Visualizing Clickstream Data as Discrete-Time Markov Chains

Shirbi Ish-Shalom[1] and Samuel Hansen[2]

***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". Each clickstream leaves a digital trace researchers can use to understand online choice behavior. Taken together, clickstreams from many users can answer a variety of questions, including how well a site predict page transitions [9][22][25], facilitate interpage traffic, and identify behavioral cohorts of users[19]. To aid exploratory analysis of clickstream data, we developed *Clickstream Explorer*, an RShiny Dashboard application that visualizes clickstream data as discrete-time first-order Markov chains.**

I. INTRODUCTION

Recent clickstream literature has focused on two main subfields: 1) how users navigate pages within a site, and 2) how users enter and exit across a site. For instance, to model within-site behavior, Mandel and Johnson (2002), examined changes in clickstream data to show that visual primes (such as advertisements) can dynamically affect user click paths. To model across-site behavior, Bucklin and Sismeiro (2003) collected clickstream data from a large website in the automotive industry to build a type II tobit model estimating the amount of time spent on each page, finding that effective websites should "[reduce] the number of page views needed to complete a transaction"[4][13]. These studies, among others, reflect growing interest in understanding clickstreams amongst academic researchers and industry professionals alike.

A. Related Work in Visualization

Growing interest in clickstreams has precipitated the need for effective visualization tools. Visualizing clickstream data involves converting sequences of page visits into an aggregate representation of user paths. Researchers and designers have proposed a multitude of ways to visualize clickstream data, the most common being directed graphs and Sankey diagrams. Numerous tools represent clickstream data as directed graphs, in which nodes represent webpages and edges represent transitions. For instance, in *Understanding Clickstreams with the WebQuilt Visualization System*, Waterson et al. present *WebQuilt*, an interactive visualization tool that aggregates clickstream data into a customizable graph (see Figure 1) [26]. *WebQuilt* adopts the directed graph approach by representing pages as nodes connected by arrows whose thicknesses convey the densities of interpage traffic at each click.

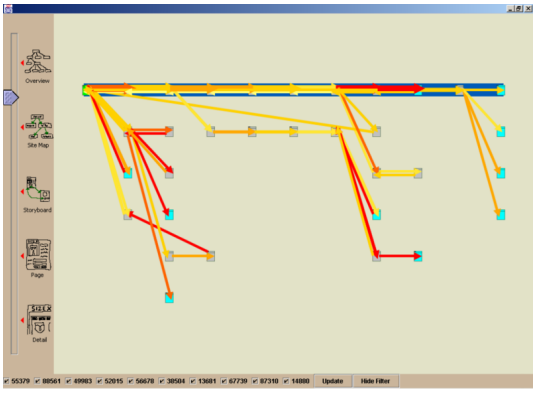

Fig. 1: Waterson et al.'s *WebQuilt* visualization system (2002).

StatViz, a similar system developed by open-source designer Alan Pinstein, builds upon the directed graph approach by introducing an implicit tree hierarchy to represent the click number in a clickstream sequence, where the i^{th} level in the tree represents the i^{th} click [16].

As an alternative to graph representations, Sankey diagrams are often used to convey the volume of flow between web pages. Sankey diagrams, which depict flow from one source to a corresponding target, represent webpages as vertical bars in a left-to-right diagram connected by frequency-weighted edges (see Figure 2).

Fig. 2: A Sankey diagram in Schulz et al.'s SAS Visual Analytics tool (2015).

[1]S. Ish-Shalom is a coterminal master's student in Biomedical Informatics, Stanford University, 450 Serra Mall, Stanford, CA shirbi at stanford.edu

[2]S. Hansen is a coterminal master's student in Management Science and Engineering, Stanford University, 450 Serra Mall, Stanford, CA sfhansen at stanford.edu

Google Analytics recommends Sankey diagrams to show "traffic flows from pages to other pages on your web site" primarily because multiple paths can be displayed with minimal occlusion [6]. In addition to Google Analytics, several independent designers[28] have developed Sankey diagrams of site traffic. For instance, Schulz et al. employ Sankey diagrams in their industry tool, SAS Visual Analytics, an interactive software package[21]. This tool (shown in Fig. 2) represents each page visit as a vertical bar in a left-to-right layout. The frequency of visits between any two pages is encoded by the thickness of the connecting edge.

