Clickstream Visualization Based on Usage Patterns

Srinidhi Kannappady, Sudhir P. Mudur, and Nematollaah Shiri

Dept. of Computer Science and Software Engineering
Concordia University, Montreal, Quebec, Canada
{s_kannap,mudur, shiri}@cse.concordia.ca

Abstract. Most clickstream visualization techniques display web users' clicks by highlighting paths in a graph of the underlying web site structure. These techniques do not scale to handle high volume web usage data. Further, historical usage data is not considered. The work described in this paper differs from other work in the following aspect. Fuzzy clustering is applied to historical usage data and the result imaged in the form of a point cloud. Web navigation data from active users are shown as animated paths in this point cloud. It is clear that when many paths get attracted to one of the clusters, that particular cluster is currently "hot." Further as sessions terminate, new sessions are incrementally incorporated into the point cloud. The complete process is closely coupled to the fuzzy clustering technique and makes effective use of clustering results. The method is demonstrated on a very large set of web log records consisting of over half a million page clicks.

1 Introduction

Web usage analysis of large and popular websites can provide vital information about online business transactions that can be used by business administrators to improve the services provided through their websites. Web usage data has normally been analyzed either in the form of user sessions or in the form of clickstream data. A session is typically the set of pages (URLS) visited by a user from the moment the user enters a web site to the moment the same user leaves it [19]. Clickstream is a generic term to describe visitors' paths through one or more web sites [13]. Analysis of clickstream data can show how visitors navigate and use the web site over time.

Visualization techniques are claimed to be among the best ways to analyze and understand web usage data [17]. Through visualization one can discover interesting patterns more easily than by looking at raw usage logs. In addition, there is also the possibility of generating recommendations from these patterns [18]. As user interests are not fixed and change over time, web usage data for a popular web site is very large, sparse, and fuzzy. The most basic way to visualize web usage data is by using the spanning tree technique to convert a log file into the users' browsing map. This technique is not robust and does not scale well enough to construct a users' browsing map when the web site is complex and the volume of clickstream data is large.

Our approach differs from earlier work in that we show dynamic web usage trends by overlaying clickstream data for every active user in the form of an animated

P. Kalra and S. Peleg (Eds.): ICVGIP 2006, LNCS 4338, pp. 339 – 351, 2006.

particle moving within a clustered visual representation of historical web usage data.Visual analysis of the paths followed by these active users can provide insight into the current online interests and trends. Our visualization process can be briefly described as follows: In the first phase, we create a three dimensional (3D) point cloud visual representation of historical web log data. For this, the large volume of web usage data available is first organized into sessions. Fuzzy clustering is then carried out on these sessions. This identifies a small number of the sessions as cluster centers. And for all other sessions, we get fuzzy membership values with respect to these clusters. Using a combination of Multi Dimensional Scaling (MDS) and Sammon Mapping (SM), the cluster centers are assigned positions in 3D space to optimally reflect the dissimilarity interrelationships amongst them. Next we render all the user sessions as a point cloud by direct use of the fuzzy membership values of these sessions with the cluster centers. This yields a 3D visual representation of the web usage pattern. The second phase involves real time visualization of clickstream data. An active user is defined as the one whose most recent web page visit was initiated within a period, say, 45 minutes. The web pages visited by each active user are maintained as a dynamically updated session. When an active session is updated by addition of the new web page he/she visited, the dissimilarities between this updated session and the current cluster centers are used to obtain the fuzzy membership values into the clusters; these values are then used to update the position of the session within the point cloud.

Given that a popular website will have a large number of new page clicks per second, the positions of active sessions are updated frequently. Periodically, as active sessions terminate (i.e., no new page visited by the active user within the last 45 minutes), we dynamically update the web usage profiles using an incremental version of the fuzzy clustering technique. When a session is incrementally added to the current clustering of the web usage data, it could be designated as a new cluster center or just be like other sessions with different membership values to the existing clusters. In the latter case, it is rendered in the usual manner into the point cloud. In the former case, we have devised an incremental version of Sammon Mapping which yields a new 3D position for this session. All subsequent visuals of user sessions and clickstream data then make use of the newly added cluster centers as well. We have experimented with our department's website with over 10,000 pages and web log data gathered over a period of a few months consisting of over half a million web page clicks. For simulating real time web page click, we divided the web log records into a historical data set and a click stream data set (the last 5,000 records).

