Visual Cluster Exploration of Web Clickstream Data

Jishang Wei†
University of California, Davis

Zeqian Shen‡
eBay Research Labs

Neel Sundaresan‡
eBay Research Labs

Kwan-Liu Ma§
University of California, Davis

ABSTRACT

Web clickstream data are routinely collected to study how users browse the web or use a service. It is clear that the ability to recognize and summarize user behavior patterns from such data is valuable to e-commerce companies. In this paper, we introduce a visual analytics system to explore the various user behavior patterns reflected by distinct clickstream clusters. In a practical analysis scenario, the system first presents an overview of clickstream clusters using a Self-Organizing Map with Markov chain models. Then the analyst can interactively explore the clusters through an intuitive user interface. He can either obtain summarization of a selected group of data or further refine the clustering result. We evaluated our system using two different datasets from eBay. Analysts who were working on the same data have confirmed the system’s effectiveness in extracting user behavior patterns from complex datasets and enhancing their ability to reason.

1 INTRODUCTION

Clickstreams record user clicking actions on the web. Analyzing clickstream data provides insights into user behavior patterns, which are extremely valuable for e-commerce businesses like eBay. For example, knowing different user behavior patterns helps conduct market segmentations in order to develop marketing strategies, or enhance personalized shopping experience. In practice, however, learning the various user behavior patterns is nontrivial. Analysts often have little knowledge but many questions about what user behaviors are hidden in a clickstream dataset. By interviewing analysts at eBay, we found the following questions are frequently asked.

- What are the most frequent user behavior patterns?
- What are the demographics of the users who follow a specific behavior pattern?
- How do the behavior patterns correlate with the performance of the online service?

An interactive data exploration environment provides a fairly effective way of finding answers to these questions. In our work, we design a visual analytics system to support such an answer-seeking process. This visual analytics system enables analysts to inspect different user behavior patterns and examine the associated pattern profiles.

Nevertheless, clickstream data have a number of characteristics [5] that make the visual analysis task challenging. For example, clickstreams are inherently heterogeneous, and uncertainty naturally arises during automatic clustering which makes it indeterminate to partition the data. In addition, it is uneasy to visually summarize a cluster of clickstreams. It might be easy to describe the group behavior verbally, but difficult to present visually.

To handle these challenges, in our work, we utilize a derived Self-Organizing Map(SOM) to map and cluster the clickstreams; design an enlightening visualization from which analysts can see the clear cluster structure; and provide intuitive interaction tools to enable detailed data examination and cluster structure refinement. The major contributions of our work include:

- An SOM with Markov chain models is derived to map clickstreams to a 2D space.
- A 2D layout algorithm is introduced to reduce visual clutter.
- An interactive cluster exploration process is designed to enable user-guided clustering.

We have used real clickstream data to test and evaluate our method with assistance from analysts and product managers at eBay. Our method proves to be helpful in discovering user behavior patterns and the corresponding demographics.

2 RELATED WORK

The visual cluster exploration of clickstreams is related to data clustering, visualization and interactive exploration.

Clustering web clickstream data to discover user behavior patterns has drawn great attention in web usage mining [22, 24]. The algorithms for clickstream data clustering mainly fall into two categories: similarity-based and model-based methods. Similarity-based methods define and utilize similarity metrics to group similar clickstreams. An appropriate similarity metric is the core of this type of methods. The edit distance, related longest common subsequence and their variations are the most frequently used metrics [2, 7, 9, 13]. Model-based clustering employs probabilistic models to cluster and describe clickstream data [5, 17]. This method assumes that user behaviors are generated by a set of probabilistic models and each model corresponds to a cluster. The clustering process is to recover the model parameters and assign clickstreams to the cluster where the model best describes them. Choosing an application suitable model is critical. There have been many models proposed to describe user behaviors [5, 17, 19, 21, 28], and Markov models (e.g. first order Markov models, or Hidden Markov models) are the most commonly utilized ones. Compared to similarity-based methods, model-based methods offer better interpretability since the resulting model of each cluster directly characterizes that cluster. Moreover, model-based clustering algorithms often have a computational complexity that is “linear” to the number of data objects under certain practical assumptions.

