Separating the Swarm: Categorization Methods for User Sessions on the Web

Jeffrey Heer, Ed H. Chi
Xerox Palo Alto Research Center
Palo Alto CA, 94304 USA
{jheer,echi}@parc.xerox.com

ABSTRACT

Understanding user behaviors on Web sites enables site owners to make sites more usable, ultimately helping users to achieve their goals more quickly. Accordingly, researchers have devised methods for categorizing user sessions in hopes of revealing user interests. These techniques build user profiles by combining users' navigation paths with other data features, such as page viewing time, hyperlink structure, and page content. Previously, we have presented complex techniques of combining many of these data features to cluster user profiles. In this paper, we introduce a user study and a systematic evaluation of these different data features and their associated weighting schemes. We present the results of our study, including accuracy measures for a number of clustering approaches, and offer recommendations for Web analysts. While further investigation over more sites is needed to definitively settle on a robust scheme, we have characterized this analytic space.

Keywords

User Profile, User Categorization, User Patterns, Web Mining, Data Mining, Clustering, Classification, World Wide Web, User Study.

INTRODUCTION

Identification of user interests on the Web has many different applications. Webmasters and content producers would like to gain an understanding of the people that are visiting their Web sites in order to better tailor sites to user needs. Marketers would like to know user interests in order to have better sale promotions and advertisement placements. News organizations would like to produce and present materials that are highly relevant to their visitors.

By now, owners of Web sites realize that the usability of their site can greatly determine the success of their business [10]. Identifying and understanding the reason for user visits could enable site owners to tailor their site better to these user needs. For example, this information could help Webmasters prioritize the navigational paths of the static content pages to optimize for more common tasks [21]. Alternatively, they could use this information to personalize content for their users [2]. Server performance experts could use this information to enhance server performance by determining which features are used the most often. The aim is to make the site *stickier* so that users stay longer because of enhanced experiences.

One way to discover user interests is through user surveys and contextual inquiries. However, these methods tend to be tedious and expensive. One promising automated approach for inferring user interests is to analyze the Web server logs and cluster the user sessions. A number of clustering approaches have been proposed which employ limited combinations of different data features, such as the order of the pages viewed, page viewing time, and site structure. While the specific techniques vary, the end goal is the same: to create groupings of user sessions that accurately categorize the sessions according to the users' information needs.

A major problem in applying these techniques is that there has been no systematic evaluation of the approaches taken: (a) We don't know how well each of these data features contributes to the clustering process in real world situations, because each clustering paper describes case studies using different data sets. (b) There is no way to know how accurate these findings are, because without knowing *a priori* what the users' tasks and information needs are, we are incapable of determining whether the technique correctly clustered these user sessions into good groupings.

In order to do an effective evaluation of the clusterings, we need user sessions for which we know the associated information goals, enabling us to evaluate whether the clustering algorithms correctly categorized the user sessions into proper groups. In this paper, we present a user study and a systematic evaluation of clustering techniques using these different data features (modalities) and associated weighting schemes. We first asked 21 users to surf a given site with specific tasks. This allows us to know *a priori* what the user information needs of the users are. We then use this *a priori* knowledge to evaluate the different clustering schemes and extract useful guidelines for Web usage analysis.

The rest of the paper is organized as follows: First, we present related work on approaches to clustering user

CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA.

sessions. Next, we describe our multi-modal clustering algorithm. In the heart of the paper, we present the method for our user experiment, and then use the results to validate and analyze the different clustering schemes. Finally, we present an up-to-date analysis of a large corporate Web log as a case study to show the scalability of the method. We finish with some concluding remarks.

RELATED WORK

Web usage mining has been a hot research topic in the last several years due to its implications for businesses on the Web. In addition to research publications, several workshops have been organized to discuss the issues surrounding the various stages of Web usage mining [15,20].

The first step in Web usage mining is always data cleaning. The most significant problems in this step usually involve the extraction of Web usage sessions [12, 18]. We apply a set of standard techniques for sessionization as described in [12]. Another problem in this step is the amount of noise in the Web logs. Many paths in the log are not significantly interesting paths, e.g. single visits to the home page. To reduce noise and to compress the information in these logs, we sometimes apply the Longest Repeated Subsequence (LRS) algorithm [14]. A longest repeating subsequence (LRS) is a sequence of items where (1) subsequence means a set of consecutive items, (2) repeated means the item occurs more than some threshold *T*, where *T* typically equals one, and (3) longest means that although a subsequence may be part of another repeated subsequence, there is at least one occurrence of this subsequence where this is the longest repeating. Pitkow et. al. showed that this algorithm compresses and extracts out the significant top 10% of all paths while retaining 90% of the predictive power of the full data set [14].

