JMLR: Workshop and Conference Proceedings 11 (2010) 152–159 Workshop on Applications of Pattern Analysis

Learning to Rank for Personalized News Article Retrieval

Lorand Dali LORAND.DALI@IJS.SI
Blaž Fortuna BLAZ.FORTUNA@IJS.SI
Jan Rupnik JAN.RUPNIK@IJS.SI
Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

Editors: Tom Diethe, Nello Cristianini, John Shawe-Taylor

Abstract

This paper aims to tackle the very interesting and important problem of user personalized ranking of search results. The focus is on news retrieval and the data from which the ranking model is learned was provided by a large online newspaper. The personalized news search ranking model which we have developed takes into account not only document content and metadata, but also data specific to the user such as age, gender, job, income, city, country etc. All the user specific data is provided by the user himself when registering to the news site.
Keywords: Information retrieval, learning to rank, personalization

1. Introduction

One of the most widely used application on the internet is search. While there are many types of search we shall focus only on news search in this paper (i.e retrieval of a ranked list of news aricles as a result to a keyword query). Most of the news search engines rank the search results based on the relevance of the content of the article to the query, and based on the date when the article was written, preferring newer articles. However this method gives as a response to a query the same search results to every user. It makes sense to try to meet the needs of different users by personalizing the search results. Such results are now possible because large news sites have vast amounts of data, both demographic and behavioral, about their users. In this paper we develop and test a personalized ranking model based on click through logs of a large news site. Furthermore we investigate how much the user's demographic data determines what kind of news articles that user reads.

Section 2 briefly talks about the related work, Section 3 describes the data, Section 4 is about the learning algorithms, in Section 5 we describe the experiments, and finally we draw the conclusions.

2. Related Work

Personalization is an important topic, and much research has been done in this area. The need to provide personalized access to news, either by means of personalized search (Sanderson and Rijsbergen, 1991) or through a personalized online newspaper (Kamba et al., 1995), has been recognized in the early days of internet already. To provide the overview of the related work as understandable as possible, we shall describe the related work along four dimensions, as proposed in a the survey about personalization on the web (Pretschner and Gauch, 1999a). The four dimensions along which the related work will be looked at are: application, user data, rating/filtering algorithm, and filtering type.

Application shows in which practical application the personalization is used. We can have contextual retrieval (Zemirli et al., 2007; Budzik and Hammond, 2000), recommendation (Mobasher et al., 2000), more specifically news recommendation (Claypool et al., 1999; Das et al., 2007), personalized search in general (Liu et al., 2004; Zemirli et al., 2007; Vallet et al., 2007; Pretschner and Gauch, 1999b), personalized news search (Sanderson and Rijsbergen, 1991), and personalized online newspapers (Kamba et al., 1995; Claypool et al., 1999). The application of our system is personalized news search.