A proper visualization of clickstream clusters reveals data patterns. One direct strategy is to use separate windows with each showing one cluster. For example, Cadez et al [5] and Manavoglu et al. [17] visualized the clusters of clickstreams with multiple windows. Each window shows the clickstream traces of one kind. This method emphasizes individual clickstream patterns, but cuts the connections among clusters. Space-filling approaches are a widely used type of visualization technique, which arrange and display the data in a constrained space. The treemap was popular among them. For instance, Xu et al. [27] applied treemap to display large digital collections. The treemap was improved through

*e-mail: jswei@ucdavis.edu
†e-mail: zeqshen@ebay.com
‡e-mail: nsundaresan@ebay.com
§e-mail: ma@cs.ucdavis.edu
Interactive, visually-aided cluster analysis has been studied to handle clickstream data. Makanju et al. [16] introduced a visual analysis tool, LogView, to cluster server log event sequences and visualize the clustering result by treemaps [10]. LogView provides interaction tools to enable searching, filtering and selection of data based on the treemaps. Lee and Podlaseck [15] described an interactive visualization system that supports interpreting and exploring clickstream data of online stores. Their system includes interaction tools such as zooming, filtering, color coding, dynamic querying, and data sampling, in addition to visualization options such as parallel coordinates and scatter plot. Takada and Koike [23] developed a highly interactive visual log browser named MieLog. MieLog uses interactive visualization and statistical analysis for manual log inspection tasks. Lam et al. [14] built Session Viewer, a visualization tool to facilitate statistical and detailed analysis of web session data and bridge gaps between them. Shen and Sundaresan [20] introduced TrailExplorer to analyze temporal user behavior patterns in webpage flows. TrailExplorer is particularly tailored to assist visual analysis of preprocessed web session data.

3 DATA AND APPROACH OVERVIEW

A clickstream is an ordered sequence of predefined actions. For example, consider a seller listing items for sale on eBay, they need to complete the Sell-Your-Item page. The page requests extensive selling related information, which are categorized into eight sections: Category: select a category where the item will appear; Title: write a title for the item; Picture: select or upload pictures of the item; Description: describe the item; Pricing: set a price for the item; Payment: set the payment options; Shipping: choose shipping methods and set the shipping cost; OtherOptions: set other information about the listing, such as tax and return policy. The clickstream data from the Sell-Your-Item page captures user actions in terms of the sections they edit. A sample of the data is shown in Figure 1, where each row represents one clickstream. We can see that users edited the sections in different ordering, skipped some sections and revisit some in the same clickstream. User demographics and other selling related information are collected as well for understanding their correlations with user behaviors.

Figure 2 shows the general visual cluster exploration process using our system. First, clickstreams are mapped to a 2D plane, while the topological relations among data are preserved. Second, the clickstreams are visually encoded, and their placement are adjusted to reduce visual clutter. This 2D visualization makes it possible to both observe an overview of the cluster structure and perceive individual clickstream patterns. Third, analysts interpret the visual representation, interact with data to obtain detailed information and select representative clickstreams as cluster prototypes. Fourth, the selected groups can be either shown with statistical summarization or used to refine the clustering result. Users can iterate between the third and fourth steps until a satisfactory result is achieved. The rest of the paper is organized as follows. Our methods for data mapping and visualization are introduced in Section 4 and Section 5, respectively. Section 6 addresses how our system supports interactive cluster exploration. Finally, two case studies are presented to demonstrate how the system is used in practice.

4 DATA MAPPING AND CLUSTERING

Practically speaking, it is intuitive for people to perceive and interact with data in a lower dimensional space. In our work, we choose to map the clickstreams on a 2D plane to facilitate data perception. Meanwhile, we hope to partition data into clusters so as to reveal different behavior patterns. The Self-Organizing Maps (SOM) [11, 12] is such a method that can achieve our goal. However, the conventional SOM is designed to handle data of the same dimensionality, which are quite different from clickstreams of different lengths. To solve this problem, we utilize probabilistic models within the SOM framework to accommodate the clickstream data mapping and clustering. In this section, we will first give a brief introduction to SOM and then elaborate on how probabilistic models are integrated into the SOM framework.

4.1 Self-Organizing Map

The Self-Organizing Map is a well-known neural network model for high-dimensional data mapping and clustering. It consists of components called nodes or neurons which are usually arranged on a regular 2D grid. One node is associated with a vector prototype representing a cluster of input data, and nearby nodes contain similar vector prototypes. A trained SOM can map high-dimensional data into a 2D space while maintaining topological relations among data objects.

The SOM is usually trained by the competitive learning method, which iteratively updates the vector prototypes. For instance, a batch algorithm trains SOM iteratively by applying the following two steps,

- **Matching**: for each prototype \( m \), collect a list of clickstreams, whose best matching (most similar) prototype is within the neighborhood of \( m \);
- **Adjusting**: adjust the prototypes using their respective collections of matching clickstreams.

