What the user wants! - Term Project

Tarun Tater, Madhumathi Komara, Sushravya GM

December 25, 2015

Abstract

Today, with a lot of topics being discussed on the web, it is becoming increasingly difficult for the search engine to know what the user wants. Personalized search addresses this challenge based on the premise that a users general preference may help the search engine to disambiguate the true intention of a query. In this report, we provide a detailed description of building a personalisation module for the search engine to learn a users preference automatically based on his/her past click and time-spending history.

1 Introduction

A vast majority of queries to search engines are short and under-specified. Given the large and growing importance of search engines, personalized search has the potential to significantly improve user experience.

Here we build a model to learn a users interest automatically based on his/her past click and time spending history, which is used to personalize search results for future queries. In the initial stags of modelling, there exist a number of important technical questions to be addressed. First, we need to adopt a reasonable user model that captures how a users click and time-spending history is related to his/her interest. Second, based on this model we use an effective learning method that identifies the users interest by analyzing the users click and time-spending history. Finally, we need to develop an effective ranking mechanism that considers the learned interest of the user in generating the search result.

Our approach involves computing multiple Topic-Sensitive Page-Rank values for each page with the traditional method of computing a single global Page-Rank value for every page. Given a query and a particular user, the search engine picks the most suitable ranking of pages based on Topic-Sensitive Page-Rank values corresponding to the given query and user. We take into account the long-term and short term profiles for each user and use it in the personalised search. We also take into account some of the navigational queries which shouldn't be personalised. The rest of this paper is organized as follows. We provide an overview of the Data set in Section 1 and its analysis in the Section 2. In Section 3, we describe our models and methods. Section 4 contains the results of our implemented models. We finally conclude the report in Section 5.

2 Description Of Data Set

The data for the given personalization problem is stored in the form of data logs. Each log represents a stream of user actions, with each line representing a session meta data, a query action, or a click action. Each line contains tab-separated values according to the following format:

```
SessionID TypeOfRecord Day USERID
SessionID TimePassed TypeOfRecord SERPID QueryID ListOfTerms
ListOfURLsAndDomains
SessionID TimePassed TypeOfRecord SERPID URLID
```

- $SessionID$ is the unique identifier of a search session. Day is the number of the day in the data (the entire log spans over 30 days).
- $TypeOfRecord$ is the type of the log record. Its either a query (Q, T), a click (C), or the session metadata (M). T letter is used only for test queries.
- $UserID$ is the unique identifier of a user.
- $TimePassed$ is the time passed since the start of the session with the SessionID in units of time. We do not disclose how many milliseconds are in one unit of time.
- $QueryID$ is the unique identifier of a query.

  Query records labelled by TypeOfRecord = T are test queries. The personalised ranking for these queries should be submitted as described in the Evaluation section. For convenience, we put the sessions with test queries in a separate file.
- $ListOfTerms$ is a comma-separated list of terms of the query, represented by their TermIDs.
- $SERPID$ is the unique identifier of a search engine result page at the session level (SERP).
- $TermId$ is the unique identifier of a query term.
- $URLID$ is the unique identifier of an URL.
- $ListOfURLsAndDomains$ is the list of comma-separated pairs of URLID and the corresponding DomainId. It is tab-separated and ordered from left to right as they were shown to the user from the top to the bottom.

These records describe the session of a user, on a particular the day. The user submits a query, which has a set of terms. The search engine receives a query and provides ten URLs along with their domains as SERP. The user clicks and time at which a URL was clicked is

recorded. The initial size of the database was 16GB, but the data was highly unstructured for further use. In order to make it easy to query on the data, we organised the data into query-able databsae, which increased the size of the database to 20GB. The final dataset used for training has web-surfing data of about 1000 users for 24 days, and testing dataset has the web-surfing data of same users for 25th to 27th day.

3 Analysis Of Data Set and Approach

The Yandex database provides information about the every session of a user such as the queries in a particular session, the SERP Urls that the search engine returned for a given query, the Urls that the user clicked and the time instant at which the click happened. The two important requirements for user personalisation are models to estimate each users interest every topic and ranking of all pages for every topic.