The rest of this paper is organized as follows. Section 2 is a review of related work on web usage visualization. Section 3 briefly describes the fuzzy clustering technique and also the metric MDS and SM technique used for projecting high dimensional data into low dimension. In Section 4 we explain the process of rendering the web usage data as a point cloud with the active sessions animated as they are incrementally updated with web page click data. We also show results of our experiments on large web usage data. In section 5, we describe the use of incremental fuzzy clustering algorithm for updating the point cloud image to accommodate new usage sessions. Section 6 concludes with some observations and potential future work.

2 Related Work

Most web usage visualizations highlight paths traversed by users in a graph of the website structure. Hence results from research on the general problem of visualizing graph structures become applicable [23]. Further, the near hierarchical structures of the web make visualizing them slightly easier than visualizing a general graph. Cone trees [8], Hyperbolic tree maps [15], and Landscape [1] are typical examples of web structure visualization approaches. SiteLens, from Inxight Software (www.inxight.com), used the hyperbolic tree technique to visualize Web site structure, while NicheWorks [21] used an angular layout similar to disk trees [4]. In most of these visualizations, the main goal is to help users navigate more effectively by visually representing the non-linear information access structure. On the other hand, it is very important for web usage analysis to get insight into usage patterns, current interests, and trends based on web clicks, particularly for large websites with high volume usage. There is considerably less attention on clickstream visualization. Unfortunately, it is hard, if not impossible, to visualize sparse voluminous data of a large number of dimensions of numerous items in a workable manner, as comprehension decreases with the amount of data displayed [8].

WebQuilt [9] is a tool that uses a proxy server to log the user's clickstream. It uses directed graphs to construct a visualization of the user's browsing path. The thickness and the color of the arrows indicate the user's browsing behavior. The thicker arrows denote a more heavily traversed path, and darker arrows mean that more time is spent. Vividence Clickstream [22] and ClickViz [3] use a similar approach to visualize the user's click stream data. Some visualization tools use 3D or multidimensional graphics, which can incorporate more features in one graph. Examples of tools using this kind of technology include Disk tree [4], VISIP [6], Parallel Coordinate [10], and Scalable Framework [14]. However, none of these scale up to be able to handle clickstream data consisting of millions of records. Further, none of them keep any visual record of usage history, thus making it difficult to gage patterns and trends. Our work differs from all the above mainly in the following aspect: we closely couple our visualization technique with a data mining technique that discovers usage patterns in the form of user profiles and then animates active users' click data by overlaying it on a point cloud rendering of clustered historical usage data.

3 Fuzzy Clustering and Dimensionality Reduction

In what follows, we describe the techniques of fuzzy clustering, dimensionality reduction, and graphic mapping used in this work.

3.1 Relational Fuzzy Subtractive Clustering (RFSC)

For discovering usage patterns, we have used Relational Fuzzy Subtractive Clustering algorithm (RFSC) [19]. We have chosen this over other fuzzy clustering techniques for its distinct advantages, namely scalability to large usage data, efficiency, ability to

handle noise, and most importantly the existence of an incremental version [20] which we use to maintain up to date usage profiles. RFSC works on web log data organized into sessions and dissimilarity values between sessions defined using the measure given in [16]. We briefly describe the core algorithm to provide a flavor of the RFSC technique. A detailed exposition can be found in [19, 20].

The RFSC algorithm starts by considering each session x_i as a potential cluster center. The potential P_i of each session x_i is calculated as follows:

$$P_i = \sum_{j=1}^{N_U} e^{-\alpha R_{ij}^2}, \text{ where } \alpha = 4/\gamma^2$$

R_{ij} is the dissimilarity between sessions x_i and x_j, N_U is the total number of objects to be clustered, and γ is essentially the neighborhood calculated from the relational matrix R. It also true that $R_{ij} \geq 0$, $R_{ij}=R_{ji}$, and $R_{ii}=0$.