After data cleaning, we can use the sessions to discover usage patterns. There are more than 30 commercially available applications that do Web log analysis [22]. At first, many of these tools were considered slow and inflexible, and most only offer low conceptual descriptive statistics [19, 22]. Data mining algorithms have recently been applied to the user sessions to discover higher-level trends. For example, researchers have applied these algorithms to discover which pages are often accessed together by doing sequential pattern, frequent itemset, or association analysis [3]. These techniques have been helpful in personalization applications [21], and Web caching and prefetching [11].

A more recent development in these analysis tools is to offer basic summarization by grouping user actions into activities [7, 1, 4, 22, 17], such as reading bulletin board messages, finding product information, or searching for technical support.

Shahabi et. al. describes a prototype system that uses viewing time as the primary feature to describe a user session [17]. Then, using a similarity measure roughly based on inner products, they cluster the sessions using KMeans clustering [9]. The system is evaluated on a fictional 34-page site with simulated path data, showing that the method has an error rate of 10-27%.

Zaiane et. al. proposed the application of On-Line Analytical Processing (OLAP) techniques to Web logs [22]. Their proposed approach could potentially generate acceptable groupings of user activities, but results were not reported. Moreover, the data features they considered in the OLAP process consisted only of information in the logs, with no page content or link structure information.

Fu et. al. suggested using the URLs to construct a page hierarchy which is used to categorize the pages [4]. For example, all pages under /authors/index.html would be classified as ‘authors’ pages. The page accesses in each user session are then described using these page categorizations. This is called ‘Generalization-based Clustering”, and is similar to using URL tokens (tokenize URLs on ‘/’ and other delimiters). Unfortunately, this approach only works if the URLs contain useful tokens, or if page categorization can be determined ahead of time manually. An algorithm called BIRCH is then used to categorize the user sessions. They evaluated this algorithm’s scalability on the University of Missouri-Rolla’s Web logs, but they didn’t do an accuracy evaluation on the resulting clusters.

Banerjee et. al. utilized the combination of time spent on a page and Longest Common Subsequences (LCS) to cluster the user sessions [1]. The LCS algorithm is first applied on all pairs of user sessions. Then each LCS path is reduced using page hierarchy in a generalization-based approach called ‘Concept-based Clustering’. This is basically a simpler form of Generalization-based Clustering, because they only use the top-most level of the page hierarchy to categorize the pages. Then similarities between LCS paths are computed as a function of the viewing time spent at each stage in the paths. A graph-partitioning algorithm called *Metis* is used to cluster these user sessions. They clustered about 23,000 user sessions, but reported only anecdotal evidence of effectiveness.

Comparing to the previously proposed approaches, the method presented by Heer and Chi in [7] encompasses all the data features proposed. This method utilizes data features from content and structure, in addition to URL tokens and the sequence ordering already contained in logs. In this paper, we further extend that method to encompass viewing time spent on each page as a weighting scheme, and evaluate it in conjunction with other clustering schemes.

MULTI-MODAL CLUSTERING

Multi-Modal Clustering (MMC) is a technique which utilizes multiple information data features (modalities) to produce clusters. In this section, we summarize how this technique can be applied to cluster user sessions as described in [7].

We first collect the Content, Usage, and Topology (CUT) data of the Web site to be analyzed. We obtain the usage logs and sessionize them using the techniques described in [12]. We obtain the Content and Topology (linkage structure) via a Web crawler. This data is used to construct a vector-space model of user profiles by first creating models of both the Web site and the user sessions and then combining them to generate the user profiles. We then define a similarity metric for comparing these profiles and use it to generate the resulting clusters.

Among the techniques we employ to create these vector space models is the Term Frequency by Inverse Document Frequency (TF.IDF) weighting scheme. A common technique in the information retrieval field, TF.IDF provides a numerical value for each term in a document, indicating the relative importance of that term in the document. This weighting is roughly equal to a term's frequency in a given document divided by the frequency of the term occurring in all documents [16, p. 542].

We model the content and structure of the Web site using a number of information sources. Each source of information (or modality) for a page is expressed as a feature vector. The *modality vectors* are:

- ***Content:*** The content of all pages is processed using the TF.IDF weighting scheme to find the importance of each word. The ***Content*** vector of a page is the TF.IDF weighted keyword vector containing all of the words on that page.
- ***URL***: Each URL is tokenized using '/', '&', '?' and other appropriate delimiters, and then the tokens are weighted using TF.IDF. The ***URL*** vector of a page is the corresponding URL token keyword vector of the URL of that page.
- ***Inlink/Outlink***: The Outlink vector of a page describes which pages are reachable *from* this page, while the Inlink vector describes which pages link *to* this page. Representing the topology of a site using an adjacency matrix, the ***Outlink*** vector of a page is the corresponding row of the matrix, while the ***Inlink*** vector is the corresponding column.

