Where to go on your next trip? Optimizing travel destinations based on user preferences


DOI
10.1145/2766462.2776777

Publication date
2015

Document Version
Final published version

Published in
SIGIR 2015

License
Article 25fa Dutch Copyright Act (https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care)

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Download date:20 May 2024
Where to Go on Your Next Trip?
Optimizing Travel Destinations Based on User Preferences

Julia Kiseleva1  Melanie J.I. Mueller2  Lucas Bernardi2
Chad Davis2  Ivan Kovacek2  Mats Stafsend Einarsen2
Jaap Kamps3  Alexander Tuzhilin4  Djord Hiemstra5

1Eindhoven University of Technology, Eindhoven, The Netherlands
2Booking.com, Amsterdam, The Netherlands
3University of Amsterdam, Amsterdam, The Netherlands
4Stern School of Business, New York University, New York, USA
5University of Twente, Enschede, The Netherlands

1j.kiseleva@tue.nl  3kamps@uva.nl  4atuzhilin@stern.nyu.edu  5d.hiemstra@utwente.nl
2{melanie.mueller, lucas.bernardi, chad.davis, ivan.kovacek, mats.einarsen}@booking.com

ABSTRACT
Recommendation based on user preferences is a common task for e-commerce websites. New recommendation algorithms are often evaluated by offline comparison to baseline algorithms such as recommending random or the most popular items. Here, we investigate how these algorithms themselves perform and compare to the operational production system in large scale online experiments in a real-world application. Specifically, we focus on recommending travel destinations at Booking.com, a major online travel site, to users searching for their preferred vacation activities. To build ranking models we use multi-criteria rating data provided by previous users after their stay at a destination. We implement three methods and compare them to the current baseline in large scale online experiments in a real-world application. The destination finder service returns a ranked list of recommended destinations.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGIR ’15, August 09 - 13, 2015, Santiago, Chile.
© 2015 ACM, ISBN 978-1-4503-3621-5/15/08...
DOI http://dx.doi.org/10.1145/2766462.2776777
http://www.booking.com/destinationfinder.html

1097
stand user interests and predict future actions \[3,4\]. Such models work by projecting users and items into a lower-dimensional space, thereby grouping similar users and items together, and subsequently computing similarities between them. This approach can run into data sparsity problems, and into a continuous cold start problem when new items continuously appear.

In multi-criteria RS \[1,2,12\] the rating function has the following form:

\[
F : U \times I \rightarrow (r_0 \times r_1 \cdots \times r_n)
\]

(2)

The overall rating \(r_0\) for an item shows how well the user likes this item, while criteria ratings \(r_1, \ldots, r_n\) provide more insight and explain which aspects of the item she likes. MCRS predict the overall rating for an item based on past ratings, using both overall and individual criteria ratings, and recommends to users the item with the best overall score. According to \[1\], there are two basic approaches to compute the final rating prediction in the case when the overall rating is known. In our work we consider a new type of input for RS which is multi-criteria ranking data without an overall rating.

There are a number of important challenges in working on the real world application of travel recommendations.

First, it is not easy to apply RS methods in large scale industrial applications. A large scale application of an unsupervised RS is presented in \[9\], where the authors apply topic modeling techniques to discover user preferences for items in an online store. They apply Locality Sensitive Hashing techniques to overcome performance issues when computing recommendations. We should take into account the fact that if it’s not fast it isn’t working. Due to the volume of traffic, offline processing—done once for all users—comes at marginal costs, but online processing—done separately for each user—can be excessively expensive. Clearly, response times have to be sub-second, but even doubling the CPU or memory footprint comes at massive costs.

Second, there is a continuous cold start problem. A large fraction of users has no prior interactions, making it impossible to use collaborative recommendation, or rely on history for recommendations. Moreover, for travel sites, even the more active users visit only a few times a year and have volatile needs or different personas (e.g., business and leisure trips), making their personal history a noisy signal at best.

To summarize, our problem setup is the following: (1) we have a set geographical destinations such as ‘Paris’, ‘London’, ‘Amsterdam’ etc.; and (2) each destination was reviewed by users who visited the destination using a set of endorsements. Our main goal is to increase user engagement with the travel recommendations as indicator of their interest in the suggested destinations.

Our main research question is: How to exploit multi-criteria rating data to rank travel destination recommendations? Our main contributions are:

- we use multi-criteria rating data to rank a list of travel destinations;
- we set up a large-scale online A/B testing evaluation with live traffic to test our methods;
- we compared three different rankings against the industrial baseline and obtained a significant gain in user engagement in terms of conversion rates.

The remainder of the paper is organized as follows. In Section 2, we introduce our strategies to rank destinations
recommendations. We present the results of our large-scale online A/B testing in Section 3. Finally, Section 4 concludes our work in this paper and highlights a few future directions.

2. RANKING DESTINATION RECOMMENDATIONS

In this section, we present our ranking approaches for recommendations of travel destinations. We first discuss our baseline, which is the current production system of the destination finder at Booking.com. Then, we discuss our first two approaches, which are relatively straightforward and mainly used for comparison: the random ranking of destinations (Section 2.2), and the list of the most popular destinations (Section 2.3). Finally, we will discuss a Naive Bayes ranking approach to exploit the multi-criteria ranking data.

2.1 Booking.com Baseline

We use the currently live ranking method at Booking.com's destination finder as a main baseline. We are not able to disclose the details, but the baseline is an optimized machine learning approach, using the same endorsement data plus some extra features not available to our other approaches.