The session with highest potential (P_1^*) is selected as the first cluster center. Next, the potential of every other session is reduced proportional to the degree of similarity with this previous cluster center. Thus there is larger subtraction in potential of sessions that are closer to this cluster center compared to those which are farther away. After this subtractive step, a session (x_t) with the next highest potential (P_t) is selected as the next candidate cluster center. Now to determine whether this can be accepted as an actual cluster center or not, two threshold values are used, $\overline{\in}$ (accept ratio) and $\underline{\in}$ (reject ratio), where we have that $0 < \overline{\in}, \underline{\in} < 1$, and $\underline{\in} < \overline{\in}$. If $P_t > \overline{\in} P_1^*$, then x_t is selected as the next cluster center, and this is followed by the subtractive step described above. If $P_t < \underline{\in} P_1^*$, then x_t is rejected, and the clustering algorithm terminates. If the potential P_t lies between $\overline{\in} P_1^*$ and $\underline{\in} P_1^*$, then we say that potential has fallen in the gray region, in which case we check whether x_t provides a good trade-off between having a sufficient potential and being sufficiently far from existing cluster centers. If this holds, then x_t is selected as the next cluster center. This process of subtraction and selection continues until $P_t < \underline{\in} P_1^*$, which is the termination condition. After finding C cluster centers, the membership degree of different x_j to each cluster c_i is calculated using the formula: $u_{ij} = e^{-\alpha R_{c_i j}^2}$, i = [1..C] & j = [1..$N_U$], in which $R_{c_i j}$ is the dissimilarity of the ith cluster center x_{c_i} with the jth session x_j. When $x_j = x_{c_i}$, we have $R_{c_i j} = 0$ and that the membership $u_{ij} = 1$. When $x_j = x_{c_i}$, we have $R_{c_i j} = 0$ and the membership $u_{ij} = 1$. While most other fuzzy clustering algorithms impose the condition $\sum_{i=1}^{C} u_{ij} = 1$, RFSC does not. This effectively makes RFSC algorithm less sensitive to noise. Noise sessions are easily identified as their membership values will always lie on the asymptote of each of the clusters.

We have used a point cloud representation in 3D for visualizing usage patterns. Hence, in the next step we need to assign 3D coordinates to each of the sessions. For this, we make use of a dimensionality reduction technique. However, as we shall soon

see, our experimental web log data of over half a million page clicks gets organized into 64,529 sessions. Dimensionality reduction for this large data can be computationally prohibitive, as it would involve Eigen value analysis using a matrix of this size iteratively. In particular, convergence can be a major problem as there are many near equal dissimilarity values in the dataset. Fortunately, RFSC provides us with a much smaller number of cluster centers. Therefore, we first map these cluster centers into 3D and then use the fuzzy membership values to render the rest of the sessions.

3.2 Metric Multidimensional Scaling

Metric MDS begins with an n×n dissimilarity matrix R with elements r_{ij}, where $1 \leq i,j \leq n$. The objective of metric MDS is to find a configuration of points in p-dimensional space (p=3, in our case) from the dissimilarities between the data points such that the coordinates of the n points in p dimensions yield an Euclidean distance matrix whose elements are as close as possible to the elements of R. Using the metric MDS, we obtain the initial configuration. Since this is quite standard, we refer the reader to [21] for details. However, fidelity to the original distance relationship is poor due to low dimensional projection. To minimize this loss, we use Sammon Mapping, with suitable modifications to be able to handle the special characteristics of web usage data, described next.

3.3 Sammon Mapping

Sammon Mapping (SM) [17] is an unsupervised, nonlinear method that tries to preserve relative distances. The algorithm that generates a Sammon map employs a nonlinear transformation of the observed distances among data items when mapping data items from a high-dimensional space onto a low-dimensional space. Let r^*_{ij} denote the dissimilarity (usually Euclidean distance) between two different data items i and j in the original dimensional space, and r_{ij} denote the distance in the required projected space. Then the error function of SM is defined as follows:

$$E = \frac{1}{\sum_{i=1}^{n}\sum_{j=i+1}^{n} r^*_{ij}} \sum_{i=1}^{n}\sum_{j=i+1}^{n} \frac{(r^*_{ij} - r_{ij})^2}{r^*_{ij}} \tag{1}$$

Here, smaller the error value E, the better is the map we obtain. However, in practice, we are often unlikely to obtain perfect maps especially when the dataset is large and in high-dimensional space. Therefore, approximate preservation is what we can expect.