Secondly, we can organize the related work according to what user data is used for personalization and how the user profiles are constructed. The user can share data about himself either explicitly or implicitly. Explicit data about the user could be a list of keywords about his interests, which he specifies himself. Such an approach is taken in (Kamba et al., 1995) and in (Zemirli et al., 2007). Another type of explicit information is used in (Sanderson and Rijsbergen, 1991), where the user gives weights to the different keywords occurring in his query. Implicit information about the user can be obtained from his browsing history. The browsing history can be either the textual content of the pages the user has visited, or just the set of pages which he has visited, without the contents. The work which analyses the textual content of the browsing history can be further divided into two categories based on how the user profile is constructed: some work extracts keywords (Kamba et al., 1995; Liu et al., 2004; Claypool et al., 1999) from the history to build the user profile, while other work builds a concept hierarchy or ontology (Pretschner and Gauch, 1999b) as the user profile. Another source of implicit information about the user is relevance feedback, which can be binary (Kamba et al., 1995) or rating on a scale (Claypool et al., 1999). One more interesting source of implicit information is described in (Budzik and Hammond, 2000), where the authors take into account the user's interactions with the applications (word processors, browsers, etc.) he is using. What makes our work different from the work mentioned above is that we use demographic user data (like age, gender, city, country, job, industry, income, etc.). This data is explicitly given by the user. The nature of our data enables us to explore how demographic attributes can be correlated with the search experience, and which attributes are most correlatable. We can also detect subgroups of users whose search experience is most improved by personalization. Moreover, in our analysis of logs, we don't focus on the articles which the user has been reading so we could learn his profile; instead we focus on the searches the user has made, and more importantly the search results he has clicked. From this we can infer (Joachims, 2002) which one of two articles is more relevant to a specific user who has made a specific query. Additionally, we could also look at the related work from the perspective of the rating/filtering algorithm which is used to match the user profile to an document in order to see how relevant that document is. Possible solutions can be automatic and semiautomatic. In the semiautomatic case, in (Sanderson and Rijsbergen, 1991) the user has to manually adjust for each query the weights of the words in his query; in (Liu et al., 2004) the user has to choose from categories he is interested in from DMOZ, for each query. Automatic approaches include: representing both the user profile and the document in the vector space model and then using similarity measures (Kamba et al., 1995; Claypool et al., 1999), using influence diagrams (Zemirli et al., 2007), analyzing the browsing history with association rule mining (Mobasher et al., 2000) or covisitation counts (Das et al., 2007), determining similar users bi MinHash clustering (Das et al., 2007) or by the Pearson correlation coefficient (Claypool et al., 1999). While talking about the rating/filtering algorithm, we have to mention that there are three ways of dealing with personalization: re-ranking, filtering and query expansion (Pretschner and Gauch, 1999b). While most of the work mentioned above falls into the filtering approach, our methods do re-ranking of search results to fit the user's needs. The re-ranking is done by predicting a score for each article and then sorting the articles by their scores. For obtaining the models which predict the relevance score for a document we use learning to rank methods like the ranking perceptron (Elsas et al., 2008) and rank SVM (Joachims, 1999).

Finally, according to the filter type, the approaches can be collaborative (Mobasher et al., 2000; Claypool et al., 1999; Das et al., 2007) or content based (Kamba et al., 1995; Sanderson and Rijsbergen, 1991; Liu et al., 2004; Zemirli et al., 2007; Pretschner and Gauch, 1999b; Claypool et al., 1999; Budzik and Hammond, 2000). In the content based approach the documents are filtered based on the text they contain, the ones which contain text similar to the user's profile are favored. In the collaborative filtering approach the contents of the documents are not important. More important is that the documents rated high by users with similar profiles to the given user are favored. Filtering is not required to be purely collaborative or content based, (Claypool et al., 1999) for example presents a way to combine the two in order to leverage the advantages from both. In spite of our approach not being a filtering (but re-ranking) approach, we use both content as well as information gained from analyzing the search results picked by similar users.

In conclusion, the contributions of our paper are the following: we develop a personalized news search, we use explicit demographic data about the user and pairwise article preference information inferred from click logs to learn a ranking model for re-ranking search results. Our re-ranking takes into account both content based and collaborative in nature. We also analyze the trained ranking models to understand which of the user-related attributes significantly influence the ranking.

3. Description of the Data

Our approach to personalized news search is re-ranking of search results to fit the user's preferences. The ranking model which does the re-ranking of the search results is obtained using learning to rank methods which we will describe later. Due to the data driven approach of our methods we consider it helpful to describe the data in this section before going into details about learning to rank algorithms and experiments in the next sections.