B. Related Work in Markov Chains

Directed graphs and Sankey diagrams attempt to visualize clickstreams by taking into account click number, displaying aggregated information about being in a certain state after a given number of clicks. However, a growing number of researchers deviate from this practice by making the following assumption: given a certain state, no additional information is needed to predict the next state [4][9][19][22][25]. This assumption implies clickstreams can be represented by discrete-time first-order Markov chains, expressed mathematically:

$$\Pr(X_n = x_n | X_{n-1} = x_{n-1}, ..., X_0 = x_0)$$
$$= \Pr(X_n = x_n | X_{n-1} = x_{n-1})$$

Numerous researchers have used this Markov property to develop models of web browsing[4][14][25], build statistical analysis packages for clickstream data[7][23][20], and predict user buying patterns[10].

However, statistical advances in the analysis of clickstream data as Markov chains have not been accompanied by complementary visualization tools. Two recent surveys by O'Reilly and Rexer Analytics revealed that R is the most widely used data mining and analytics tool amongst data science professionals, compared to other resources[12][18]. Nevertheless, preeminent analysis packages in R include poor visualizations of Markov chain objects (see Fig. 3).

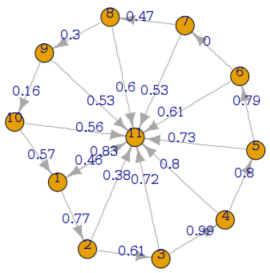

Fig. 3: Default Markov chain plot in the markovchain R Package.

These built-in plotting functions provide no interactivity, occlude many edges, and are largely unaesthetic. For instance, the default plot from the *markovchain* package in R (Figure 3) includes bidirectional arrows with respective transition probabilities, which leads to considerable occlusion when there are even slightly more states or edges. This representation omits important information such as invariant distribution information, chain properties (i.e. periodicity, irreducibility, etc.), and overall interactivity.

Given the popularity of R as an analysis tool and the lack of effective visualization functions for Markov chain visualizations of clickstreams, we developed an R-compatible application that integrates better Markov chain visualizations with statistical analysis packages. We present our methods of development, visualization results, and proposed future work.

II. Methods

A. Data Collection

Our objective was to design a visualization tool capable of generating Markov chain representations of clickstream data. Although our tool is designed to generalize, we chose to test our implementation using clickstream data from Internet Information Server (IIS) logs for *msnbc.com*, a popular news site. The data, which are made available by *msnbc.com*, were collected for the entire day of September 28, 1999 (PST). Each log in the data represents a clickstream of page sequences of a given user. Importantly, each page is not a unique URL, but rather a category page such as news, weather, etc. The data include 17 categories: "frontpage", "news", "tech", "local", "opinion", "on-air", "misc", "weather", "health", "living", "business", "sports", "summary", "bbs" (bulletin board service), "travel", "msn-news", and "msn-sports". Additionally, "Any page requests served via a caching mechanism were not recorded in the server logs and, hence, not present in the data"[8].

We chose this dataset as a template for three reasons. First, the data were pre-encoded into categories, which is necessary to limit the size of the state space for the Markov chain, and consistent with previous literature[5][14]. Second, Cadez et al.[5] have presented prior experiments using the same data set that validate the suitability of first-order Markov models. Third, the open-source availability of the data lends itself to reproducibility by future researchers. Although we present our visualization using MSNBC data, our tool is intended to generalize to other categorized clickstream data (see supplementary).