Let $E(m)$ be the mapping error after the iteration step m, i.e.,

$$E(m) = (1/c)\sum_{i<j}^{n} [r^*_{ij} - r_{ij}(m)]^2 / r^*_{ij} \tag{2}$$

where $c = \sum_{i<j}^{n} r^*_{ij}$, $r_{ij} = \sqrt{\sum_{k=1}^{p} [y_{ik}(m) - y_{jk}(m)]^2}$, and r^*_{ij} is the original distance matrix.

The new *d*-space configuration at iteration step *m*+1 is given by:

$$\mathbf{y_{pq}}\,(m+1) = \mathbf{y_{pq}}(m) - (\mathrm{MF}) \times \Delta_{pq}(m) \tag{3}$$

where $\Delta_{pq}(m) = \dfrac{\partial E(m)}{\partial y_{pq}(m)} \Bigg/ \left| \dfrac{\partial^2 E(m)}{\partial y_{pq}(m)^2} \right|$

and MF is the "magic factor" determined empirically to be about 0.3 or 0.4.

$$\frac{\partial E(m)}{\partial y_{pq}} = \frac{-2}{c} \sum_{j=1}^{n} \left[\frac{r_{pj}^{*} - r_{pj}}{r_{pj} r_{pj}^{*}} \right] (y_{pq} - y_{jq}) \tag{4}$$

$$\frac{\partial^2 E}{\partial y_{pq}^{2}} = \frac{-2}{c} \sum_{j=1}^{n} \frac{1}{r_{pj}^{*} r_{pj}} \left[\left(r_{pj}^{*} - r_{pj} \right) - \frac{(y_{pq} - y_{jq})^2}{r_{pj}} \left(1 + \frac{r_{pj}^{*} - r_{pj}}{r_{pj}} \right) \right] \tag{5}$$

This is an iterative process which terminates when the Sammon stress value *E* cannot be decreased anymore. The guidelines for best stress values suggested by Kruskal [12] are given in the following table:

Table 1. Stress guidelines suggested by Kruskal [12]

Stress	0.3	0.2	0.1	0.025	0.0
Goodness of fit	Poor	Fair	Good	Excellent	Perfect

3.3.1 Modified Sammon Mapping

We can observe that if any two points in the *d*-space have identical values, then the Sammon stress *E* will go beyond 1, which is not desirable. When going through the Sammon Mapping iterations for web usage data, we observed that quite often, the distance between some pair of clusters reaches close to zero, thus blowing up the stress value disproportionately. To overcome this problem, we modified equations (4) and (5) above so that even though the *d*-space has identical values, the stress *E* does not blow up. This is done by observing that r_{pj} in the denominator of these equations essentially provides a scale factor which can be avoided. The corresponding modified equations are as follows:

$$\frac{\partial E(m)}{\partial y_{pq}} = \frac{-2}{c} \sum_{j=1}^{n} \left[\frac{r_{pj}^{*} - r_{pj}}{r_{pj}^{*}} \right] (y_{pq} - y_{jq}) \tag{6}$$

$$\frac{\partial^2 E}{\partial y_{pq}^{2}} = \frac{-2}{c} \sum_{j=1}^{n} \frac{1}{r_{pj}^{*}} \left[\left(r_{pj}^{*} - r_{pj} \right) - (y_{pq} - y_{jq})^2 \left(1 + r_{pj}^{*} - r_{pj} \right) \right] \tag{7}$$

If we consider the mapping error in equation (5), we note that it is not necessary to maintain *c* in equations (6) and (7) for a successful solution of the optimization problem, since minimization of $(1/c)\sum_{i<j}^{n} [r_{ij}^{*} - r_{ij}(m)]^2 / r_{ij}^{*}$ and $\sum_{i<j}^{n} [r_{ij}^{*} - r_{ij}(m)]^2 / r_{ij}^{*}$ yield the same result.

We tested the modified SM algorithm using some benchmark datasets such as *iris* and *wine* [2] and confirmed that the results were the same as those obtained using the original SM algorithm. We use this modified SM algorithm in our work.

4 Rendering Fuzzy Clustering of Usage Sessions

We have chosen a simple 3D point cloud visual representation for reflecting the web usage patterns discovered by RFSC. Keeping in mind scalability requirements given the huge volume of web usage data, its sparseness, the inherent fuzziness and noise, and our need for dynamic update to handle clickstream data in real time, we feel that a point cloud, though very simple, is quite adequate. This was based on the following observations.