During the training process, the neighborhood size decreases at each iteration. The neighborhood relation \( h(i, j) \) between two prototypes \( i \) and \( j \) is determined by the geometric relations of their corresponding grid points \( o_i (x_i, y_i) \) and \( o_j (x_j, y_j) \). The commonly used neighborhood relation is the Gaussian function, i.e.,

\[
h(i, j) = \frac{1}{2\pi} \exp\left(-\frac{|o_i - o_j|^2}{2 \times \delta^2}\right)
\]

After the SOM training, a set of vector prototypes are obtained representing the input data, with similar prototypes staying closer. The input data are also projected into a low-dimensional space, i.e., a 2D regular grid.

4.2 SOM with Markov Chain Models

Regarding the conventional SOM, the vector prototypes have the same dimensionality as input data. At the matching step, the similarity between an input data item and a prototype is measured by a
Figure 2: The major steps of our system for visual cluster exploration of clickstream data include data mapping, data visualization, interactive cluster exploration, and clusters verification and validation.

pre-defined similarity metric, such as Euclidean distance. At the adjusting step, the prototypes are simply updated by taking the mean value over their respective lists of matching data items. However, because the clickstream data are heterogeneous and of different lengths, it is not trivial to design a feasible similarity metric and to use specific vectors to represent clickstreams. To deal with this issue, we take advantage of the probability models as prototypes to describe the clickstreams. The “similarity” between a clickstream and a probability model is measured by the probability at which the clickstream fits in the model. As such, SOM with probability models can be applied to map the clickstream data onto a 2D plane.

In the literature, there have been many models proposed to describe clickstreams [5, 17, 19, 21, 28]. We use a first-order Markov chain as in [5] since the Markov chain is a simple but efficient model to describe user behaviors. Other models can also be adapted in the SOM framework provided that the model can capture the data characteristics well. Regarding clickstreams, the set of M pre-defined actions, \( A = \{ a_1, \ldots, a_m, \ldots, a_M \} \), corresponds to the set of states of the Markov chain model. We define a dataset of \( N \) clickstreams as \( U \), where \( U = \{ \mathbf{u}_1, \ldots, \mathbf{u}_n, \ldots, \mathbf{u}_N \} \). A clickstream \( \mathbf{u}_n \) is an ordered sequence with \( n_a \) actions: \( \{ \mathbf{u}_{n,1}, \ldots, \mathbf{u}_{n,i}, \ldots, \mathbf{u}_{n,n_a} \} \), where \( \mathbf{u}_{n,i} \in A \). The set of \( K \) Markov chain models across the SOM grid points is represented as \( \Theta = \{ \theta_1, \ldots, \theta_k, \ldots, \theta_K \} \).

A Markov chain model is determined by its parameters, \( \theta_k = \{ \theta_k^1(u_{n,1} = a_m), \theta_k^2(u_{n,i} = a_t | u_{n,i−1} = a_m) \} \), where \( \theta_k^1(u_{n,1} = a_m) \) denotes the probability of a clickstream starting with an action \( a_m \) and \( \theta_k^2(u_{n,i} = a_t | u_{n,i−1} = a_m) \) denotes the transition probability of two consecutive actions \( a_m \) and \( a_t \).

The training method for SOM with Markov chain models is similar to that of training a conventional SOM with vector prototypes. Nevertheless, at each SOM training iteration, rather than searching the best matching data points and calculating a mean vector, the Expectation-Maximization (EM) algorithm [18] is applied to recover the optimal parameters of SOM with Markov chain models. Algorithm 1 gives an overview of the competitive learning algorithm for training SOM with Markov chain models.

Specifically speaking, at each iteration of the SOM training, the EM algorithm searches the domain of model parameters and updates \( \Theta \) in order to maximize the coupling likelihood \( \mathcal{L}_c(\Theta | U) \) [6], which measures how well the models fit the dataset.

\[
\mathcal{L}_c(\Theta | U) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \frac{1}{R} p_c(\mathbf{u}_n | \theta_k) \tag{2}
\]

\( p_c(\mathbf{u}_n | \theta_k) \) represents the coupling likelihood between a clickstream \( \mathbf{u}_n \) and a model \( \theta_k \). It is defined as the joint probability of \( \mathbf{u}_n \) fitting in \( \theta_k \) and models in its neighborhood, which is defined by \( h(k, r) \).

\[
p_c(\mathbf{u}_n | \theta_k) = p(\mathbf{u}_n | \theta_k) \prod_{r \neq k} p(\mathbf{u}_n | \theta_r)^{h(k, r)} \tag{3}
\]

\( p(\mathbf{u}_n | \theta_k) \) gives the probability of the \( n \)-th data object fitting in the model \( k \) with parameters \( \theta_k \).