The next phase of our method consists of modeling the user sessions. We represent each session as a vector that describes the session's sequence of transactions. For example, if a Web site consists of 5 pages labeled A through E, a session consisting of page views A→B→D could obtain a vector (1,1,0,1,0) corresponding to the space (A,B,C,D,E).

We have explored a number of possibilities for assigning the actual vector values. These **Path Weightings** consist of several combinations of schemes:

(a) *Uniform*: Each page receives equal weighting in the session, e.g. A→B→D = (1,1,0,1,0).

(b) *TF.IDF*: Treating each session as a document and the accessed pages as the document terms, each page receives a TF.IDF weighting.

(c) *Linear Order (or Position):* The order of page accesses in the session is used to weight the pages, e.g. (1,2,0,3,0).

(d) *View Time*: Each page in the session is weighted by the amount of viewing time spent on that page during the session, e.g. A(10sec) → B(20s) → D(15s) = (10,20,0,15,0).

(e) *Various Combined Weighting*: Each page in the session is weighted with various combinations of the TF.IDF, Linear Order, and/or View Time path weighting. Here is an example with both Linear Order+View Time: A(10sec) → B(20s) → D(15s) = (10,40,0,45,0).

Next we create a representation (or profile) of user interests based on the pages that lie on each user's surfing session. We assume implicitly that each page a user sees is a part of that user's information interest. To represent this profile, we build up a feature vector of each page, and then construct the profile as a linear combination (weighted vector sum) of the feature vectors, using the user sessions to formulate the weightings.

To do this, we first construct a vector ***S*** to describe each session as described previously. Then each page is described using a *multi-modal vector* ***P***, which is a concatenation of the *Content*, *URL token*, *Inlink*, and *Outlink* modality vectors. A user profile ***UP*** is then constructed as linear combinations of the page vectors ***P*** using the weights in ***S***. Each user profile then undergoes normalization, with each modality subvector being normalized to unit length.

We then define a similarity metric ***D()*** for the user profile vectors. To do this, each modality subvector from one vector is compared to the corresponding modality subvector in the other vector using the cosine similarity function, which measures the cosine of the angle between two vectors [16]. The values of these comparisons are then linearly combined to obtain a single similarity value between [0,1].

Modality Weightings are used to help define the relative contribution of each modality in the similarity function. For example, we might specify that the *Content* modality vector should contribute 75%, while *Inlink* should contribute only 25% to the value of the similarity. So, $\mathbf{D(UP_1,UP_2) = .75*cos(UP_1^{content},UP_2^{content}) + .25*cos(UP_1^{Inlink},UP_2^{Inlink})}$.

Using this similarity function, we can then apply traditional clustering algorithms to the user profile vectors. In our studies we used a bisection-based variant of the traditional K-Means algorithm, described in [8]. The algorithm starts with one cluster consisting of all sessions, and uses K-Means to repeatedly bisect clusters until a site-dependent, user-specified number of clusters is achieved.

Summary

In summary, we create vectors to describe various features of each Web page. Each page can then be described as a multi-modal vector. We then model user sessions as multi-modal vectors that are combinations of the multi-modal page vectors. Finally, we cluster the user session vectors to obtain categorizations of the user sessions.

Now that we have described the clustering method, we turn to a description of our plans for validating and understanding these different clustering weighting schemes.

USER STUDY METHOD

Subjects

21 Xerox PARC employees and interns participated in our study. They were told that they were involved in a project exploring Web user session categorization techniques and Web usability improvement. They were not paid for the study.

Material

The live www.xerox.com site was used. The WebQuilt proxy-based logger [5] was used to capture all of the user sessions. We verified that WebQuilt added very little latency from the overhead of using a proxy.