Choice of 3D over 2D: When clustering web usage sessions, the number of clusters for large data could lie in the range of few hundreds. The added dimension of "depth" in 3D provides the ability to better reflect the distance relationships. With the current trends in 3D graphics hardware, it becomes possible to use a simple metaphor of navigating in space and looking around a collection of clusters (clouds, in our case) to visually inspect the dataset and gain more insight.

Choice of Point cloud: The point cloud can easily handle the fuzziness captured by the clustering technique and visually depict this fuzziness with considerable fidelity.

Scalability of the Visual Mapping Technique: Sessions represented as particles in 3D space is a simple mapping computation and intensity can be varied to reflect closeness of association with a cluster. Large volumes of data can be handled efficiently and more importantly without undue computational overhead.

Noise Visualization: Noise sessions are easily detected in RFSC, as their membership values lie on the asymptotes of each of the clusters. Noise sessions are therefore assigned random positions in the 3D visual space. The effectiveness of this visual mapping is discussed in [11].

Close Integration with Clustering Method: Lastly, it was desired to have a simple method integral to clustering, so that one could show the navigational path in real time to help the web administrator get insight into current trends and interests.

We have the 3D positions of cluster centers from MDS and modified SM. Every point in the dataset has a membership value with every cluster center, which we use to assign 3D positions. The method is simple and described below.

4.1 Assigning 3D Position to Click Data Received from Active User Sessions

Every session (other than cluster centers and noise) is classified into one of following categories:

i) The first category consists of sessions having large affinity towards only one cluster center. The sessions that belong to this category will have one high membership value and all other membership values will be much lower.

ii) In the second category, sessions will have high affinity towards two cluster centers, and much lower membership values for all other clusters.
iii) In the third category, sessions will have high affinity towards three cluster centers, and much lower membership values for all other clusters.
iv) All other sessions are treated as noise.

For each session, cluster centers are addressed in the order of their membership values (m_1, m_2, m_3 …) say, C_1, C_2, C_3, etc. Let a be the average distance between clusters, and R be any random 3D vector.

Steps for rendering sessions that belong to the first category:

1) We consider Polar coordinates, i.e., (r, Θ, Φ), where the radius $r = 0.3a(1 - m_1)$.
2) The values Θ, Φ are chosen randomly to account for the fuzziness.
3) Then we convert these spherical coordinates to Cartesian coordinates, which gives a position (dx, dy, dz) in 3D space relative to the position of C_1.
4) These points are assigned full intensity.

Steps for rendering sessions that belong to the second category:

1) Multiply vector difference, $C_2 - C_1$ by $(1 - m_1)$ to get the vector P.
2) Carry out cross product of $C_2 - C_1$ with random vector R to get vector N.
3) Multiply vector N with $0.5a(1 - m_2)$.
4) Obtain the desired point coordinates by adding the vectors C_1, P, and N.
5) Lastly, assign intensity values reduced in proportion to the distance from the cluster centre.

Steps for rendering sessions that belong to the third category:

1) Multiply vector difference, $C_2 - C_1$ by $(1 - m_1)$ to get the vector P.
2) Carry out cross product of $C_2 - C_1$ with random vector R to get vector N.
3) Multiply vector N by $0.5a(1 - m_2)$.
4) Obtain point coordinates by adding the vectors C_1, P, and N.
5) Follow steps 1 to 4 for the first and third cluster centers.
6) Take the weighted average of the two points (step 4) to get the final point.
7) Assign intensity values reduced in proportion to the summed distances from the cluster centers.

The above procedure yields a computationally efficient method for assigning 3D positions to sessions. Use of dominant membership values results in preserving the inherent relationships much better. We have used the user access logs from our department server during January 15, 2004 to May 5, 2004. This file is cleaned in a preprocessing phase, organized into session data and then a relational data (dissimilarity) matrix is computed using all but the last 5000 log records. This relational matrix is then input to the RFSC algorithm. The total number of user sessions obtained was 64,529 and the number of cluster centers identified by RFSC was 46. Dissimilarity values between cluster centers are extracted from the relational data matrix and used as the input to the dimensionality reduction technique (partially

shown in left table in Fig. 1). The result of applying MDS to this dissimilarity matrix is shown in the middle table in Fig. 1. The reader may note some of the zero distances, illustrating the importance of the proposed modification to Sammon Mapping for this kind of data. The Sammon stress value *E* obtained for this dataset using our method was 0.11 and the much improved result is shown in the right most table in Fig. 1.