\( \mathcal{Q}(\Theta; \Theta^{old}) \) is calculated in the last iteration. The posterior expectation of \( \mathcal{L}_c \), the so-called Q-function, is calculated as follows,

\[
Q(\Theta; \Theta^{old}) = \sum_{l=1}^{K} \sum_{n=1}^{N} \sum_{k=1}^{M} p_c(\mathbf{u}_n | \Theta^{old}) h(k, l) \log p(\mathbf{u}_n | \theta_k) \tag{5}
\]

- E-step: Calculate the posterior probability \( p_c(\mathbf{u}_n | \Theta^{old}) \) that gives the probability of the \( n \)-th data object fitting in the model \( k \) with parameters \( \Theta^{old} \) calculated in the last iteration. The posterior expectation of \( \mathcal{L}_c \), the so-called Q-function, is calculated as follows,

\[
\mathcal{Q}(\Theta; \Theta^{old}) = \sum_{l=1}^{K} \sum_{n=1}^{N} \sum_{k=1}^{M} p_c(\mathbf{u}_n | \Theta^{old}) h(k, l) \log p(\mathbf{u}_n | \theta_k) \tag{5}
\]

- M-step: Maximize the Q-function with respect to each subset of parameters \( \Theta \). The update rules for each set of parameters are shown below, and it guarantees to increase the coupling-likelihood \( \mathcal{L}_c \):

Initial state probability,

\[
\Theta_k^{l}(u_{n,1} = a_m) = \frac{\sum_{n=1}^{N} \sum_{k=1}^{M} p_c(\mathbf{u}_n | \Theta^{old}) h(k, l) \delta(u_{n,1}, a_m)}{\sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{l=1}^{K} p_c(\mathbf{u}_n | \Theta^{old}) h(k, l) \delta(u_{n,1}, a_r)} \tag{6}
\]

Transition probability,
Clickstreams are sequences of user actions, which are of various lengths. We encode each click action as a rectangle, and color each action differently. Thus, one clickstream is represented by a sequence of colored rectangles. Take Sell-Your-Item for example: a seller lists an item for sale on eBay and carries out a series of actions: “edit title, upload pictures, write description, set prices, select shipping methods, set other options”. The corresponding visualization is shown in Figure 3. In order to help users identify the most frequent behavior patterns, the size of a rectangle is proportional to the frequency of the clickstream pattern’s existence in the data set.

5.2 Clickstreams Placement

Considering that visual metaphors take up space unlike mapped data points, it would cause a serious overlapping problem if we naively place visual rectangles of all clickstreams where they are mapped (See Figure 5(a)). Although we can see the cluster structure in the visualization, it is impossible to tell the representative behavior patterns of each cluster because of the overlapping. Thus, the layout has to be adjusted to reduce visual clutter. In addition, it is unscaleable and unnecessary to display all clickstreams on a limited screen, especially when the data size is large. A proper placement strategy is expected to satisfy the following principles:

- Uncluttered, the clutter ought to be at a low level which doesn’t affect visual pattern perception;
- Consistent, the topological relations among the mapped clickstreams should be preserved;
- Representative, important clickstream patterns should be guaranteed to be presented.

We fully considered the above three principles during the placement strategy design. First, we move clickstreams to avoid overlapping as much as possible. Second, regarding the consistency principle, when moving a clickstream, we constrain the placement strategy design. First, we move clickstreams to avoid overlapping as much as possible. Second, regarding the consistency principle, when moving a clickstream, we constrain the placement in its surrounding area. Lastly, we employ a significance factor to measure the representativeness of each clickstream. For important clickstreams, we guarantee that they have higher priorities so that their existence, and $p(u_k | \theta_k)$ is the probability of $u_k$ fitting in model $k$.

$$s_{n,k} = f_n p(u_n | \theta_k)$$

where $f_n$ represents the frequency of a clickstream pattern $u_n$’s existence, and $p(u_n | \theta_k)$ is the probability of $u_n$ fitting in model $k$.

We define the associated significance of one clickstream as,

$$s_n = \max_k s_{n,k}$$

Figure 3: Visualization of an example clickstream from the Sell-Your-Item page on eBay website. Different colors denote different actions. In this example, a user first edits the title for the item to sell, uploads pictures, writes descriptions, and then sets the prices, shipping methods and other options.

