Characterizing Customer Groups for an E-commerce Website

Qing Wang
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, CANADA
qiw237@mail.usask.ca

Dwight J. Makaroff
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, CANADA
makaroff@cs.usask.ca

H. Keith Edwards
Department of Computer Science
University of Western Ontario
London, ON, CANADA
hkedward@uwo.ca

ABSTRACT

In conventional commerce, customer groups with similar interests or behaviours can be observed. Similarly, customers in E-commerce naturally form groups. These groups allow the organization to provide quality of service (QoS) and perform capacity planning. From a system point of view, overall server performance can be improved and resources managed considering customer session behaviour.

Previous studies have grouped customers using clustering techniques. Different data metrics have been selected as criteria for grouping, in order to analyze different problems. The limitation for these approaches is that problems are analyzed separately. In order to manage an E-commerce server well, we must analyze many related problems comprehensively rather than separately. For example, we would like to know the impact on resource usage when optimizing revenue. Thus, we must understand the differences and similarities between session groups chosen by different metrics.

This paper characterizes customer groups for an E-rental business and compares customer groups created according to different criteria including services requested, navigation pattern and resource usage. A significant finding of this study shows that using each of the three criteria independently yields roughly similar results, since customers looking for similar services tend to have similar navigation pattern as well as similar server resource usage. Thus, it is sufficient to group customers in only one of these ways. Grouping customers by services requested is suggested since this method yields relatively better results and is simple to implement.

Categories and Subject Descriptors

H.4.1 [**Information Systems Applications**]: Communications Applications; H.3.5 [**On-line Information Services**]: Web-based services; H.3.3 [**Information Search and Retrieval**]: Clustering; K.4.4 [**Electronic Commerce**]: Distributed Commercial Transactions

EC'04, May 17–20, 2004, New York, New York, USA.

General Terms

Performance, Measurement

Keywords

workload characterization, session behaviour, electronic commerce

1. INTRODUCTION

In conventional commerce, groups of customers with similar interests or behaviours can be observed. Likewise, a customer's interaction with an E-business site contains characteristics which are similar to other customers. Each user's interactions with the site can be represented by a *session* comprising the sequence of requests issued by a single customer to the site during a specific period of time. Hence, a customer group in E-commerce can also be referred to as a session group. An important task of E-commerce workload characterization is to analyze customer behaviour to identify session groups. Recognizing and adapting to session groups can be used to improve server performance (either throughput or revenue), implement quality of service (QoS) and admission control, and perform capacity planning.

Previous work, in both the areas of workload characterization [1, 13] and web usage mining [2, 5, 8, 16], created session groups using clustering techniques. The main difference between these studies is the metric chosen to represent a session. A session has many data features, including services requested, navigation pattern, resource usage, page viewing time, page content, page links, etc. Each data feature provides a "view" of a session. One or more features can be selected as a metric to represent a session.

Existing work has used independent grouping schemes for independent problems. It is natural to assume that the criteria would be dependent on the website characteristics in general and on the nature of the performance issue being studied. For example, Menascé et al. [14] grouped sessions based on navigation patterns to improve server resource management and optimize revenue, while Arlitt et al. [1] chose resource usage as the metric in order to discuss scalability issues. Session groups based on navigation patterns can be used for server resource management and optimize revenue, but not necessarily for scalability issues, or vice versa, unless the groups generated are similar.

Nevertheless, the approach of grouping sessions by a specific metric to analyze a specific problem has provided useful

results. In order to manage an E-commerce server well, it is more efficient to analyze many related problems comprehensively rather than separately. For example, we would like to know what is the impact on resource usage when optimizing revenue. The major goal of this work is to examine the need for the distinctions made in previous work, and indeed, if the grouping criteria lead to substantially different group formations with different characteristics. This has not been explicitly addressed in the literature.