	C#1	C#2	C#3	C#4	C#5	C#6
C#1	0.0	1.0	1.0	1.0	1.0	1.0
C#2	1.0	0.0	1.0	1.0	1.0	1.0
C#3	1.0	1.0	0.0	1.0	0.889	1.0
C#4	1.0	1.0	1.0	0.0	1.0	1.0
C#5	1.0	0.889	1.0	1.0	0.0	1.0
C#6	1.0	1.0	1.0	1.0	1.0	0.0

	C#1	C#2	C#3	C#4	C#5	C#6
C#1	0.0	0.0	0.086	0.342	0.167	0.158
C#2	0.0	0.0	0.086	0.342	0.167	0.158
C#3	0.086	0.086	0.0	0.324	0.109	0.232
C#4	0.342	0.342	0.342	0.0	0.420	0.232
C#5	0.167	0.167	0.109	0.420	0.0	0.310
C#6	0.158	0.158	0.232	0.379	0.310	0.0

	C#1	C#2	C#3	C#4	C#5	C#6
C#1	0.0	0.291	0.321	0.545	0.644	0.587
C#2	0.291	0.0	0.252	0.523	0.845	0.718
C#3	0.321	0.252	0.0	0.725	0.841	0.752
C#4	0.545	0.523	0.725	0.0	0.702	0.532
C#5	0.644	0.845	0.841	0.702	0.0	0.226
C#6	0.587	0.718	0.752	0.532	0.226	0.0

Fig. 1. Dissimilarity values: original (left), after MDS (middle), after Sammon Mapping (right)

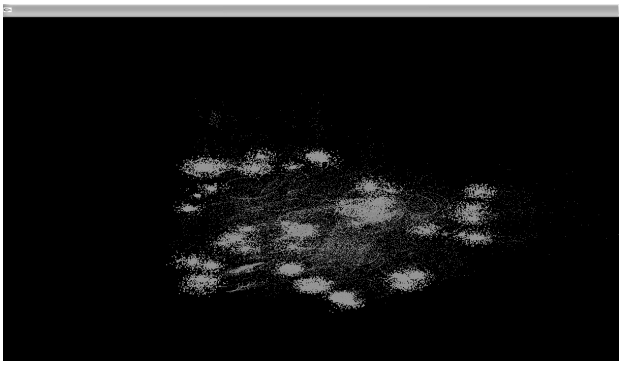

Fig. 2a. Point cloud image of fuzzy clustering of web usage sessions

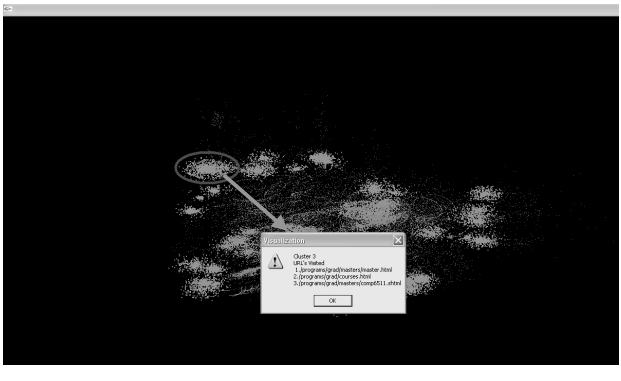

Fig. 2b. Representation of profile details of clicked cluster

The time needed to determine the coordinate values using our method was around 1 hour. This was because the distances between any pair of these 46 cluster centers were almost the same (close to 1), as can be seen in the left table in Fig. 1. In comparison, the time it took to find the coordinates for the *iris* dataset with 150 elements was less than a second since the distances were far more distinct. Fig. 2a shows a point cloud visual representing this usage data. Clicking on any point on this image will yield the preferences of the associated user profile (Fig. 2b).