We refer further to this method as ‘Baseline’.

Next, we present two widely employed baselines, which we use to give an impression how the baseline performs. Then we introduce an application of the Naive Bayes ranking approach to multi-criteria ranking.

2.2 Random Destination ranking

We retrieve all destinations that are endorsed at least for one of the activities that the user is searching for. The retrieved list of destinations is randomly permuted and is shown to users.

We refer further to this method as ‘Random’.

2.3 Most Popular Destinations

A very straightforward and at the same time very strong baseline would be the method that shows to users the most popular destinations based on their preferences [6]. For example, if the user searches for the activity ‘Beach’, we calculate the popularity rank score for a destination d as the conditional probability: \( P(\text{Beach}|d) \). If the user searches for a second endorsement, e.g. ‘Food’, the ranking score for \( d \) is calculated using a Naive Bayes assumption as: \( P(\text{Beach}|d) \times P(\text{Food}|d) \). In general, if the user provides \( n \) endorsements, \( e_1, \ldots, e_n \), the ranking score for \( d \) is \( P(e_1|d) \times \ldots \times P(e_n|d) \).

We refer further to this method as ‘Popularity’.

2.4 Naive Bayes Ranking Approach

As a primary ranking technique we use a Naive Bayes approach. We will describe its application to the multi-criteria ranking data (presented in Equation [4]) with an example. Let us again consider a user searching for ‘Beach’. We need to return a ranked list of destinations. For instance, the ranking score for the destination ‘Miami’ is calculated as

\[
P(\text{Miami}, \text{Beach}) = P(\text{Miami}) \times P(\text{Beach}|\text{Miami}),
\]

where \( P(\text{Beach}|\text{Miami}) \) is the probability that the destination Miami gets the endorsement ‘Beach’. \( P(\text{Miami}) \) describes our prior knowledge about Miami. In the simplest case this prior is the ratio of the number of endorsements for Miami to the total number of endorsements in our database.

If a user uses searches for a second activity, e.g. ‘Food’, the ranking score is calculated in the following way:

\[
P(\text{Miami}, \text{Beach}, \text{Food}) = P(\text{Miami}) \times P(\text{Beach}|\text{Miami}) \times P(\text{Food}|\text{Miami})
\]

If our user provides \( n \) endorsements, Equation [4] becomes a standard Naive Bayes formula.

We refer further to this method as ‘Naive Bayes’.

To summarize, we described three strategies to rank travel destination recommendations: the random ranking, the popularity based ranking, and the Naive Bayes approach. These three approaches will be compared to each other and against the industrial baseline. Next, we will present our experimental pipeline which involves online A/B testing at the destination finder service of Booking.com.

3. EXPERIMENTS AND RESULTS

In this section we will describe our experimental setup and evaluation approach, and the results of the experiments. We perform experiments on users of Booking.com where an instance of the destination finder is running in order to conduct an online evaluation. First, we will detail our online evaluation approach and used evaluation measures. Second, we will detail the experimental results.

3.1 Research Methodology

We take advantage of a production A/B testing environment at Booking.com which performs randomized controlled trials for the purpose of inferring causality. A/B testing randomly splits users to see either the baseline or the new variant version of the website, which allows to measure the impact of the new version directly on real users [10, 11, 12].

As our primary evaluation metric in the A/B test, we use conversion rate, which is the fraction of sessions which end with at least one clicked result [13]. As explained in the motivation, we are dealing with an exploratory task and therefore aim to increase customer engagement. An increase in conversion rate is a signal that users click on the suggested destinations and thus interact with the system.

In order to determine whether a change in conversion rate is a random statistical fluctuation or a statistically significant change, we use the G-test statistic (G-tests of goodness-of-fit). We consider the difference between the baseline and the newly proposed method significant when the G-test p-value is larger than 90%.

3.2 Results

Conversion rate is the probability for a user to click at least once, which is a common metric for user engagement. We used it as a primary evaluation metric in our experimentation. Table 1 shows the results of our A/B test. The production ‘Baseline’ substantially outperforms the ‘Random’ ranking with respect to conversion rate, and performs slightly (but not significantly) better than the ‘Popularity’ approach. The ‘Naive Bayes’ ranker significantly increases the conversion rate by 4.4% compared to the production baseline.

We achieved this substantial increase in conversion rate with a straightforward Naive Bayes ranker. Moreover, most computations can be done offline. Thus, our model could be
Table 1: Results of the destination finder online A/B testing based on the number of unique users and clickers.

<table>
<thead>
<tr>
<th>Ranker type</th>
<th>Number of users</th>
<th>Conversion rate</th>
<th>G-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>9.928</td>
<td>25.61% ± 0.72%</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>10.079</td>
<td>24.46% ± 0.71%</td>
<td>94%</td>
</tr>
<tr>
<td>Popularity</td>
<td>9.838</td>
<td>25.50% ± 0.73%</td>
<td>41%</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>9.895</td>
<td><strong>26.73% ± 0.73%</strong></td>
<td>93%</td>
</tr>
</tbody>
</table>

This research has been partly supported by STW and is part of the CAPA project.

References


Acknowledgments

This work was done while the main author was an intern at Booking.com. We thank Lukas Vermeer and Athanasios Noulas for fruitful discussions at the early stage of this work.

http://www.win.tue.nl/~mpechen/projects/capa/