B. Data Cleaning

The data were imported into R in CSV form, where each row represented a comma-separated string of integers corresponding to the codes for each page category. The original data include 989,818 clickstream observations; however, many clickstreams included only one page visit. In turn, we filtered out observations with only one page visit in order to visualize clickstreams that had at least one page transition, yielding a final sample of 704,747 observations. We made this assumption to reduce noise produced by singleton page visits.

C. Transition Matrix Computation

To convert the clickstream data into an aggregate transition matrix, we first computed an $m \times m$ matrix with bigram counts from each state *i* to each state *j*. To do so, we counted all instances of some state *i* preceding state *j* across all clickstreams. Then, we normalized this matrix by each row sum to compute transition probabilities to obtain a transition matrix *P*, where:

$$P(i,j) = \frac{count(i,j)}{\sum_k^m count(i,k)}$$

D. Chain Properties Analysis

Subsequently, we created a pipeline to analyze three properties of the Markov chain defined by the computed transition matrix: 1) irreducibility, 2) periodicity, and 3) invariant distribution existence. A Markov chain is irreducible if all states are reachable from all other states; otherwise, it is reducible if absorbing states exist. A Markov chain is periodic if it includes a cycle of period *k*; otherwise, it is aperiodic. Lastly, the Convergence Theorem guarantees that a unique invariant distribution exists iff the chain is irreducible and aperiodic; otherwise, no unique invariant distribution exists. The invariant distribution of a discrete-time Markov chain represents the long-run amount of time spent in each state. This property is particularly useful for clickstream analysis because it provides an estimate of which pages are visited most often.

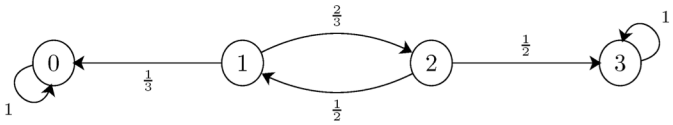

Fig. 4: Example of a reducible, aperiodic Markov chain without a unique invariant distribution.

The Markov chain in Figure 4, for example, is reducible because it is impossible to get to state 1 from state 0 and aperiodic because no cycles exist. Because the chain is reducible, the Convergence Theorem implies that no unique invariant distribution exists.

We computed these chain properties by constructing a Markov chain object from the transition matrix using the markovchain R package. Then we applied mathematical functions to return boolean values for whether the chain was irreducible and aperiodic. If the chain was irreducible and aperiodic, we computed the invariant distribution using the statdistr function in the *DTMCPack* R package, which returns a vector of m numerics corresponding to the long run time spent in each state. This function solves:

$$\pi = (1, ...1)(I - P + ONE) - 1$$

Here, "*I* is an $m \times m$ identity matrix, *P* is an $m \times m$ transition matrix, and *ONE* is an $m \times m$ matrix whose entries are all 1"[15]. We stored the boolean values and numeric vector for subsequent use in the visualization.

E. Edge List Conversion

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 visualizations including our directed graph and heatmap.

F. RShiny Dashboard User Interface

Because many Markov chain analysis tools exist in R, we integrated our visualization tool into pre-existing frameworks. This would allow statistical programmers who employ R packages such as *markovchain*, *DTMCPack*, and others to easily jump between the RStudio command line and *Clickstream Explorer*. In turn, we built an RShiny Dashboard application that can be run directly from the RStudio command line. As a first step, we built the user interface (UI) component, which included 1) a sliderbar to filter by transition probability threshold, 2) a toggle button to display probability edge labels, 3) a dropdown menu to filter by source node, 4) clickable tabs for a graph, data table, and heatmap, and 5) infoboxes for to display chain properties.