Tasks

There were a total of five information need groupings (task groups). Each group has three different tasks. We designed the tasks such that each group had an easy, medium, and hard task. The idea was to simulate real world task conditions. The tasks were designed by looking through email feedback from the Web site. Here is a brief description of each task:

- *Product*
  (a) Find spec. for Xerox WorkCentre XK50cx.
  (b) Find copier capable of at least 12 pages per minute, and is also a network printer and scanner, and is less than $3000.
  (c) Find a high-end production system with at least 600 dpi, automated production of books, and at least 100 pages per minute.
- *Support*
  (d) Find Win2000 driver for Xerox Document Center 255 printer.
  (e) Find the user manual for Xerox WorkCentre 385.
  (f) Troubleshoot a Xerox 5845 copier where copies are light and faded.
- *Supplies*
  (g) Find desktop laser printer mailing labels.
  (h) Research how to recycle used toner cartridges.
  (i) Find toner cartridge for a HP LaserJet 4L printer.
- *Company Info*
  (j) Find the company's 2nd quarter earnings report.
  (k) Find info on Xerox's new CEO and her plans for the company.
  (l) Research Xerox's invention of electronic paper.
- *Jobs*
  (m) Find jobs in sales in Southern California.
  (n) Research the company's employee benefits.
  (o) Find application and eligibility information on mechanical engineering internships.

Experimental Procedure

We sent each subject an email containing a URL link designed specifically for the subject. The link contained an online consent form and instructions for the study. Subjects were asked to perform the study in the comfort of their office or anywhere else they chose. Subjects were allowed to abandon a task if they felt frustrated, and they were also told that they could stop and continue the study at a later time if they so chose. The idea was to have them work on these tasks as naturally as possible.

Each subject was assigned a total of five tasks, one from each of the 5 task groups. The assigned tasks were counterbalanced for difficulty and then presented in random order. Three users volunteered to do 10 tasks instead of 5. We felt that this did not contribute any undue variability in the study, because some users in the real world do surf longer than others. In the end, each task was assigned roughly the same number of times. We recorded the time of each page access. Whenever the user wanted to abandon a task, or if they felt they had achieved the goal, the user clicked on a link signifying the end of the task. Subjects were then taken to an online form, where they were able to give feedback on whether they felt they completed the task and on any usability problems they might have encountered. We recorded the time they took to handle each task as well as the View Time of each page during each task session.

USER STUDY RESULTS

User Sessions Obtained

3 users started but did not complete any of the tasks, leaving us with 18 users (15 users with 5 tasks each, and 3 users of 10 tasks each, giving a total of 105 user sessions). We had to throw out one of these user sessions because the user had only gone to the home page for that task. While we were reasonably happy with ***104*** user sessions, we realize that a better option is to conduct a more expensive large-scale study with hundreds of users and thousands of user sessions. We shall show here, however, that our data set is large enough to show the differences between the different clustering schemes.

Overview of Results

We studied a total of 320 different algorithm schemes. For each scheme, we clustered the captured user sessions into 5 clusters. We then measured accuracy by counting the number of correct classifications (comparing against our *a priori* task categories) and then dividing by the total number of sessions, thus providing a percentage measure of categorization accuracy.

One trend was immediately obvious: all schemes involving the Outlink modality performed more poorly than others (mean=67.2%, s.d.=23.1%), and those results were omitted in this analysis. The interesting data sets are presented in Table 1. In this Table and the following Figures, each modality is represented with a single character, e.g. *C=Content, U=URL Token, I=Inlink, O=Outlink.* The weightings are specified in brackets, respectively.

For the purpose of comparing to traditional algorithms, we also examined clustering Raw Path vectors without using any of the modalities, but with various path-weighting methods. Raw Path Vectors are simply frequency counts of the pages occurring on the path, e.g. A→B→D→B = [1,2,0,1,0]. This is a common approach taken in the past, though it obtained an accuracy of only 77/104=74% (entry marked in green).

What is most immediately striking is that we're able to achieve results with very high accuracy, with certain cases reaching classification accuracy as high as 103/104=99%!

	uniform	tfidf	time	pos	tfidf, time	tfidf, pos	time, pos	tfidf, time, pos	*Ave-rage*	*Std. Dev.*
RAW PATH	74%	87%	89%	83%	70%	87%	74%	62%	*78%*	*10%*
CONTENT	99%	79%	98%	99%	98%	97%	98%	98%	*96%*	*7%*
URL	79%	54%	94%	81%	91%	66%	89%	89%	*81%*	*14%*
INLINK	68%	64%	71%	72%	68%	61%	72%	73%	*69%*	*4%*
OUTLINK	63%	59%	56%	57%	54%	59%	55%	54%	*57%*	*3%*
CU [0.75,0.25]	88%	78%	98%	99%	97%	81%	98%	93%	*92%*	*8%*
CU [0.5,0.5]	81%	78%	97%	87%	96%	80%	97%	92%	*88%*	*8%*
CU [0.25,0.75]	79%	79%	94%	87%	93%	78%	97%	92%	*87%*	*8%*
CI [0.75,0.25]	80%	94%	96%	97%	96%	95%	97%	98%	*94%*	*6%*
CI [0.5,0.5]	87%	94%	94%	94%	96%	95%	96%	98%	*94%*	*3%*
CI [0.25,0.75]	87%	88%	94%	93%	92%	95%	90%	93%	*92%*	*3%*
CUI [0.5,0.25,0.25]	83%	94%	94%	95%	96%	85%	97%	98%	*93%*	*6%*
CUI [0.25, 0.5, 0.25]	92%	81%	95%	95%	95%	79%	97%	94%	*91%*	*7%*
CUI [0.25, 0.25, 0.5]	87%	94%	94%	94%	96%	69%	94%	95%	*91%*	*9%*
CUI [0.33, 0.33, 0.33]	92%	93%	94%	95%	95%	85%	97%	95%	*93%*	*4%*
Average over all	*83%*	*81%*	*91%*	*89%*	*89%*	*81%*	*90%*	*88%*	*86%*	
Std. Dev over all	*9%*	*13%*	*12%*	*12%*	*13%*	*13%*	*13%*	*14%*	*13%*	