In order to obtain a reasonable estimate of users' interests in various topics, we first build a User-Url matrix, which captures each users interest in every page on the web. The User-Url matrix, U is has elements $u_{i,j}$, which are computed as the $Cumulative - Relevance(i,j)$. The $Cumulative - Relevance(i,j)$ of page j for user i is proportional to the summation of relevances of the page for the user at every user-page encounter.

$$U_{i,j} = Cumulative - Relevance(i,j) =$$

$$\frac{\Sigma_t Relevance_t(i,j)}{\Sigma_j \Sigma_t Relevance_t(i,j)}$$

,

The $Relevance_t(i,j)$ of page j for user i at a particular encounter is calculates in the following way:

- If the page p is not clicked by the user, then $Relevance_t(i,j) = 0$.
- If the page p is clicked by the user and dwell-time of less than 50 time units, then $Relevanc_t(i,j) = 1$.
- If the page p is clicked by the user and dwell-time between 50 to 400 time units, then $Relevanc_t(i,j) = 2$.
- If the page p is clicked by the user and dwell-time is at least 400 time units, then $Relevanc_t(i,j) = 3$.
- In addition, the relevance of the documents associated with clicks which are the last actions in the corresponding sessions are also 3.

Once the User-Url matrix U is estimated in this way, we try to decompose this matrix into User-Topic Matrix T and Topic-Url Matrix L. The User-Topic matrix models the each

user's interest in every topic, and The Topic-Url matrix models a Topic's relevance for a particular web page.

The decomposition technique we have used for getting T and L from U is Non-negative matrix factorisation technique. This method ensures that even after decomposing the matrix U, the two factor matrices do not have any negative valued elements. This is necessary because the user-topic interest values and topic-page relevance values cannot be negative.

$$U_{m,n} = T_{m,l} \times L_{l,n}, \text{ where m = \# of users, l = \# of topics, n = \# of pages.}$$

In order to get the user's topic interest vector we just have to pick the row of the matrix T corresponding to user u.

So, in order to calculate the Topic-sensitive Page Ranks, the remaining part is to find the Topic-relevance of a given topic 't', for a page p. As, the Topic-Url matrix L doesn't account for the incoming and outgoing links of a page p and any network-related information, it cannot be trusted for estimating the usefulness of page p, for a user for a given topic.
In order to obtain the relevance of every topic for a page which includes the consideration of network strength, we need to get Topic-Url relevance from Query-Url matrix, instead of User-Url matrix. Thus, we first build a Query-Url matrix Q, which models the relevance of each url for every query.

$$Q_{i,j} = \text{\# of times url } j \text{ has occurred in the top SERP of query } i.$$

Now, we factorize the Query-Url matrix, Q into Query-Topic matrix p and Topic-Url matrix R. Here again we use Non-negative matrix factorization to find both P and R. The Topic-Url matrix R accounts for both relevance of a page for a topic and the network strength of a page. And, the finally $TSPR(u, p)$, Topic-Sensitive Page Rank of a page p for a user u is computed by,

$$TSPR(u, p) = \Sigma_i T_u(i).R_p(i),$$

where, $T_u(i)$ is the interest of a user u in topic i and $R_p(i)$ is the importance(relevance + network strength) of a page p for topic i.

The TSPR is calculated for the urls in a SERP and they are re-ranked only if the the the query turns out to be a non-navigational queries. A navigational query is a query for which most of the times the Url clicked is the same irrespective of the kind of user. The examples of such queries are Facebook Login, Amazon, etc. Such query results are better off untouched.

4 Results

4.1 Evaluation Scheme

The evaluation scheme used here is called the Normalized Discounted Cumulative Gain(NDCG). It is a measure of ranking quality. In information retrieval, it is often used to measure effectiveness of web search engine algorithms or related applications. It measures the usefulness,

or gain, of a document based on its position in the result list. The premise of DCG is that highly relevant documents appearing lower in a search result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result.The following formulation of DCG places stronger emphasis on retrieving relevant documents:

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

For every pair of substantially different ranking functions, the NDCG can decide which one is better in a consistent manner.

$$NDCG_p = \frac{DGC_p}{IDCG_p}$$

Comparing a search engine's performance from one query to the next cannot be consistently achieved using DCG alone, so the cumulative gain at each position for a chosen value of p should be normalized across queries. This is done by sorting documents of a result list by relevance, producing the maximum possible DCG till position p, also called Ideal DCG (IDCG) till that position.