Several design principles inspired our development of the user interface. First, we attempted to maximize ease of interpretability by visualizing the Markov graph as a static plot, rather than a moving network. Our initial iteration visualized the Markov chain as force-directed graph using D3 wrappers in R. Although this visualization provided interactivity with nodes (by dragging them around the screen), the slow Gaussian movement of the nodes and edges impeded readability of edge labels and node names. After consulting prior literature, which suggested people have difficulty tracking more than 4-6 moving objects[17][1][24], we redesigned our dashboard to display filter-based static plots.

Second, to assist exploratory analysis, we incorporated three displays into our dashboard: a Markov graph, a data table view, and a transition matrix heatmap. Each display serves a distinct purpose: the Markov graph conveys the chain's overall structure, including the direction of edges and the size of nodes (which encoded invariant distribution magnitude). The graph view was designed to convey a holistic view of the chain's structure and behavior. The data table view enables the user to search for information about individual source and target nodes, and displays the numeric values of transition probabilities. Lastly, the heatmap displays a color-coded transition matrix from each state i to each state j, which visualizes overall patterns in the chains transition behavior.

As a third design principle, we developed data filtering methods that would allow rapid exploration of the Markov chain. For instance, we added a sliderbar to specify the minimum transition probability threshold. This allowed the user to iteratively view smaller versions of the chain that displayed the most prominent transitions. Importantly, this sliderbar simultaneously updated the Markov graph as well as the data table, thereby facilitating joint exploration across the two views.

G. RShiny Dashboard Server

After developing the widgets of the user interface, we connected the UI to a server script, which acted as a back-end for the data filtering and graph generation processes. Thus, all of the data manipulation and Markov chain computations were handled by the server, and the front-end display was handled by the UI.

III. Results

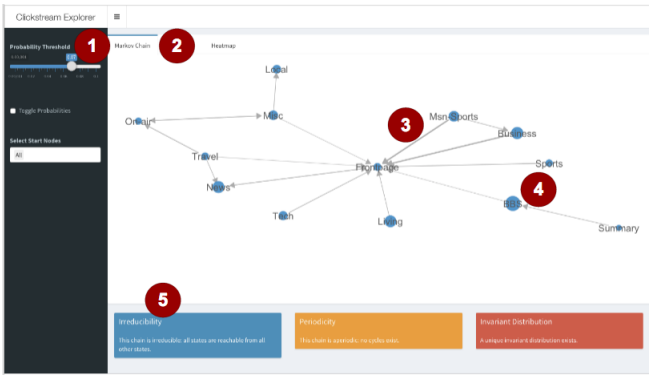

Fig. 5: *Clickstream Explorer*'s Markov graph view displaying 1) the probability threshold sliderbar, 2) three tab views, 3) variable edge thicknesses, 4) invariant distribution node sizes, and chain property information boxes.

To present our results, we include 4 screenshots from *Clickstream Explorer* and discuss its features. Each numbered item in the following list corresponds to the feature with the same number overlayed on figures 5-8. Figure 5 displays the Markov graph view of our dashboard with the following features:

1) The sliderbar sets the minimum transition probability between nodes. When the slider is dragged all the way to the left, the dashboard displays the entire connected chain; when it is dragged all the way to the right, the dashboard displays a partial version of the whole graph.
2) The three tabs allow the user to transition between the graph, data table, and heatmap views.
3) The thickness of an edge conveys the magnitude of the transition probability, where thicker edges represent more probable transitions. Figure 5 shows that most transitions lead to or from the frontpage.
4) The size of a node represents the magnitude of the invariant distribution for that state: bigger nodes are more likely to be visited in the long run. If no unique invariant distribution exists (i.e. if the chain is periodic), all nodes are equally sized. Figure 5 shows that BBS (Bulletin-Board Service) has the largest invariant distribution value, which is consistent with intuition because it is an MSNBC messaging service.
5) The three information boxes display the chain's class properties, including whether the chain is irreducible, aperiodic, and has a unique invariant distribution. In this case, the chain is irreducible, aperiodic, and possesses a unique invariant distribution.