This paper characterizes customer groups for an E-rental business and compares customer groups created according to different criteria including requested services, navigation pattern and resource usage. The results show that using each of the three criteria independently yields roughly similar results. The performance implication is that we can analyze problems associated with functionality, navigation patterns and resource usage based on the same set of session groups. Although it is sufficient to group customers in only one of these ways, grouping customers by requested services is suggested since this method yields relatively better results and is simple to implement.

The remainder of this paper is organized as follows: Section 2 briefly describes the related work; Section 3 describes how sessions were isolated from web traces and presents a simple statistical analysis of sessions. Section 4 presents the algorithm to group sessions based on our selected criteria while the characteristics of the identified session group types are analyzed in Section 5. Section 6 compares the session grouping methods and Section 7 contains the conclusions and suggests directions for further work.

2. RELATED WORK

Workload characterization and web usage mining research puts user sessions into groups using clustering techniques. In general terms, clustering associates items with similar characteristics into groups. An item is represented by a data set; the similarity between items is defined and computed based on chosen data features. The existing clustering approaches differ from each other in the definitions of user sessions and similarity and the clustering algorithm used. Further details of the approaches are described in the remainder of this section.

Menascé et al. [13] examined E-commerce workload based on customer behaviour (navigation pattern). Customer interaction during a session is captured in a Customer Behaviour Model Graph (CBMG). A CBMG is a first-order Markov chain with states representing what types of services a customer may request. Customers navigate from one state to another with measured probabilities. A session is described by an $n \times n$ matrix of transition counts between states i and j, $[c_{i,j}]$. The k-means clustering algorithm is applied to the sessions. In this algorithm, a session is considered as a point in a virtual space; then k points in the space are selected as estimated centroids of the k clusters; the remaining points are grouped to the cluster with the nearest centroid. A strength of CBMGs is that they allow us to understand the customer transitions and identify customer groups. These groups provide a basis for implementing personalized service and priority-based server resource management policies [14]. However, CBMGs characterize E-commerce workload based on only customer behaviour.

Arlitt et al. [1] characterized E-commerce workload based on the level of demand on resources in order to study scalability. Requests can be classified into roughly three classes: cacheable, non-cacheable and search. These request classes are distinguished by their different resource demands. A session is then described by a vector of three attributes: $(a_1/n, a_2/n, a_3/n)$, where a_1, a_2 and a_3 are the number of requests of each type, and n is the total number of requests in the session. The k-means clustering algorithm was applied to group sessions, resulting in four session groups: heavy cacheable, moderate cacheable, search and non-cacheable, which are distinct in CPU demand. They demonstrates that the system scalability is sensitive to the request class mix, request cache hit rate and the degree of personalization of services. However, the behaviour of the customer groups identified in this way is not clear.

Shahabi et al. [16] considered page viewing time as a primary feature to describe a session and clustered sessions using k-means clustering algorithm. The accuracy of this method is low. Banerjee [2] improved this method by representing a session with a sequence of pages visited and calculating the similarity between two sequences based on longest common subsequence and page viewing time. Partitioning the graph was used to cluster the sessions. Although it seems reasonable to use page viewing time as an indication of a user's interest on the page, it is application dependent.

Heer and Chi [8, 9, 10] utilize multiple modalities of information to group similar user 'profiles' into user categories. A user profile represents a path of page traversals. A page is further represented as a multi-modal vector with four modalities: page content, URLs, in-links, and outlines. Sessions were clustered using Wavefront Clustering technique, which is a variant of k-Means clustering algorithm. As this method models sessions in a finer degree of granularity, there is a potential scalability problem. Fu et al. [5, 6, 7] grouped pages with the same URL prefix to reduce the number of different pages in a session before applying the clustering algorithm.

Xiao et al. [18] proposed a measurement of similarity between users based on a chosen data feature, which could be page-view, frequency viewing a page, time viewing a page or viewing order. An $n \times n$ similarity matrix containing the similarity measurement among all n users is then computed. Clustering users with similar interests is performed by permutation of the similarity matrix. This method is unique with respect to other work mentioned here, but when n is large, the computation is very expensive.