Algorithm 2 Layout Generation by Randomized Greedy Algorithm

**Input:** clickstream rectangles $V = \{v_1, \ldots, v_N\}$, the corresponding mapped positions $P = \{p_1, \ldots, p_N\}$, the associated significance measure $S = \{s_1, \ldots, s_N\}$, and the flag signs $F = \{f_1, \ldots, f_N\}$ to indicate whether a clickstream is representative.

1. Sort rectangles according to $S$, and move the ordered representative clickstreams to the beginning of the list;
2. for each rectangle $v_n$ in the sorted list do
   3. while $v_n$ doesn’t reach the outer end of the spiral do
      4. Move $v_n$ a bit along the spiral path and try to place it;
      5. if $v_n$ doesn’t intersect with other placed rectangles then
         6. Place $v_n$;
         7. break;
      8. end if
   9. end while
10. if $v_n$ is representative, but didn’t find a placement then
11. Place $v_n$ at $p_n$;
12. end if
13. end for

5 Visualization

Although the clickstreams are successfully projected onto a 2D space after the data mapping step, creating a visualization that can clearly present the clickstream clusters is yet to be resolved. In this section, we address this problem by introducing a self-illustrative visual representation of clickstreams and an effective layout algorithm.

5.1 Visual Representation of Clickstreams

Clickstreams are sequences of user actions, which are of various lengths. We encode each click action as a rectangle, and color each action differently. Thus, one clickstream is represented by a sequence of colored rectangles. Take Sell-Your-Item for example: a seller lists an item for sale on eBay and carries out a series of actions: “edit title, upload pictures, write description, set prices, select shipping methods, set other options”. The corresponding visualization is shown in Figure 3. In order to help users identify the most frequent behavior patterns, the size of a rectangle is proportional to the frequency of the clickstream pattern’s existence in the data set.

$$p = \sum_{k=1}^{K} \theta_k p(u_k | \theta_k)$$

where $\beta(u_{n,j} = a_i | u_{n,j-1} = a_m)$ is an indicator function that equals to 1 if action $a_i$ follows right after action $a_m$ in the clickstream $u_n$ and 0 otherwise.

After the SOM with a set of Markov chain models is trained, each clickstream is then mapped to a 2D position $p_k$ determined by the coordinates of a model, $\theta_k(x_k, y_k)$ and the probabilities $p(u_k | \theta_k)$ of that clickstream fitting into the respective models,

$$p_n = \sum_{k=1}^{K} \theta_k p(u_k | \theta_k)$$

In other words, a clickstream is placed close to the models that it fits better.
The layout algorithm is illustrated in Algorithm 2. All clickstreams are first sorted in a descending order according to their maximum significance values across all models. Then, a randomized greedy algorithm is applied to place the clickstream rectangles. Every clickstream rectangle is trying to be placed along a spiral path starting at the clickstream’s mapped position, as shown in Figure 4. After a limited number of trials, the rectangle is either placed or discarded finally. However, a set of clickstreams that best fit each model based on the significance values are selected as representative clickstreams. They are guaranteed to be placed even if overlapping cannot be avoided. We use a Boolean sign $b$ to indicate whether one clickstream is representative, $B = \{b_1, \ldots, b_N\}$, for all clickstreams. A little clutter would not prohibit the patterns perception because human eyes are “low-pass” filters [4]. When similar data objects are grouped and form a significant pattern to attract users’ attention, they can discover the clustered patterns and ignore high-frequency “noise”.

This layout generation approach is straightforward, and the final visualization shows distinct patterns with little clutter. Additionally, we evaluate the completeness of placement of significant clickstreams by the factor $CoS = \frac{\sum \text{significance of the placed clickstreams}}{\sum \text{significance of all clickstreams}}$. Figure 5 shows the comparison between the visualizations before and after our placement method is applied to the Sell-Your-Item dataset. In Figure 5(b), the $CoS$ value is 93.2%, which means the majority of the clickstream patterns are displayed.

6 INTERACTIVE EXPLORATION

Interaction is the key to exploratory data analysis. It provides the opportunity for people to communicate with data. Our system enables examining details about one clickstream or any chosen group of clickstreams. Based on the visualization, analysts may want to divide the whole clickstream dataset into a number of clusters based on their perception. Our system supports interactive cluster analysis under the analysts’ supervision.