Table 1: Raw accuracies of 120 out of 320 clustering schemes, with interesting entries highlighted. Each entry specifies the % of user sessions out of 104 that were correctly clustered using the given scheme. All Outlink-based schemes were eliminated due to poor performance.

Content Modality Improves Accuracy

First, we analyze the cases where we use only a single modality in the clustering (uni-modal schemes). Figure 1 shows that the Content modality performed best (mean=95.8%, s.d.=6.9%), with URL (mean=80.5%, s.d.=14.0%) and Raw Path (mean=78.1%, s.d.=9.7%) following behind. While the traditional Raw Path method performed reasonably well with various path weightings, it had a wide variation. What's clear from this chart is that Inlink and Outlink both performed poorly on their own.

Content performed admirably, with poor performance only when used with the TF.IDF path weighting alone. We used Linear Contrast to test differences between Content and the other 4 uni-modal schemes, and found significance there ($F(1,35)=33.36$, $MSE=.007332$, $p<0.0001$). Expanding this to all multi-modal schemes, we compared all of the content-based schemes vs. non-content-based schemes. We found significant differences there also ($F(1,105)=32.51$, $MSE=.005361$, $p<0.0001$). To summarize, crawling the site and using the page content to help cluster user sessions greatly increases algorithm accuracy. This is far from surprising; intuitively, the words that the user sees during each session are good indicators of their information need.

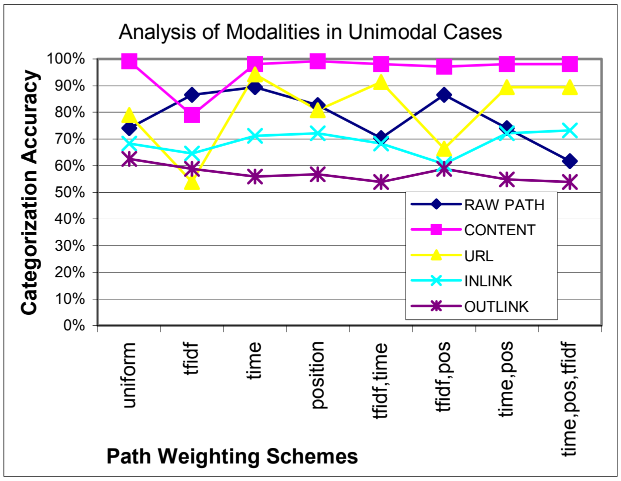

Figure 1: Plot of each different modality's accuracies across different path weighting schemes.

View Time Improves Accuracy

Figure 2 shows the analysis of path weighting. The left portion of the chart shows the uni-modal cases, while the right side shows the multi-modal cases. TF.IDF and uniform weighting performed poorly in general.

What's most interesting from this chart is that View Time path weighting performed well, staying at the top of the curve across the chart. This is true regardless whether we're looking at the uni-modal or the multi-modal schemes. A paired t-Test found a significant difference between View-Time-based schemes vs. non-View-Time-based schemes (n=60, *V.T.mean*=89.5%, *s.d.*=12.7%, *non-V.T.mean*=83.2% *s.d.*=12.0%, $t(59)=4.85$, $p=4.68e-6$).

Interestingly, using the View Time path weighting with methods that do not require crawling a Website gave reasonably good results. Raw Path clustering performed decently (93/104=89.4%), while clustering URL tokens provided even better results (98/104=94.2%).