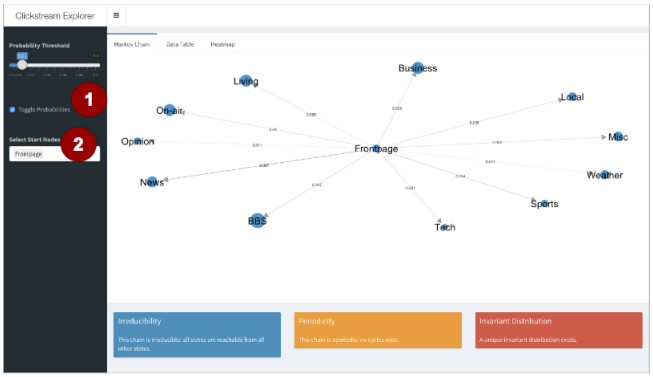

Fig. 6: *Clickstream Explorer*'s toggle button and source node search feature.

Figure 6 shows an alternate view, which highlights 2 additional features of the Markov graph view:

1) The "Toggle Probabilities" button displays the graph with labeled transition probabilities on the edges. This feature is useful because it allows the user to toggle between seeing the holistic graphs structure (without edge labels) and seeing the values of individual transitions (with edge labels).
2) The "Select Start Nodes" dropdown selector allows the user to display individual source nodes, such as "Frontpage" in Fig. 6. This feature enables the user to reduce the complexity of the graph and concentrate on the outbound transition behavior of a single node. Additionally, users can select arbitrary combinations of source nodes, such as "Tech" and "Business", to compare transition patterns.

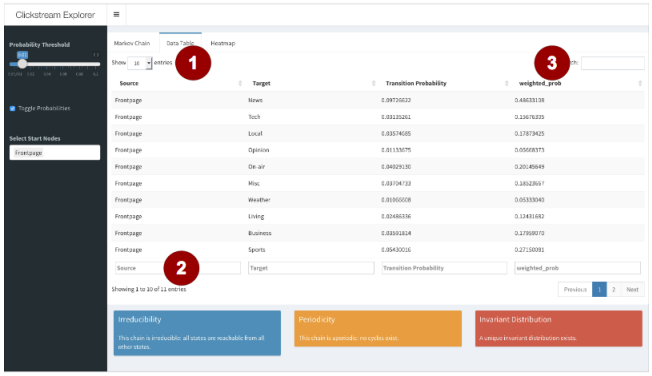

Fig. 7: *Clickstream Explorer*'s data table view, featuring 1) the observation number dropdown selector, 2) individual column search bars, and 3) a global search bar for the entire table.

Figure 7 shows the data table view, which displays the numeric transition probability values of states that satisfy the probability threshold of the sliderbar. The data table includes the following features:

1) Users can select from a dropdown menu to display 10, 20, or 50 observations in the data table.
2) Users can search for individual source, target, and transition probability values in each respective column.
3) Users can search the entire data table for occurrences of specific states - regardless of whether they are source or target nodes.

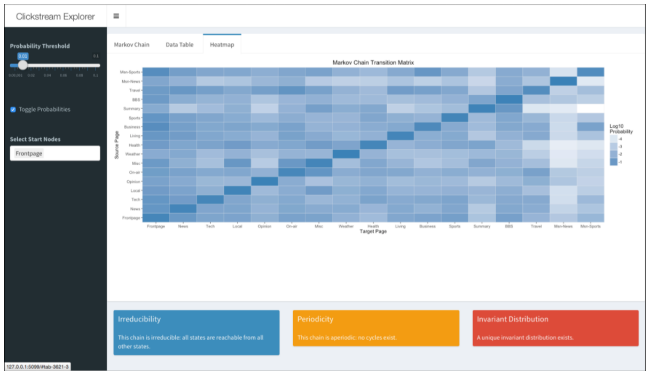

Fig. 8: *Clickstream Explorer*'s transition matrix heatmap view.

