Visualizing Clickstream Data as Discrete Time Markov Chains
Shiribi Ish-Shalom, Samuel Hansen
CS448B: Data Visualization, Spring 2016, Stanford University

Abstract
From eCommerce to online dating, understanding how users navigate web pages is vital to online businesses. When a user visits a web page, she generates a sequence of page visits, known as a "clickstream". Taken together, aggregated clickstreams can answer many important questions, including how well a site predict page transitions, facilitate interpage traffic, and identify behavioral cohorts of users.

To assist the analysis of clickstream data, we present Clickstream Explorer, a visualization dashboard that represents aggregate clickstream data as a discrete time Markov chain. Clickstream Explorer improves upon prior visualization tools by employing:
1) Integration with the R computing language;
2) Multiple display options (graph view, table view, heatmap view, etc.);
3) Dynamic updating via user inputs.

In doing so, Clickstream Explorer enables the rapid exploration of clickstream data and associated Markov chain properties.

Motivation
Prior work such as WebQuilt\(^1\) depicted in Figure 1, and StatVis\(^2\) have represented clickstreams as hierarchical trees, in which nodes represent webpages, links represent transitions, and tree depth represents click number.

However, such tools become difficult to interpret as tree depth increases, prompting researchers to adopt the Markov assumption: given a certain state, no additional information is needed to predict the next state.\(^3\) Because of this, they have expressed formally:
\[
P(X_n | X_0, X_1, \ldots, X_{n-1}) \neq P(X_n | X_{n-1})
\]
Although researchers have used this Markov property to develop models of web browsing,\(^4,5\) build statistical analysis packages for clickstream data\(^6,7\), and predict user buying patterns,\(^8\) few tools provide comprehensive visualizations of Markov chain representations of clickstream data. Current Markov visualization tools in R - the most widely used statistical language - suffer from occlusion and lack of scalability. To solve this problem, we built an interactive dashboard to minimize issues of occlusion, display Markov chain properties, and facilitate rapid comprehension of aggregate clickstream behavior.

Methods
We created our dashboard in the following 6 steps:

1. **Clean Data**
   - We obtained 993,815 clickstreams from msnbc.com that were collected on Sept. 28, 1999. Each URL was pre-categorized into 17 page categories (e.g., "News", "Weather", "Sports"). Each clickstream was a comma-separated string, represented:
   
   
   2. **Convert to Transition Matrix**
   - To convert our clickstreams into an aggregate transition matrix, we first computed a matrix with counts from each state to each state. Then, we normalized this matrix by row sum to compute transition probabilities.

3. **Compute Markov Chain Properties**
   - We created a pipeline to analyze the following properties of the Markov chain defined by our computed transition matrix:
     1. **Irreducibility**: Are all states reachable from all other states?
     2. **Periodicity**: Do cycles exist in the chain?
     3. **Invariance**: Does an invariant distribution exist?

   For instance, our pipeline would classify the chain in Fig. 3 as non-irreducible (because there are absorbing states), aperiodic (because self-loops exist), and having no unique invariant distribution (given the Convergence Theorem).

4. **Create Weighted Edge List**
   - Using the transition matrix, we created a weighted edge list, which included the source page, target page, and respective transition probability as the edge weight. We used this edge list to create several visualizations, including our directed graph and heatmaps.

5. **Build RShiny Dashboard UI**
   - Because many Markov chain analysis tools exist in R, we wanted to integrate our visualization tool into pre-existing frameworks. This would allow statistical programmers who employ R packages such as markovChain, DTMC, and others to easily jump between the RStudio command line and our Clickstream Explorer.

6. **Connect RShiny Dashboard Server**
   - As a second step, we connected our UI to a separate server script, which dynamically updates graphical displays to be rendered on the dashboard. We used the computed edge list and Markov chain properties to build customized Markov graphs, interactive data tables, and heatmaps.

Results
The dashboard implements the following functionalities:

1. **Filter edges by transition probability**
2. **Convert to Transition Matrix**
3. **Select individual source node**
4. **Select multiple source nodes**
5. **View transition matrix heatmap**
6. **Search nodes in data table**

Future Work
1. **Automate Classification of URLs to Page Categories**
   - Markov chains representations of clickstreams are most effective when the state space is relatively small (e.g., less than 50). However, treating each unique URL as its own state leads to enormous state spaces, implying URLs must be classified into more general categories.
   - Future work should pursue strategies to classify URLs into Markov state categories (e.g., via similarity-based clustering techniques).

2. **Implement Markov-Property Validation Tool**
   - We selected our dataset partly because prior empirical testing demonstrated first-order Markov models are suitable to approximate aggregate clickstream behavior on msnbc.com. However, to generalize to future datasets, future work could develop a pipeline to validate whether a given dataset satisfies the first-order Markov assumption. Such a validation tool could be integrated into Clickstream Explorer, in addition to a dataset upload feature.

References