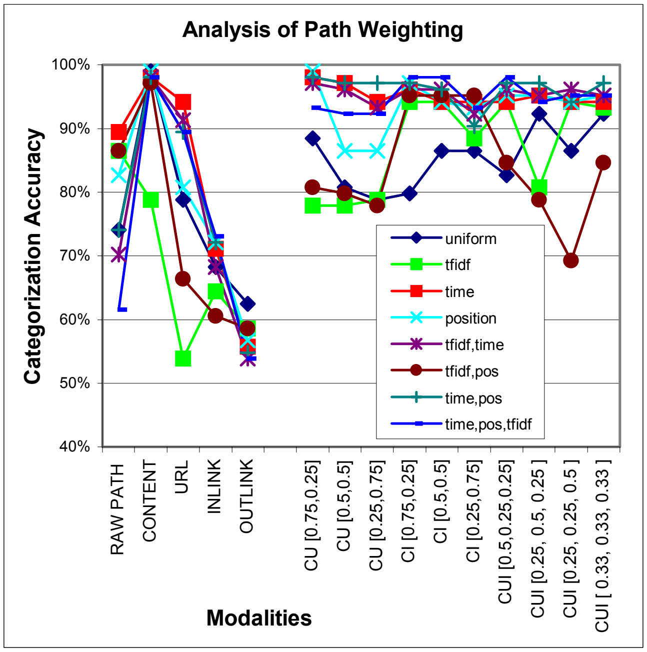

Figure 2: Plot of each path weighting schemes against modality.

Multi-modal vs. Uni-modal

The data here suggest that by employing a multi-modal approach, we retain the same high accuracy that is attained with content-based clustering. Linear Contrast shows that there is no significant difference between multi-modal content-based schemes vs. the uni-modal content-only scheme ($F(1,77)=1.63$, $MSE=.004407$, $p=.21$).

However, multi-modal clustering should be more robust in real life applications. This is because there are many Web sites that have pages containing only images, sounds, videos, or other media formats. These pages cannot be parsed for content, making the usage of other modalities much more important. In particular, the Inlink modality does not even rely on specific features of the page, but instead depends only on the other documents in the collection which link to that page. Moreover, in our experience, once the content is being parsed to enhance algorithm performance, the other modalities such as URL Token or Inlink do not add significant processing times.

Recommendations

Based on this analysis, we offer some recommendations:

First, our analyses show that good results can be achieved with simple schemes. Given that the View Time path weighting makes any clustering scheme more robust, we recommend using it when possible. Simple schemes such as *Raw Path+View Time*, or *URL Token+View Time* give good results, without incurring the cost of having to process the content of the pages.

Second, using the Content modality makes the clustering highly accurate. If extra computation time and resources are available, it will almost guarantee excellent results. For example, *Content+(Inlinks and/or URL)* could be used.

Perhaps most importantly, using the analyses provided here, we could tailor the clustering technique to the uniqueness of the site being analyzed. If the site has many pages without word content, then Inlink and URL Token modalities could be used in combination with View Time.

ISSUES

There are a number of issues to consider when applying the techniques discussed here:

While content-based clusterings provide the highest accuracy rates, they require much more work – the site must be crawled and then processed. Though Raw Path and URL Token schemes did not perform as well, they still achieved high accuracy when the appropriate weightings are applied and are much easier to apply, requiring only the server logs. The data suggest that the appropriate method to apply for a specific case should be motivated by the particular needs and resources.

Some factors could affect the clustering accuracy results reported here: (a) We believe that logs from well-designed sites are easier to cluster, because each user session vector will be more distinct and separable from other vectors. However, we know that Xerox.com is typical of corporate sites with serious design issues, thus making our study more applicable to real-world situations. (b) Task choice also greatly affects the ease of clustering. Our tasks were chosen such that they are typical of the Xerox.com user tasks. Moreover, we believe some tasks in the product group were hard to separate from the support task group, and some tasks across different task groups were similar, but we were still able to cluster them correctly.

Another outstanding issue is determining the number of clusters to create. A simplistic approach is to choose a suitably large number, and then merge or recluster as necessary. The selection of this number is dependent, of course, on the size and diversity of the site being analyzed. Automating the choice of clusters is an area for future research.

CASE STUDY

To demonstrate the efficacy of these session categorization methods in the real world, we conducted a case study on the current www.xerox.com site. We obtained HTTP server access logs from July 24, 2001 and then processed them using the LRS method. The final output consisted of 32,813 distinct LRS paths. The Xerox site was then crawled using the freely available wget utility. We also retrieved any relevant URLs found within the logs that were not captured by the crawl, ensuring that all documents were included.