The data comes from search logs of a large online newspaper. Each search made is recorded by the system. Every recorded search contains the following information: the user who made the search, the query which was asked, the time when the search was made, the search results obtained, and the search result clicked by the user. Knowing which search result the user has clicked, we can assume that the article represented by the clicked search result is more relevant than every article which has a higher rank in the search results. We cannot assume anything about the articles below the clicked one. If the user clicks on the first article in the search results, then we can assume that the first article is more relevant than the second article, but we cannot say anything about the other ones. This way of deriving pairwise relevance from search logs is described in detail in (Joachims, 2002). Therefore we can conclude two things: one is that we will try to estimate relevance of an article to a given query made by a specific user, and the other is that we will use a pairwise setting for learning to rank. Thus we consider a pair to be the following:

$$\langle (u, q, d_R), (u, q, d_N) \rangle \tag{1}$$

where d_R is an article which is more relevant than the article d_N for the query q made by the user u. Thus to provide training data for the learning to rank algorithms we have to extract such a list of pairs from the search logs. We represent (u, q, d) as a sparse vector.

4. Learning Algorithms

In this section we shall present two linear learning algorithms, the Perceptron and SVM, which we have used to learn a ranking model. We recall from section 3 that the training data comes in the form:

$$\{(x_R^i, x_N^i)\}_{1 \leq i \leq n}, x_R^i, x_N^i \in R^P \tag{2}$$

where:

$$x := (u, q, d) \tag{3}$$

Moreover, x also contains quadratic features of the vector (u, q, d) by extending it with entries containing pairs in the initial vector. For example the pair (gender = female, "Canada" in locations mentioned) would be a feature in the new vector. This being said we can start detailing the linear learning methods which we will use. The goal is to learn a weight vector $w \in R^P$, of the same dimensions as the training vectors x. Then given a new vector y we can compute the score of this vector, which is equal to the inner product between the weight vector w and the vector y.

$$score = w \cdot y \tag{4}$$

The ranking then consists in ordering articles by their scores decreasingly.

In the following subsections we shall describe two linear methods of obtaining the weight vector w and a method for extracting most informative features from the model.

4.1. Ranking Perceptron

The Perceptron (Rosenblatt, 1958) algorithm was initially designed for binary classification, but we can easily adapt it to the pairwise ranking problem as described in (Elsas et al., 2008).

The general idea is that we start with a randomly initialized vector w, and successively adjust it during T iterations if it has failed in giving x_R a higher score than x_N. The algorithm is described in the pseudocode below: η is called learning rate and is a real number chosen between 0 and 1. The higher η the faster the perceptron learns. On the other hand perceptrons with high learning rate are more sensitive to noise.

$w^0 \leftarrow random$
for $i = 0$ to T **do**
  for all (x_R, x_N) **do**
    if $w^i . x_R < w^i . x_N$ **then**
      $w^{i+1} \leftarrow w^i + \eta(x_R - x_N)$
    end if
  end for
end for

4.2. Rank SVM

Linear SVM (Joachims, 1999) is another popular way of learning the weight vector w. Originally SVM is formulated as a binary classification problem, where w is the separating hyperplane with maximum margin. The linear soft margin SVM for classification can be adapted to the pairwise ranking problem (Joachims, 2002). The objective is to make the inner product $w \cdot x_R$ be greater than $w \cdot x_N$ by the margin 1 and allowing for some errors ξ. We have:

$$w \cdot (x_R^i - x_N^i) \geq 1 - \xi_i, \forall i \tag{5}$$

We recall that the maximum margin separating hyperplane is the one which minimizes:

$$\frac{1}{2}\|w\|^2 + C \sum \xi_i \tag{6}$$

This is called the primal problem, and it is the one which we shall solve as described in (Rupnik, 2008). By substituting ξ_i we get the hinge loss

$$\frac{1}{2}\|w\|^2 + C \sum (1 - w \cdot (x_R^i - x_N^i))_+ \tag{7}$$

Where the function $(\cdot)_+$ is defined as $(\cdot)_+ := max(0, \cdot)$. We minimize the hinge loss by using the subgradient method, which gives an approximate solution but converges very fast.