Lastly, Figure 8 shows the transition matrix heatmap, which displays a color-coded representation of the most and least common transitions between states. Darker colors represent more frequent transitions, and lighter colors represent less frequent transitions. For instance, the dark diagonal of the matrix implies self-loops are common in the MSNBC data, likely because visitors stay in one topic category (i.e. Sports, or Travel), rather than jumping across categories. We also observe that the MSN-News column is relatively lighter, meaning that fewer states lead to MSN-News, other than MSN-News itself. Insights such as these are evident from the heatmap.

We used a log base 10 scale for the color range because transition probabilities ranged from 0.0000125 to 0.98. We chose the blue color scheme of the heatmap to match the blue color of the sliderbar, toggle button, and first infobox.

IV. Evaluation

To test the efficacy of *Clickstream Explorer*, we performed a usability study using a small sample of Stanford engineering students. Participants performed two time-based comprehension tasks and answered two survey questions.

First, participants we were instructed to use any features of *Clickstream Explorer* to answer the question, "Which is the most visited node?" This question was meant to test the effectiveness of encoding node size as invariant distribution magnitude. We timed their responses and plotted the time distribution in Figure 10. Fig 10 shows that the median time to report the most visited page was 3.38 seconds.

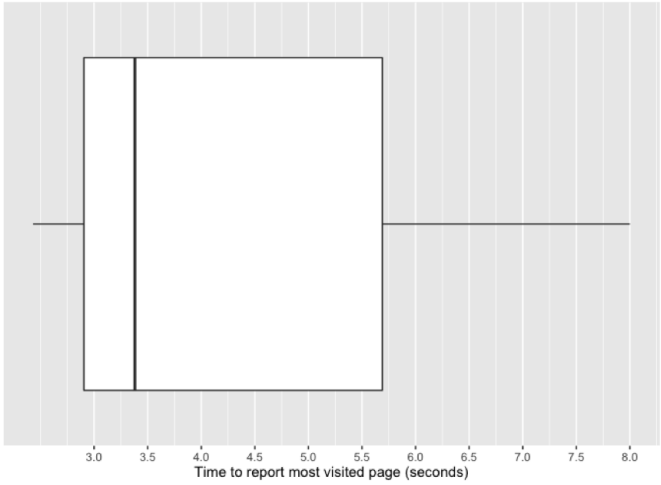

Fig. 9: Distribution of time to report most visited page.

Second, participants were were instructed to use any features of *Clickstream Explorer* to answer the question, "Which page is most likely to be visited, starting from the Tech page?" Again, we timed their responses, which are plotted in Figure 9. Figure 9 shows a slightly larger time distribution, with a median task completion time of 15 seconds. This slightly larger time distribution is likely due to differences in learning effects amongst participants. We expect the median time to identify transitions between two nodes to decrease with time spent using *Clickstream Explorer*; however, further research is needed to evaluate *Clickstream Explorer* against relevant alternative tools, such as *WebQuilt*, *StatViz*, and default R package plots.

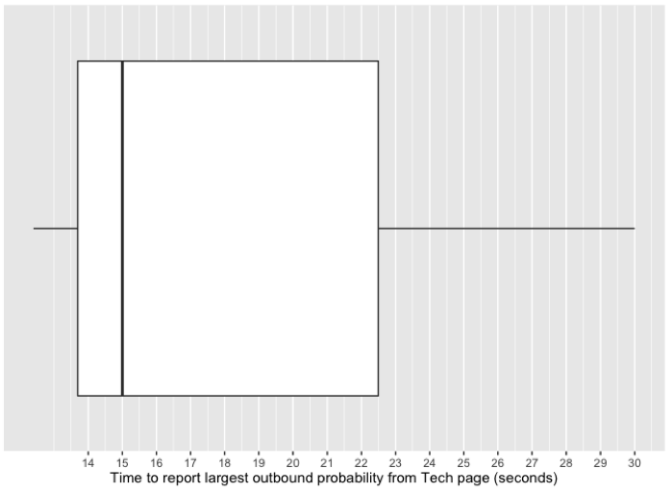

Fig. 10: Distribution of time to report largest outbound probability from Tech page.