User profile vectors were then created using the *Content* and *Inlink* modalities. The vectors were clustered into 15 clusters using bisecting-KMeans and the weighted cosine measure discussed previously. The *Content* and *Inlink* modality weights were 0.75 and 0.25 respectively. The data set was clustered on a 1.7GHz Linux compute server in approximately 13 minutes.

The number of clusters (15) was chosen to provide enough clusters to reveal the major top-level usage trends. Some of these clusters were then manually combined due to high similarity, leaving 9 final clusters. We determined cluster labels by examining the cluster output, particularly the highest weighted keywords and the nearest Web pages.

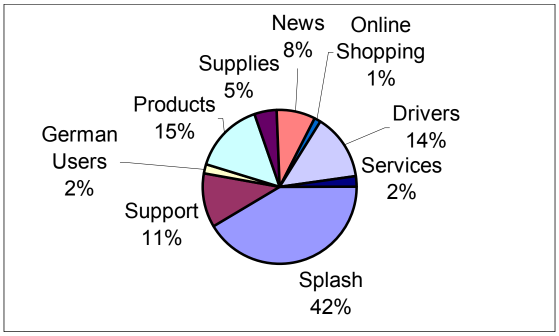

Figure 3: CI[0.75,0.25] Clustering (Xerox.com, 7/24/2001)

Figure 3 depicts the results of the case study. The most striking result is that a large segment of user sessions (41.4%) center around the splash page. Viewing the actual clustered paths revealed that these sessions consisted primarily of the site's splash page, with many paths jumping between the splash page and other linked pages. This could indicate that a large segment of users may come to the site without well-defined information needs and/or that the site may suffer possible usability problems that prevent users from successfully moving deeper into the site.

Other substantial groupings include Xerox Product Catalog browsing (15%), Driver downloads (13.9%), Technical Support (11.5%), and Company News and Information (8.2%). One unexpected result was that there was a strong, concentrated group of German users that necessitated a unique cluster (1.7%). Xerox Sales and Marketing might also be interested to know the number of Online Shopping / Purchase related sessions (1.3%) in comparison to the number of product catalog viewers. As discussed in [7], more detailed information about these groupings can be obtained by reclustering a given cluster. For example, we learned that within the Products group, 43% of the sessions centered around the Phaser line of printers.

This case study illustrates that these user session clustering methods are indeed applicable to real world situations, and are scalable to large, heavily used Websites. Given the accuracy levels reached in our previous data analysis, we can conclude with higher confidence that the generated clusters accurately represent the interests of Xerox.com visitors. Finally, the case study shows that these clustering techniques can reveal site usage trends that are of great interest to Web designers and marketers.

CONCLUSION

Content providers and eCommerce businesses on the Web are realizing that Web usability directly affects the success of their Web sites. As usability professionals, we are deeply involved in making the Web more accessible to users. We need to know what our users are doing in order to better optimize the Web sites.

Recent research seeks to understand the composition of user traffic using Web usage mining techniques on Web server logs. The commonality is to first build up user profiles based on the user visitation paths, and then apply clustering techniques to these user profiles. However, each technique's validation is conducted on a different Web site, making it extremely difficult to compare the different algorithm results. What's worse is that since there is no way of knowing *a priori* what the true user information need is for each user session, we had no way of knowing whether the algorithms performed correctly.

In this paper, we present the results from a systematic evaluation of different clustering schemes by conducting a user study where we asked users to surf a large corporate site with *a priori* specified tasks. By knowing what the tasks were and how they should be grouped in advance, we were able to do post-hoc analysis of the effectiveness of different clustering schemes. What we discovered was that, by counting the number of correct categorizations, certain combinations of data features enabled us to obtain accuracies of up to 99%. The naïve scheme of using Raw Path vectors gives an accuracy of only 74%, while certain combinations give accuracies below 60%.

We showed in detail that two aspects are the most important in the clustering schemes: (a) Using the *Viewing Time* of each page on the user path improves the clustering accuracy and robustness; (b) If extra precision is required, we can obtain up to 99% accuracy by building user profiles using page content vectors. However, the disadvantage is that this requires retrieving and parsing each page's content.

Lastly, while the *Inlink* and *URL* vector performed less than optimally as data modalities on their own, they performed well in combinations with other modalities. Because certain hyperlinked sources on the Web cannot be parsed for words (e.g. images, sound, and video files), we believe these two modalities may be able to compensate for missing content vectors, thus making the clustering extremely robust.

In summary, our experiment shows that clustering user sessions should be done carefully, so that designers do not use wrong conclusions to make optimization decisions. More importantly, we were able to obtain extremely high accuracy by paying attention to the data modalities used in the clustering process. This is encouraging news for people trying to understand site usage.