4.3. Feature Selection

An important question to which we are interested in finding the answer, is which features play a more important role in computing the score of the vector for an article to be ranked. Because of the way how we compute this score as an inner product between the weight vector w and the vector x, we could say that those features matter most which have the highest weights in absolute value in the corresponding positions of w. However, averaging the elements on each position in the inner product turns out to be more stable (Brank et al., 2002).

$$w \cdot x^i = w[0] \cdot x^i[0] + \cdots + w[P] \cdot x^i[P], \forall 1 \leq i \leq M \tag{8}$$

$$weight[k] = \frac{1}{M} \sum_{i=1}^{M} w[k] \cdot x^i[k] \tag{9}$$

This way we can say that the features k with greatest $|weight[k]|$ contribute most to ranking a document high if $weight[k] > 0$, or ranking a document low if $weight[k] < 0$.

5. Experiments

In this section we focus on the following questions: can we improve the ranking by training a ranking model based on user preferences and demographic data, how well can we predict the user's demographic data based on the content of the news articles the user reads.

For training a ranking model we analyzed all together 380,000 searches performed by 326,000 unique users over a period of two weeks. The presented search results covered 134,000 unique articles and other content pages from the news website. In Section 3 we described how this data was transformed into a list of pairs for the learning to rank algorithms. This resulted in 750,000 pairs out of which chronologically first 70% were used for training and the rest for evaluation. In the evaluation we compared the personalized ranking model with purely content based ranking models such as BM25 and the vector space model. We tried two evaluation methods: one to determine the accuracy we can achieve in correctly ordering the pairs in the test set, and the other which determines on which positions our ranking model placed the relevant documents (the ones clicked by the user). However, our experiments did not confirm our hipothesis that user demographic data would help in better ranking of the search results. We did not observe significant differences in the performance of the tree ranking models we tested (vector space, BM25, personalized).
As one of the reasons why the personalized ranking failed, we identified that the click through data is highly biased by the position in which the clicked article was originally displayed. Indeed, more than 50% of user clicks are on the first search result, 20% are on the second, 10% on the third, and less than 3% of the users ever click on the 10th result or lower. This can mean that either the ranking of the news website is very good, or that users tend to click on the results displayed on higher positions regardless of how relevant they actually are. If the latter is true, this would cause us to deduce faulty relevance feedback which would 'confuse'the learning algorithm.

Another possible reason for the lack of success of personalization could be if the contents of articles the user reads is not highly dependent on his demographic data. To see to what extent demographic data determines what articles the user reads, we have tried to predict user attributes based only on the features of the articles the user visits. We regard this as a multiclass classification problem where the users are data points represented by the centroid of the the feature vectors of the visited articles. The attributes we try to predict (e.g gender) are the target variables and each possible value is a class (e.g for gender the possible classes are male and female). For the classification we have used SVM and 5-fold cross validation. The results are shown in Figure 1. The evaluation measure is Break Even Point (BEP) - a hypothetical point at which precision (ratio of positive documents among retrieved ones) and recall (ratio of retrieved positive documents among all positive documents) are the same. We observe that it is easiest to predict gender, while it's almost impossible to predict income reliably. For age, the middle age groups are very hard to predict, while the youngest and the oldest age group is somewhat easier.

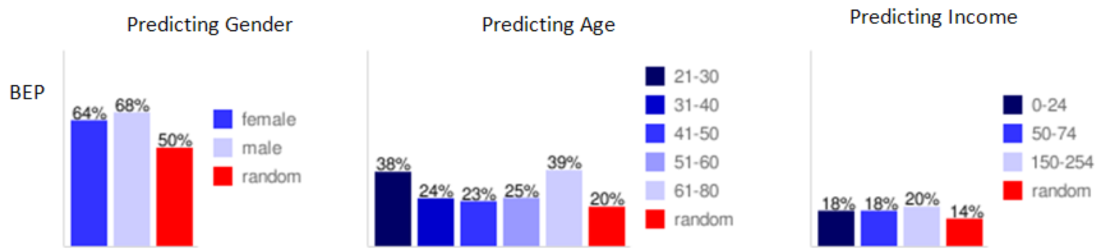

Figure 1: Predictions of user's gender, age and income.

