Machine Learning Project

Pablo Bollansée & Kenan Alcı

March 2016

1 Literature

Ordered in perceived relevance to our project.

"A Web Page Prediction Model Based on Click-Stream Tree Representation of User Behavior" proposes a system that uses both order-of-pages and time-spend-on-page data to cluster similar user sessions together. They use this to assign the current session to a cluster, and so predicting future clicks. Their results also show the importance of time spend on a page, as this is a very good indicator of interest in the page.

"Predicting sequences of user actions" proposes a similar matrix representation as we want to use, but for console commands.

"Modeling Online Browsing and Path Analysis Using Clickstream Data" analyzes clickstream data of potential customers at a webshop. They show that sequences of webviewings are informative in predicting a user's path. They also show that shorter more directed paths tend to indicate interest and likelyhood of purchasing something.

"Selective Markov Models for Predicting Web Page Accesses" briefly explains previous attempts of using Markov Chains to predict web page accesses, and shows that higher-order Markov Chains typically give good results. However, these Markov models require quite a lot of computational power, at least when they are of higher order, so they propose a way of combining multiple Markov Chains, of different orders, to create a model that's both faster and more accurate than previous attempts.

We of course also read the other three papers suggested in the assignment. As well as skimmed through several papers that turned out to not be very interesting.

2 Overall pipeline

We want to test a system where we combine multiple learners to achieve a very fast and accurate prediction of pages that will be visited next. We want to include both information about sequential clicks, and time spend on a certain page.

Concretely this would mean that we plan to use the "click" events, which include the current page and the next one, to create a learner that learns which pages are accessed from which pages. We plan to use a Markov Chain representation: a simple matrix where rows represent current page and colums the clicked page. Each cell represents a chance of that page being visited. Multiplying the matrix with itself will give us chances for n-future pages.

Next we plan to use the "load" and "beforeunload" events to capture times spend on a page, as well as their age. This will give us a second learner that predicts how important a page is for the reader. As other papers have shown, time spend on a page is a good indicator of the interest of a person in that page. Visiting a page multiple times will also increase its importance.

Lastly we would like to create a learner that takes into account previously visited pages in the current session. Again using the "click" events, but now in reverse. Using the same Markov Chain data, we can predict what the chances are of going to a page, given a previous page.

In all these learners we also plan to include derivatives of the full url.

We will then combine the predictions of these different learners, to give suggestions of which pages would be most interesting for the user in the current session. We want to make a simple learner, that can learn while it's also being used for predictions without the need of too much preprocessing, and that is fast enough to be used in real-time.

3 Questions

- Can a simple matrix representation of clicks capture the sequence data in an acceptable amount of memory? Even with the derivatives of urls?
- How important are derivative urls in the final decision?
- How important is each learner, i.e., how much should each learner weigh in the final decision of presented link?
- Can this multi-learner approach give good predictions?
- Will this multi-learner approach be fast enough to use in a real-time application, which the browser is?