the UI. The weightage for such queries are very high in the final evaluation metric. The 'query-full' sessions are based on some query terms that the user typed to which the default SERP has responded with some products.

We plan to create a model which tries to predict the most likely category given the query string. The dataset has enough information for such model. There are 1217 categories which will be our class labels and the bag of words representation of queries will be the input features. This model can be computed once and reused. Since the number of classes is more, we are thinking of a neural network to do the job. The target labels per query can be found using the "train-queries.csv" dataset which has the items that the default SERP returned. With these items we can find the majority category they belong to, which will be the target label. The query-category model will help us to proceed to a unified model for both "query-full" and "query-less" queries.

Now with the query-category relationship we can filter out products that the query is mostly likely referring to. With these filtered products we can now incorporate the user preference and product popularity to rank the set of products. User preference includes how much he likes a particular category. This can be obtained from the browsing history and we can have a user category activity count which captures how much time he has clicked/viewed/purchased products in that category wherein view/click/purchase is given different weightages. This user category activity can be used to account user-user relationship by using a collaborative filtering approach using appropriate similarity measures. Now for a particular user we can sort products based on the his/her preferences and buying pattern of top similar users. Based on this product ordering, we can now reorder the products recommended by default SERP.

The dataset has significant number of anonymous users. Queries issued by such users cant be modeled using user-user similarity approaches since we dont have any information for anonymous users.Such user queries can be handled by first finding the category using category-query model and then we can find the most popular product in that category to reorder the SERP results. This approach can be used to handle new users thereby getting a way to handle 'cold-start' problem.

Further improvements in the model includes taking into account the session information. Its highly probable that a user will be aligned to a category of products in a session. We can use this information to further sort the product list. The product pages dwell time can be used to further enhance the model.

References

[1] Volkovs, Maksims. "Context Models For Web Search Personalization." arXiv preprint arXiv:1502.00527 (2015).

[2] Masurel, Paul, et al. "Dataiku's solution to yandex's personalized web search challenge." WSCD workshop. Vol. 13. 2014.