4.2 Animating Clickstream Data Received from Active User Sessions

As mentioned earlier, once we have the historical web usage data imaged as a point cloud, we consider the currently active sessions. As each active user navigates through different web pages in the website, we animate this as a linear path in 3D overlaid on the 3D point cloud model. This is done as follows: The web page clicks are retrieved every frame and analyzed. This could create a new active session or update the pages visited by an active user session. We first calculate the session dissimilarity of each updated active user session with the current cluster centers. Then we obtain the fuzzy memberships with all existing cluster centers. Lastly, we assign a new position to each updated user session by the method described in section 4 and render it in a distinct color.

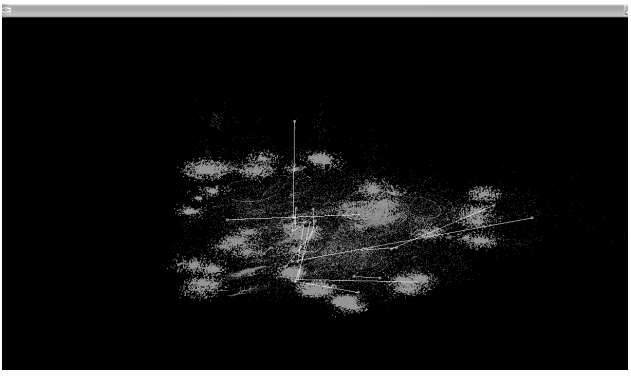

Fig. 3. Active user path visualization from clickstream data

For experimental purposes, we analyzed the last 5000 records of our web log and extracted the updates to active user sessions in an incremental fashion. Fig. 3 shows a screenshot which illustrates a sample of active user sessions as darker dots and the paths followed until this time. This type of animation enables one to visually get deeper insight into current trends and interests. For example, when an existing cluster is the attractor for many active user paths, it indicates that this is a “hot” usage profile.

5 Incremental RFSC and Visualization

Over a period of time, active sessions are terminated; in our case when there is no activity from that user for 45 minutes. The paths of terminated active sessions are removed from the display. However, as the number of such sessions increases, for correct visual depiction of the web site usage history, these must be reflected in the point cloud. For high volume usage, the over head is quite high if we have to carry out complete reclustering of the entire usage data (old clustered sessions plus newly completed sessions). Instead, we make use of the Incremental RFSC algorithm [20]. Whenever a new session is added, this algorithm either makes it a new cluster center or assigns fuzzy membership values to existing clusters.

As new clusters are discovered, it becomes essential to add the new clusters into the point cloud without changing the position of the existing clusters, to avoid any visual confusion to the viewer. We have again devised a method for plotting the new cluster without having to run the MDS and SM methods for the whole data again. We first obtain an initial coordinate value using the distance between the new cluster center and the existing cluster centers. Then we use the SM method described in section 3 to decrease the error in distance between the newly found cluster center and the existing cluster centers. When the Sammon stress goes beyond a pre-defined threshold, we need to perform the MDS and SM for the entire data set. We have experimented by removing the sessions belonging to a cluster and then found that Incremental RFSC does add that new cluster and this method assigns a new 3D position to the cluster center, sufficiently distinguishable from the rest.

6 Conclusions and Future Work

Historical data of web usage must be used in any visualization of clickstream data, if the web administrators have to gain insight into changes in trends and interests over time. Web usage data is however, very large, sparse, noisy, non-Euclidean and fuzzily classified, making its visualization a difficult task. In this paper, we have proposed using a combination of techniques: (i) RFSC for fuzzy clustering, (ii) a combination of Multidimensional Scaling followed by modified Sammon Mapping, we introduced, for dimensionality reduction to enable point cloud like visual rendering of the usage data, and (iii) incremental RFSC for continued update of the point cloud and (iv) animation of active user paths to get insight into trends and interests. By cleverly using the membership values assigned by RFSC to the other sessions, we developed a fast method for rendering the large data.

Future work is primarily on improvements to the current technique. First, we plan to provide another window which displays the structure of the website and highlights any usage profile, selected by user clicking on the point cloud. This will obviate the need for the message box which we currently display over the point cloud. Second, we plan to provide interrogation facility in the form of “if then” queries. For example, the web administrator can change the structure by editing one or more links, and the

system would react by illustrating the effect of this change on usage profiles, for instance, in terms of the number of links to be traversed.

Acknowledgements

This work was supported in part by NSERC (Canada) and ENCS, Concordia University. We thank B.S. Suryavanshi for his contribution in the development of the RFSC algorithm used in this work. We also thank the CSE Department at Concordia for providing us with the web log records used as part of our experiments.