6.1 Data Profile Exploration

Visualization is a way of communicating messages from data. Analysts observe, interpret and make sense of what they see to understand the clickstreams. Figure 6 shows the system interface for data visualization and exploration. The clickstreams are visualized on the left side (Figure 6(a)) with the legends of actions displayed at the upper-right corner (Figure 6(b)). The analyst can move the mouse over an interesting clickstream pattern to check the data profile. For example, the clickstream “I” highlighted in Figure 6(a) is an interesting clickstream. Figure 6(c) shows the visual pattern of clickstream “I” and the existence frequency of this clickstream pattern in the dataset. Figure 6(d) presents the demographic information and statistical summary about the corresponding clickstream pattern by using histograms. The blue histograms indicate the statistical distribution of each data attribute for the specific clickstream pattern, while the background grey histograms show the overall distribution of the entire dataset. The analyst can also freely select a group of clickstreams (the highlighted group “G” in Figure 6(a)) to check the group statistical information. This intuitive exploration approach assists analysts to learn about data details from multiple aspects and at different scales.

6.2 Interactive Cluster Analysis

Although SOM projects clickstreams onto the 2D plane with similar data staying together, deciding which clickstreams belong to the same group still depends on domain knowledge. Since the displayed clickstreams are only representative samples, it is also necessary to support cluster analysis of the whole dataset in order to obtain a thorough statistical summary. We introduce an interactive cluster analysis approach, the semi-supervised K-means, to meet this need. The analyst can specify distinct groups of clickstreams on the visualization and then cluster the whole dataset using the specified groups. During the process of cluster analysis, the clickstreams are represented by their 2D mapped coordinates. Because the original inter-clickstream topological relations are preserved while data mapping, it is reasonable to use the 2D coordinates to cluster clickstreams.

People can easily perceive and verify clickstream patterns by using the provided visualization and interaction tools. We introduce an interactive cluster analysis method by combining the automatic K-means algorithm [8] and experts’ domain knowledge through interactive visualization. We utilize the analyst’s input as initialization and constraints in the K-means algorithm. Considering K-means has one drawback that it is only feasible for searching hyperspherical-shaped clusters [1, 3, 25], we adopt a centroid chain technique to deal with this problem. In our method, each cluster is represented by a centroid chain instead of a centroid as in the standard
The WCSS is defined as,

$$\text{WCSS} = \sum_{k=1}^{K} \sum_{p_n \in S_k} \text{Dist}(p_n, C_k)$$

(11)

where $S_k$ corresponds to the cluster set K and $p_n$ is a data sample within $S_k$. $C_k$ is the centroid chain of cluster K with $C_k = \{p_1, \ldots, p_m, \ldots, p_M\}$. The distance Dist$(p_n, C_k)$ between a data point and a centroid chain is defined as,

$$\text{Dist}(p_n, C_k) = \min_{p_m \in C_k} ||p_n - p_m||$$

(12)

In each iteration of the K-means algorithm, at the assignment step, the selected clickstreams by the analyst are assigned to the specified group, and the unselected data are assigned to its closest cluster. Meanwhile, each centroid $p_m$ along the centroid chain records a collection of data points that are most close to it. At the updating step, the centroid chain of each cluster is updated by taking the mean over $p_m$ and its associated collection of close points recorded at the assignment step. Details of our semi-supervised K-means algorithm are shown in Algorithm 3. The final clustering results not only are presented in the visualization immediately but also can be exported to files for further analysis.

7 Case Studies

Understanding how people use the website is critical to the success of an e-commerce business like eBay. As mentioned earlier, our work was motivated by the fact that analysts and product managers at eBay have difficulties in obtaining insights into user behavior patterns from the clickstream data. We invited some of them to use our system, explore data in their domains, and give feedback. In this section, we present our observations of how the system was used in two case studies.

7.1 Understanding Behavior Patterns on Sell-Your-Item Page

For the eBay marketplace, listing items for sale is the beginning of all seller activities. Thus, making the listing process intuitive and efficient is important. As described in Section 3, sellers are required to fill eight sections on the Sell-Your-Item page. From our collaboration with the product team we learned that the layout and ordering of these sections are critical to the website usability. Currently, the sections are laid out in a sequential order from top down, which was designed and evaluated by user experience designers based on user studies conducted in usability labs. They would like to understand how users interact with the page in real-world scenarios. The two analysts who were invited to this study would like to seek answers particularly to the following questions:

Do users fill all the information requested on the page?

Do they follow the pre-defined order to fill in information?
What are the scenarios when the answers to the questions above are "No"?

In preparing the data, we sampled from one day clickstream data on the Sell-Your-Item page of eBay US site. Each visit is a sequence of actions in terms of the sections they edit as described in Section 3. In order to answer the last question, the following selling related information is collected based on the analysts’ recommendations.