Lastly, we asked participants, "Which feature was your favorite?" and "Which feature did you wish was present?" Participants identified the probability threshold sliderbar and the source node searchbar as their favorite features because they facilitated rapid exploration of the Markov chain. Additionally, most participants wanted transition probability edges

to be color-coded along a gradient. We hope to pursue this feature in future work.

V. Discussion

A. Summary of Findings

We used *Clickstream Explorer* to investigate two clickstream-related questions.

First, we were motivated by the question, "Are users more likely to stay within a single page category, or are they more likely to transition across categories?" We examined the transition matrix heatmap and observed an apparent trend: the dark diagonal revealed self-loops are more common than inter-category transitions. In other words, users are more likely to stay in a given category, rather than jump between categories.

Second, we asked, "Which pages are uncommon targets?" After adjusting the sliderbar, the Markov graph revealed fewer edge links to MSN-News. Similarly, the heatmap showed lighter colors in the MSN-News column, implying MSN-News is an uncommon target page across source nodes. We investigated these questions as a proof of concept: namely, that our *Clickstream Explorer* tool lends itself to exploratory analysis of clickstream data.

B. Sources of Error

Clickstream Explorer faces one main limitation: the tool is optimized to model within-site behavior, rather than across-site behavior. Because our data did not include how users entered or exited the site, we were limited to modeling within-site activity. Although our tool could handle future data with "Enter" and "Exit" categories, these categories would likely be too general to yield meaningful across-site insights. However, *Clickstream Explorer* still offers value because within-site behavior is a rich field of inquiry its in own right.

C. Comparison to Prior Work

Prior work such as *WebQuilt*[26] and *StatViz* [16] represent clickstreams as hierarchical trees in which levels represent click numbers. However, this approach suffers from severe occlusion when trying to visualize longer clickstreams because deeper trees obscure information of individual nodes. *Clickstream Explorer* addresses this problem is two ways: 1) it uses the Markov assumption to reduce the number of transitional edges, and 2) it incorporates dynamic filtering to visualize subcomponents of the Markov graph. Furthermore, *Clickstream Explorer* goes beyond the default Markov plots in R packages such as *markovchain* and *clickstream* by incorporating interactivity and multiple tabs (i.e. Markov graph, data table, and heatmap views).

D. Future Work

Markov chain 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 automate the classification of URLs into Markov state categories (e.g., via similarity-based clustering techniques).

Furthermore, we selected our dataset partly because prior empirical testing demonstrated first-order Markov models are suitable to approximate aggregate clickstream behavior on *msnbc.com*[5]. 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 conclusion, we hope *Clickstream Explorer* adds better visualization functionality to existing Markov chain packages and aids the analysis of Markov stochastic processes in R.

Acknowledgments

We thank the University of California, Irvine Machine Learning Repository and *msnbc.com* for making the data available and CS448B teaching staff Ludwig Schubert, Scott Cheng, Peter Washington, and Maneesh Agrawala for their guidance in developing *Clickstream Explorer*. We additionally thank the Stanford students who participated in our usability study.

References

[1] Alvarez, George, and Steven L Franconeri. How Many Objects Can You Track? Evidence for a Resource-Limited Attentive Tracking Mechanism. Journal of vision 7.13 (2007): 14.110. Web.

[2] Baldini, Paolo, and Paolo Giudici. Improving Web Clickstream Analysis: Markov Chains Models and Genmax Algorithms. 233235. Web.

[3] Bodesinsky, Peter et al. Exploration and Assessment of Event Data . (2015): n. pag. Web.

