Code Suisse 2018 - Dynamic Locus Display Design Document

Sebastian Coates, Prabhath Kotha

May 24, 2018

Problem Description

User experience optimization is a multifaceted issue. It involves minimizing effort on the user's end, timely and reliable rendering, and ease of use/adoption. For this event's purpose, we are focusing on effort minimization in the use-case of an intranet-browsing user. Specifically, we aim to decrease click count and time spent.

Our solution will accomplish this by shifting the majority of searching and ordered browsing work to the back-end. We will provide the user with an intelligently shortlisted set of suggestions.

Assumptions and Constraints

Utility Metric

For both recommendation and exploration, there needs to be a metric to define 'utility,' each users' interests in specific pages. The best value given in the dataset in regard to user interest is the amount of time spent on a page. As such, we quantify utility in the following way:
For a given website and user, the utility is the fraction of time spent on that site (by the user) with respect to all other sites, falling in the range $[0, 1]$

Learning Assumption (IID)

For the purpose of machine learning, especially cross-validation and test-train split, it is assumed that the data are IID (independent and identically distributed).

Recommendation Assumption

When performing site recommendations, the learning algorithms assume that all existing Locus sites are reflected in the dataset. Therefore, any website not listed in the data cannot be recommended.

Local websites/files will not be included in the recommendation. See below for justification.

Data Assumption

Some websites are permissioned, but permissioning information is not available in the data set. We assume that by including enough relevant features, especially internal vs. external, the learning algorithm will generally avoid recommending unpermissioned sites to users.

Some websites are local to the users. This sites are tricky because there are potentially many unique websites, and these websites are only visited by a single user. As such, they provide a burden to learning algorithms in various ways. These sites will be ignored in the learning process, and as such, they cannot be recommended.

Each day will be broken into three time period, corresponding to morning, midday, and evening. Before 10:00 will be considered morning, after 15:00 will be considered evening. Any time in between will fall into midday.

Algorithms

Recommendation

Initially, a simple Bayes (Multinomial or Gaussian) model will be used to determine the homepage and other recommended sites for a user. The features for this model will be the current time period and the utility values for the users' sites visited in this time period. The top n results will be taken as the largest output probabilities and displayed as relevant pages.

A more complicated algorithm may be tested that incorporates the order in which users visit sites, in addition to the time of day and other relevant features. Using a model that is sensitive to one-hot-encoded time series data, like multinomial bayes or recurrent neural network (RNN) can predict a series of sites visited. This output can be cross-referenced with the simpler model to produce a more effective set of relevant pages.

To provide exploratory recommendations, SVD collaborative filtering will be used to make recommendations for websites the user may not have visited yet. These will complement the prior algorithms by providing a distinct list of elements.

Structure and Frameworks Used

Tool	Purpose
Python	General language for data analysis and web hosting
NumPy, Scikit	Processing dataset, creating feature vectors
Scikit-Learn	Machine learning (Naive Bayes classification)
Keras	Deep learning with time-series data (Recurrent Neural Network, LSTM)
TensorFlow	Backend for Keras
Flask	Simple web hosting with python

MVP and Stretch Goals

A minimum viable product to solve our problem is a functioning algorithm which identifies user presentations with reasonable accuracy. From this algorithm's output, we can better understand feature correlations between users with similar behavior.
Using this algorithm, we can build an intelligent back-end which utilizes users' and similarly behaving users' browsing metadata (including location, time, and page paths) to identify trends and predict sites of interest.
Stretch Goals
Our end goal is to deliver a functional UI which makes our intelligent back-end tangible. In addition, we aim to use correlations between our users' historical intranet site paths and external events (economic calendars, corporate events, day of week) to contextualize our prediction and increase accuracy.