For example let the search engine return P_1, P_2, P_3 for a query, and the user give these pages relance scores as 0, 3, 2 respectively. Now CG = 0 + 3 + 2 = 5. DCG = $1/1 + 8/log_2(3) + 4/2 = 3 + 8/log_2(3)$.
IDCG = $8 + 4/log_2(3) + 1/2$. Therefore, NDCG = $\frac{3+8/log_2(3)}{8+4/log_2(3)+1/2}$.

4.2 Performance Results

The initial database included the 16GB of web-surfing data of the following kind.

- Unique queries: 21,073,569
- Unique urls: 703,484,26
- Unique users: 5,736,333
- Training sessions: 34,573,630
- Test sessions: 797,867
- Clicks in the training data: 64,693,054
- Total records in the log: 167,413,039

Organising the data into query-able data-base took the size of the data to 20GB. In a short while we found from successive attempts that the time required for running programs for such huge database could take painfully long. Thus, we decided to reduce our data set, in a way that will have not affect on the correctness of the data or the algorithm used.
The final data set used for training has web-surfing data of about 1000 users for 24 days,

and testing data set has the web-surfing data of same users for 25th to 27th day. Reducing the number of users we choose to personalize web-searches for mihgt affect the results of personalisation for a particular person, as each user's personalization needs not only his web-surfing information, but the query-Url relevance as well, irrespective of the user.

Using the NDCG evaluation measures, the testing on the reduced 3-day test data set gave an accuracy(average NDCG measure) of 0.65059178, where as the winning algorithm's accuracy was 0.8075.

5 Conclusion

The size of data we used for the model is much smaller than the entire data set. The primary challenge that we faced was the huge size of the data set. Since the web-surfing history was almost 20 GB, it early on we realised it would take really long to run as our algorithm required to factorise a matrix of order 5 million X 700 million. As this was extremely slow and time consuming, we decided to cut down our number of users to 1000. This is still complex and time consuming considering the step of matrix factorisation involved. The accuracy of the winning solution is 0.8075, which is higher than the accuracy that we were able to achieve which was 0.65059178. The reason for which we think this gap in accuracy exists is mainly because of the reduction of the data set. The reduction of the data set results in loss of important in formation captured in the query-URl matrix, which is a significant part of parameter in our algorithm.

6 Acknowledgements

We would like to thank Prof. G. S. Raghavan sir for his help, support and encouragement throughout the course of the project.

7 References

1. Qiu, Feng, and Junghoo Cho. "Automatic identification of user interest for personalized search." Proceedings of the 15th international conference on World Wide Web. ACM, 2006.

2. Dou, Zhicheng, Ruihua Song, and Ji-Rong Wen. "A large-scale evaluation and analysis of personalized search strategies." Proceedings of the 16th international conference on World Wide Web. ACM, 2007.

3. Teevan, Jaime, Susan T. Dumais, and Daniel J. Liebling. "To personalize or not to personalize: modeling queries with variation in user intent." Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008.

4. Teevan, Jaime, Daniel J. Liebling, and Gayathri Ravichandran Geetha. "Understanding and predicting personal navigation." Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011.

5. Sontag, David, et al. "Probabilistic models for personalizing web search." Proceedings of the fifth ACM international conference on Web search and data mining. ACM, 2012.

6. Bennett, Paul N., et al. "Modeling the impact of short-and long-term behavior on search personalization." Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2012.

7. Eickhoff, Carsten, et al. "Personalizing atypical web search sessions." Proceedings of the sixth ACM international conference on Web search and data mining. ACM, 2013.

8. Shokouhi, Milad, et al. "Fighting search engine amnesia: Reranking repeated results." Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013.

9. Wang, Hongning, et al. "Personalized ranking model adaptation for web search." Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013.

10. Song, Guocong. Point-Wise Approach for Yandex Personalized Web Search Challenge. Technical report, IEEE. org, 2014.

11. Masurel, Paul, et al. Dataikus Solution to Yandexs Personalized Web Search Challenge. Technical report, Dataiku, 2014.

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