- **Seller segment**
- **User gender**
- **Years being an eBay user**
- **Selling category**

Figure 5 (b) shows the generated visualization. As the participants expected, most users follow the default ordering and fill most of the sections on the Sell-Your-Item page. Although no cluster with very distinct patterns stands out, the visualization effectively shows the variation in the data. The participants pointed out interesting behavior patterns as outlined in the figure. For example, the patterns included in the red box show that users start filling the page by uploading pictures rather than editing the title. There are also patterns in which certain actions are not performed. For example, in the green box users do not upload pictures, and those in the blue box do not write a description. The participants then would like to investigate in what scenarios such behaviors happen in order to infer potential causes. They conducted such analysis by selecting the patterns of interests one by one and investigating corresponding summary statistics.

First, the participants noticed in the visualization that a significant number of clickstreams start by uploading pictures rather than editing the title. They considered this interesting, because the title is one of the most critical parts of a listing on eBay, and the title input box is placed at the beginning of the page. By selecting these clickstreams, they can inspect related demographics and selling information on the side (See Figure 7). Please note that due to the paper space limitation, in Figure 7, Figure 8 and Figure 9, we choose to show only part of the visualization that contains the clickstreams of interest and the statistics summary panel for illustration purposes. The statistical information on the side reveals that such activities are more likely to happen in the Fashion category and less likely in the Tech category compared to the selling category distribution of the entire dataset. They considered this to be reasonable because pictures are generally more effective than text in describing clothes and shoes.

Next, the analysts investigated the scenario where users do not upload a picture (See Figure 8). It turned out that the majority of this behavior happens in the Media category, which includes books, CDs, DVDs, and etc. For such products, a pre-filled Sell-Your-Item page with standard product pictures are often provided by eBay, while others are not, e.g., clothes in the Fashion category and antiques in the Collectibles category. Therefore, most users use the provided product pictures instead of uploading their own. Before the study, the analysts thought that user behavior during listing could correlate with various characteristics of the users and listings. Therefore, they recommended that we include seller segments, gender, years being an eBay user, and selling category for study. These findings suggest that among these factors, the selling category is most correlated to user behavior.

The participants continue to examine other clickstream patterns that do not contain picture uploads. They noticed that a large number of users not only do not upload a picture, but also skip the description section (See Figure 9). For a listing on eBay, users are encouraged to write a detailed description of the items they are selling. The participants mentioned that previous studies showed that the quality of the description plays a critical role in the buyer’s purchase decision. These listings are also more likely to belong
to the Media category. However, the analysts observed that the distribution of years being an eBay user is different from that of the group selected in Figure 8. These users are less experienced. The analysts' intuition was that these inexperienced users have not learned the importance of descriptions in selling. Based on this observation, they considered providing more explicit messaging on the Sell-Your-Item page to encourage especially the inexperienced users to write a description.

Overall, the analysts from the selling team liked the system, which enabled them to explore the data interactively. They expected more diverse behavior patterns on the Sell-Your-Item page. After the study, they realized that most users followed the designed process, except several unique categories, e.g., Media and Fashion. This was a significant finding for them. They considered investigating the feasibility of providing customized listing processes for these categories.

7.2 Finding Factors That Correlate to Drop-offs in Shopping Cart

When users make purchases at eBay, items are first added into the shopping cart. Users can view their items in the cart, add more items, remove items, update item quantities and save items for future purchase. When users finally decide to buy the items in the cart, they proceed to checkout, which is considered a successful cart visit. When users end their visit without proceeding to checkout, we call it a drop-off. The performance of the cart is measured by the drop-off rate, which is defined as the percentage of drop-offs in all cart visits. The lower the drop-off rate, the better the shopping cart performs. As the shopping cart is considered the most critical step in the buying process, optimizing user experience in the cart and reducing drop-off rate will greatly benefit eBay. Furthermore, predicting cart drop-offs provides opportunities for eBay to prevent drop-offs and boost sales. For example, alternative merchandise that might suit users’ demands and incentives can be offered to encourage users to checkout when users’ intentions to drop-off are detected by eBay. Understanding factors that correlate with cart drop-offs is the key to all of the above. Knowing that we had developed the visual analytics system, an analyst and a production manager from the cart team reached out for help. Based on their definitions, a user behavior in the cart is composed of any of the 5 possible actions mentioned above. In addition, other factors, such as users’ buyer segment, gender, years being an eBay user, time of the day when the visit happens, and the number of items in the cart, are also captured based on their recommendations.