[4] Bucklin, Randolph E, and Catarina Sismeiro. A Model of Web Site Browsing Behavior Estimated on Clickstream Data. Journal of Marketing Research XL.August (2003): 249267. Web.

[5] Cadez, Igor et al. Visualization of Navigation Patterns on a Web Site Using Model Based Clustering. Data Mining and Knowledge Discovery 7.4 (2000): 399424. Web.

[6] Google.com. Sankey Diagram. N.p., 2016. Web. 22 May 2016.

[7] Hahsler, M, and M H Dunham. rEMM: Extensible Markov Model for Data Stream Clustering in R. Journal of Statistical Software 35.5 (2010): 131. Web.

[8] Heckerman, David. MSNBC.com Anonymous Web Data Data Set. University of California, Irvine Machine Learning Repository. N.p., n.d. Web. 24 May 2016.

[9] Lakshminarayan, Choudur, Ram Kosuru, and Meichen Hsu. Modeling Complex Clickstream Data by Stochastic Models: Theory and Methods. WWW 16 Companion. Geneva, Switzerland: N.p., 2016. 879884. Web.

[10] Lim, Miseon, Hyunsoo Byun, and Jinhwa Kim. A Web Usage Mining for Modeling Buying Behavior at a Web Store Using Network Analysis. 8.October (2015): n. pag. Web.

[11] Mackinlay, Jock. Automating the Design of Graphical Presentations of Relational Information. ACM Transactions on Graphics 5.2 (1986): 110141. Web.

[12] Magoulas, Roger, and John King. 2013 Data Science Salary Survey. N.p., 2014. Web.

[13] Mandel, Naomi, and Eric J. Johnson. When Web Pages Influence Choice: Effects of Visual Primes on Experts and Novices. Journal of Consumer Research 29.2 (2002): 235245. Web.

[14] Montgomery, Alan L et al. Modeling Online Browsing and Path Analysis Using Clickstream Data. Marketing Science 23.4 (2004): 579595. Web.

[15] Nicholson, William. Package DTMCPack. 2015. Web.

[16] Pinstein, Alan. StatViz. 2013. Web.

[17] Pylyshyn, Z W, and R W Storm. Tracking Multiple Independent Targets: Evidence for a Parallel Tracking Mechanism. Spatial vision 3.3 (1988): 179197. Web.

[18] Rexer Analytics. Rexer Analytics 2013 Data Miner Survey. Boston: N.p., 2013. Web.
[19] Sadagopan, Narayanan, and Jie Li. Characterizing Typical and Atypical User Sessions in Clickstreams. Proceedings of the 17th international conference on World Wide Web (2008): 885. Web.
[20] Scholz, Michael. Package Clickstream . 2016. Web.
[21] Schulz, Falko, and Olaf Kratzsch. Taking the Path More Travelled SAS Visual Analytics and Path Analysis. (2015): 114. Web.
[22] Scott, Steven L, and Il-Horn Hann. A Nested Hidden Markov Model for Internet Browsing Behavior. Wp (2006): n. pag. Web.
[23] Spedicato, Giorgio Alfredo, and Mirko Signorelli. The R Package "markovchain": Easily Handling Discrete Markov Chains in R. Cran (2014): n. pag. Web.
[24] Tversky, B., Morrison, J. B., and Betrancourt, M. Animation: Can It Facilitate? International journal of human-computer studies (2002): 247262. Web.
[25] Wang, Gang et al. You Are How You Click: Clickstream Analysis for Sybil Detection. (2013): n. pag. Web.
[26] Waterson, Sarah J et al. What Did They Do? Understanding Clickstreams with the WebQuilt Visualization System. Proceedings of the Working Conference on Advanced Visual Interfaces AVI 02 (2002): 94. Web.
[27] Zhao, Jian et al. MatrixWave. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI 15 (2015): 259268. Web.
[28] Zwitch, Randy. Visualizing Website Pathing With Sankey Charts. N.p., 2014. Web. 22 May 2016.