6. Conclusions

In conclusion, we have presented a ranking model which takes into account not only features about document content and metadata, but also demographic features of the user who performs the search. The demographic data (gender, job industry, income, age, city, country) is given by the user upon registration. The pairwise relevance feedback is inferred from click trough logs which were provided by an important news site. We could not experimentally confirm the intuition that knowledge of the user's demographic data would result in obtaining a better ranking model.

References

T. Kamba, and K. Bharat and Albers. The Krakatoa Chronicle-An interactive, personalized, newspaper on the web. In Proceedings of the Fourth International World Wide Web Conference (Boston, Massachusetts, USA, December 11-14, 1995) , 11-14.

Sanderson, M. and Rijsbergen, C.J. 1991. NRT: news retrieval tool. Electronic Publishing Vol. 4 (December 1991), 205-217

Pretschner, A. and Gauch, S. 1999. Personalization on the Web. Technical Report ITTC-FY2000-TR-13591-01 (December 02 - 1999).

Liu, F. and Yu, C. and Meng, W. 2004. Personalized web search for improving retrieval effectiveness. Journal of IEEE Transactions on knowledge and data engineering, 2004, 28-40.

Zemirli, W.N. and Tamine-Lechani, L. and Boughanem, M. 2007. A personalized retrieval model based on influence diagrams. Sixth International and Interdisciplinary Conference on Modeling and Using Context (Roskilde University, Denmark, 20-24 August 2007)

Mobasher, B. and Cooley, R. and Srivastava, J. 2000. Automatic personalization based on Web usage mining. ACM New York, NY, USA, 2000.

Vallet, D. and Fernï£¡ndez, M. and Castells, P. and Mylonas, P. and Avrithis, Y. 2007. Personalized information retrieval in context. 3rd Int. Workshop Modeling Retrieval Context 21st Nat. Conf. Artif. Intell. (Boston, MA, 2007).

Pretschner, A. and Gauch, S. 1999. Ontology based personalized search. In Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (Chicago, Illinois, USA, 8-10 November 1999), 391-398

Claypool, M. and Gokhale, A. and Miranda, T. and Murnikov, P. and Netes, D. and Sartin, M. 1999. Combining content-based and collaborative filters in an online newspaper. In Proceedings of ACM SIGIR Workshop on Recommender Systems (Berkeley, CA, USA, 15-19 August, 1999).

Das, A.S. and Datar, M. and Garg, A. and Rajaram, S. 2007. Google news personalization: scalable online collaborative filtering. In Proceedings of the 16th international conference on World Wide Web (Banff, Alberta, Canada, May 8-12, 2007). 280

Budzik, J. and Hammond, K.J. 2000. User interactions with everyday applications as context for just-in-time information access. In Proceedings of the 5th international conference on Intelligent user interfaces, 2000, 44-51.

Salton, G. Developments in Automatic Text Retrieval. Science, Vol 253, 974-979, 1991.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC 1994). Gaithersburg, USA, November 1994.

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological review, 386 - 408.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 133 - 142.

Elsas, J.L. and Carvalho, V.R. and Carbonell, J.G. 2008. Fast learning of document ranking functions with the committee perceptron. In Proceedings of the international conference on Web search and web data mining, 55 - 64.

T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

Rupnik, J. 2008. Stochastic subgradient approach for solving linear support vector machines - an overview. SiKDD, 2008.

Brank J., Grobelnik M., Milic-Frayling N., Mladenic D. Feature selection using support vector machines. Proc. of the Third International Conference on Data Mining Methods and Databases for Engineering, Finance, and Other Fields, 2002.