Within the last few years we have seen Web usability grow as a field. While problems in this area are being understood and solved daily, given the size and the growth of the Web we will continue to need improvements in accurate and scalable methods for understanding user behaviors. We believe that this research contributes to the understanding of this puzzle.

ACKNOWLEDGMENTS

We would like to thank George Karypis for providing an implementation of the bisecting-KMeans algorithm. We also thank Pam Schraedley for help on statistics and suggestions. This research was funded in part by the Office of Naval Research. Finally, we would like to thank the subjects who volunteered to participate in our experiment.

REFERENCES

1. Banerjee, A. and Ghosh, J. Clickstream Clustering using Weighted Longest Common Subsequences, in *Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining* (Chicago IL, April 2001), 33-40.
2. Barrett, R., Maglio, P.P., and Kellem, D.C. How to personalize the Web, in *Proc. of the ACM Conference on Human Factors in Computing Systems, CHI '97* (Atlanta GA, March 1997), 75-82.
3. Cooley, R. *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data*. Ph.D. Thesis, University of Minnesota, May 2000.
4. Fu, Y., Sandhu, K., Shih, M. Asho Generalization-Based Approach to Clustering of Web Usage Sessions, in *Proc. of WEBKDD 1999* (San Diego CA, August 1999), 21-38.
5. Hong, J.I., Heer, J., Waterson, S., and Landay, J.A. WebQuilt: A Proxy-based Approach to Remote Web Usability Testing, to appear in *ACM Transactions on Information Systems*. Available at: http://guir.berkeley.edu/projects/webquilt/pubs/acmTOIS-webquilt-final.pdf
6. Huang, Z., Ng, J., Cheung, D.W., Ng, M.K., Ching, W. A Cube Model for Web Access Sessions and Cluster Analysis, in *Proc. of WEBKDD 2001* (San Francisco CA, August 2001), 47-57.
7. Heer, J. and Chi, E.H. Identification of Web User Traffic Composition using Multi-Modal Clustering and Information Scent, in *Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining* (Chicago IL, April 2001), 51-58.
8. Karypis, G. and Han, E. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. Technical Report TR-00-0016, University of Minnesota, 2000.
9. MacQueen, J. Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1 (1967), UC Berkeley Press, 281-297.
10. Nielsen, Jakob. Did Poor Usability Kill E-Commerce?, in *Jakob Nielsen's Alertbox* (August 19, 2001). http://www.useit.com/alertbox/20010819.html
11. Padmanabhan, V.N. and Mogul, J.C. Using Predictive Prefetching to Improve World Wide Web Latency. *ACM SIGCOMM Computer Communications Review*. 26(3), 1996.
12. Pirolli, P. and Pitkow, J.E. Distributions of Surfers' Paths Through the World Wide Web: Empirical Characterization. *World Wide Web*, 2(1-2), 1999. 29-45.
13. Pitkow, J.E. Summary of WWW Characterizations. *World Wide Web*, 2(1-2), 1999. 3-13.
14. Pitkow, J. and Pirolli, P. Mining longest repeated subsequences to predict World Wide Web surfing, in *Proceedings of USITS '99: The 2nd USENIX Conference on Internet Technologies and Systems* (Boulder CO, October 1999).
15. *Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining* (Chicago IL, April 2001).
16. Schuetze, H. and Manning, C. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge MA, 1999.
17. Shahabi, C., Zarkesh, A.M., Adibi, J., and Shah, V. Knowledge Discovery from User's Web-page Navigation, in *Proc. 7th IEEE Intl. Conf. On Research Issues in Data Engineering* (1997), 20-29.
18. Spiliopoulou, M., Pohle, C., and Faulstich, L.C. Improving the Effectiveness of a Web Site with Web Usage Mining, in *Proc. of WEBKDD 1999* (San Diego CA, August 1999), 142-162.
19. Stabin, T. and Glasson, C.E. First Impression: 7 commerical log processing tools slice and dice logs your way, (1997). Available at http://www.netscapeworld.com/netscapeworld/nw08-1997/nw-08-loganalysis.html
20. *Proc. of the SIGKDD Workshop on Web Data Mining (WEBKDD01)* (San Francisco CA, August 2001).
21. Yan, T.W., Jacobsen, M., Garcia-Molina, H., and Dayal, U. (1996), From User Access Patterns to Dynamic Hypertext Linking. *Computer Networks*, vol. 28, no. 7-11 (May 1996), 1007-1014.
22. Zaiane, O.R., Xin, M., and Han, J. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs, in *Proc. Advances in Digital Libraries ADL'98* (Santa Barbara CA, April 1998), 19-29.