After a simple walkthrough demo of the system, we let them explore by themselves. Their goal was to investigate the correlation between cart drop-offs and various factors, including user behavior patterns in the clickstream data. Since the visualization provides a clear overview of the clickstream data, they immediately spotted groups of different patterns. They then started to circle these groups and investigate summary statistics of them. We noticed that the participants quickly went through many iterations of merging and splitting the groups. Finally, six groups were created, as illustrated in Figure 10. We can see that clickstreams in the same group have similar patterns. The overall drop-off rate varies among groups, which suggests a strong correlation between user behavior and cart drop-offs. The number of items in the cart also shows correlation with the behavior pattern and drop-off rate. Group F has the lowest drop-off rate. Most clickstreams in this group have only one item in the cart. As the overall drop-off rate increases, the distribution of the cart size shows a larger variance.

In Figure 10, irregular circles were drawn to group clickstreams that were not spatially close to each other, e.g., group B. In other words, our automatic data mapping algorithm did not consider them to be similar patterns. We asked the participants their reasons for such groupings and which criteria they used to evaluate similarity. They demonstrated their reasoning as illustrated in Figure 11. Our model-based SOM did a good job in mapping clickstreams in group B and those in group A together, because they both have patterns of alternating between View and Remove. The difference is that group B starts with an Add while group A does not. Clickstreams in group C also start with Add, but do not have the alternating pattern. However, when the analysts investigated the statistics of these groups, they found that the drop-off rate of group B was almost twice that of group A, but was close to group C. Based on their knowledge of the shopping cart, the participants knew that when users enter the shopping cart, their intentions are very different depending on whether they added an item before. Therefore, they chose to group B and C together. Such domain knowledge is hard to summarize and implement directly in the clustering algorithm. It is most effective to capture it through interacting with a visual analytics system.

Finally, our system performed a semi-supervised clustering based on the grouping in Figure 10, and the results are illustrated in Figure 12. Six clusters with different patterns were created. The results were exported and used by the analysts in further statistical analysis. Both participants appreciated the insights they obtained.
Figure 11: Interactive exploration of clickstream patterns in eBay shopping carts. The analyst groups similar patterns using the interactive lasso tool. The result shows strong correlations between user behavior and cart drop-offs. The histograms show corresponding statistical information about the clusters. Red bars denote drop-offs, and blue ones denote successful visits.

through the study. The analyst learned that user behavior and the number of items in the cart can be good features for predicting cart drop-offs. She said that this definitely will help her develop better prediction models. The production manager considered developing a user segmentation based on the clustering results.

7.3 Feedback

All the participants, including analysts and product managers, had little experience with information visualization besides basic charts. Therefore, we conducted short training sessions before they started to use the system in both studies. In the training, we explained the visual encoding of patterns and the layout. They were able to understand the visualization and identify clusters of similar patterns after the training. However, one product manager did mention that he would have difficulties interpreting the visualization by himself. One participant who considered the visualization intuitive said that the analogy between this and tag clouds helped. After the initial learning period, the participants all found the visualization very useful. One analyst said, “Being able to see all the user actions is so powerful. I now immediately know not only the most common behavior patterns but also the outliers.”

Figure 12: Final clustering results obtained by semi-supervised clustering based on input from the analysts. Different colors indicate different clusters. Six clusters were created, and each cluster of clickstreams showed its own distinct behavior patterns. The results were also exported and used by the analysts in further statistical analysis.

Most participants found the interactions of the system intuitive. Some of them even started to select groups of patterns by circling without being told. They really liked that they could investigate the summary statistics of the selected groups. One analyst said, “This totally enables us to correlate users’ behavior with their demographics. More important, I can do it interactively.” The interactivity of our system enables very effective data exploration and reasoning.

8 Conclusions and Future Work

We present a visual cluster exploration approach to analyze valuable web clickstream data. Our approach maps the heterogeneous clickstreams in a 2D space, visualizes representative data samples, and enables user-guided data exploration and clustering. The visual exploration approach helps analysts uncover user behavior patterns, learn the pattern demographics and make sense of the interrelationships between the patterns and their demographics. This knowledge will help managers make better business strategies, leading to better services. More importantly, our problem solving framework is not constrained to analyzing the web clickstream data. It can be extended to deal with a broader class of categorical sequences data in many other fields.

While our approach is effective, there are a few aspects to improve. For instance, the current visual encoding is suitable for clickstreams with a small number of predefined action options. When the number of action options increases to tens or hundreds, the visual encoding method should be reconsidered. Because there is always a mismatch between the ever increasing data size and the limited display screen space, we need to investigate how to take advantage of level-of-detail layout techniques to handle very large datasets.

Acknowledgements

The first author was a summer intern at eBay Research Labs while this work was done. This work has also been supported in part by the U.S. National Science Foundation through grants CCF-0811422, IIS-1147363, CCF-0808896, and CCF-1025269.