References

1. Andrews, K.: Visualizing Cyberspace: information visualization in the harmony internet browser. In Proc. 1st IEEE Symp. On Information Visualization, (1995), pp. 90-96.
2. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. (1998)
3. Brainerd, J., Becker, B.: Case Study: E-commerce Clickstream Visualization. In Proc. of the IEEE Symp. On Information Visualization, (2001), pp. 153-156.
4. Chi, E.H.: Improving Web Usability through Visualization. Internet Computing, 6(2), (2002), pp. 64-71.
5. Chi, E.H.: WebSpace Visualizations. In Proc. 2nd Int'l World Wide Consortium (W3C), IEEE Internet Computing, 6(2), (1994), pp. 64-71.
6. Cugini, J., Scholtz, J.: VISIP: 3D Visualization of Paths through Websites. In Proc. Int'l workshop on Web-Based Information Visualization, Florence, Italy, (1999), pp. 259-263.
7. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, (2000).
8. Herman, I., Melancon, G., Marshall, M.S.: Graph Visualization and Navigation in Information Visualization: a survey. IEEE TVCG, 6(1), (2000), pp. 24-43.
9. Hong, J.I. Landay, J.A.: WebQuilt: A Framework for Capturing and Visualizing the Web Experience, In Proc. 10th Int'l World Wide Web Conference, Hong Kong, China, (2001), pp. 717-724.
10. Inselberg, A., Dimsdale, B.: Parallel coordinates: A tool for visualizing multi-dimensional geometry. In Proc. Visualization 90, San Francisco, CA, USA, (1999), pp. 361-370.
11. Kannappady, S., Mudur, S.P., Shiri, N.: Visualization of Web Usage Patterns. In Proc. 10th Int'l Database Engineering & Applications Symposium (IDEAS), New Delhi, India, (2006).
12. Kruskal, J.B.: Multidimensional Scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 29(1), (1964), pp. 1-27.
13. Lee, J., Podlaseck, M., Schonberg, E., Hoch, R.: Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising. Int'l Journal of Data Mining and Knowledge Discovery, 5(1), (20001), Kluwer Academic Publishers.
14. Lopez, N., Kreuseler, Schumann, H.: A scalable framework for information visualization, Trans. on Visualization and Computer Graphics, (2001).
15. Munzner, T.: Drawing Large Graphs with H3Viewer and Site Manager. In Proc. Graph Drawing 98, Springer-Verlag, New York, (1998), pp. 384-393.
16. Nasraoui, O., Krishnapuram, R., Joshi, A., Kamdar, T.: Automatic Web User Profiling and Personalization using Robust Fuzzy Relational Clustering, in E-commerce and Intelligent Methods Ed., Springer-Verlag, (2002).

17. Sammon, J.W. Jr.: A non-linear mapping for data structure analysis, IEEE Trans. on Computers, 18, (1969), pp. 401-409.
18. Simonson, J., Fuller, G., Tiwari, A.: A Survey of Web History Data Analysis and Visualization, In http://www.math.grinnell.edu/~lindseyd/ResearchState.html
19. Suryavanshi, B.S., Shiri, N., Mudur, S.P.: An Efficient Technique for Mining Usage Profiles Using Relation Fuzzy Subtractive Clustering. In Proc. Int'l workshop on Challenges in Web Information retrieval and Integration, (2005), pp.23-29.
20. Suryavanshi, B.S., Shiri, N., Mudur, S.P.: Incremental Relational Fuzzy Subtractive Clustering for Dynamic Web Usage Profiling. In Proc. WEBKDD Workshop on Training Evolving, Expanding and Multi-faceted Web Clickstreams, Chicago, Illinois, USA, (2005).
21. Trevor, F.C., Michael A.A.C.: Multidimensional Scaling, 2nd Edition, Chapman & Hall/CRC, (2001).
22. Vivince Clickstreams, In http://www.vivdince.com/resources/public/solutions/demo/demo-print.htm
23. Wills, G.J.: Nicheworks-Interactive Visualization of Very Large Graphs. In proc. Graph Drawing 97, Lecture Notes in Computer Science, Springer-Verlag, Berlin, (1997).