Estivill-Castro and Yang [4] pointed out that most clustering algorithms in the literature are difficult to use for grouping users by navigation, since the similarity between two navigation paths is a high-dimension problem. This is especially true when more data features are considered to compare two paths. They presented a randomized, iterative algorithm to solve the problem.

The existing research on clustering shows that complicated approaches are used to select an appropriate data abstraction for a user session and define the similarity between two sessions. This is because that there are many data features for a session and that the importance of a data feature was assessed differently by each of the research groups. It is difficult to compare existing approaches, since the data features chosen to represent a session and the methods to evaluate the similarity of two sessions are different from one another; the nature of websites is also very different between different approaches. The issue of data features representing a session has not been addressed in detail.

In this research, we used three different data features for grouping sessions: services requested, navigation pattern and resource usage. Sessions in an E-commerce site are grouped independently by each of these chosen criteria, thus the association and the comparison among them can be explored. These data abstractions were selected since they are important features relating to basic issues in E-commerce server performance and management, such as QoS, personalization, server resource management and capacity planning. There are some other session data features which are also, to some degree, associated with the issues mentioned above. However, it would be desirable to choose as few as possible to simplify the analysis. These abstractions have been used in previous approaches, but the comparisons between them have not been understood well.

Previous E-commerce performance literature provides insufficient detail on clustering algorithms to reproduce all the previous characterizations precisely for our data set. For example, selecting the k centroids to start the clustering process in the k-means method is still a tricky issue. A combination of Minimum Spanning Tree method and the k-means method [**?**] is used in this research to deal with this issue. Both the data abstractions and clustering methods were chosen for the potential for accuracy and the degree of computational complexity.

3. OBTAINING USER SESSIONS

In this section, we explain the general characteristics observed for user sessions without distinction between the different types of sessions. This rudimentary analysis is used to obtain a general picture of sessions on the site and to compare the behaviour of users with previous work.

The HTTP logs used in this study are from an E-rental business. The files capture customer interaction for one day (24 hours). The web server is Microsoft Internet Information Server 5.0 (IIS). The logs used in this research are in W3C extended log format.

Requests to web servers are divided into two categories: a) explicit user requests, issued by customers for web pages containing services they want, and b) browser generated requests, which are issued automatically by web browsers for embedded objects in the user-requested web pages. In this case, explicit user requests make up of about 6% of the work load and the other 94% are for embedded objects. Explicitly requested pages are all dynamic pages (ASP), while embedded objects are mainly images (.gif and .jpg), javascript (JS), and cascading style sheets (CSS). To focus on customer behaviour, the logs are filtered to remove requests for embedded objects. The two filtered logs were then combined into one with requests ordered by their time value.

To obtain customer sessions from HTTP logs, we must determine what requests are from the same customer. A reliable and efficient way to do that is by the use of cookies. About 82% of the filtered requests came with cookies. Most of the requests without cookies are believed to be the first requests of user sessions. Typically, the first request from a customer does not contain a cookie, then the server will assign a cookie to the request stream from that IP address and the cookie will be used for the session. Hence, it is difficult to identify the first request for a session.

Considering cookies in isolation may be insufficient to identify sessions coming from the same IP address. A customer may initiate many sessions during the period when the logs are collected and these may have the same cookie in another way, as the time-out for cookies can be long. Thus, session boundaries must be identified. Session boundaries are determined by a period of inactivity by a customer. If the time difference between the current request and the last request of an ongoing session is less than a request interval threshold, the current request belongs to that session.

Some web access technologies have specific policies which govern session boundaries. By default, an ASP (Active Server Pages) session ends after 20 minutes of inactivity [3], but this timeout value can be customized if necessary. The accuracy of identifying sessions based on cookies also depends on the percentage of distinct cookies for all sessions. If the percentage of distinct cookies is high, most of the sessions will be correctly identified regardless of correctness of request interval thresholds. In this case, for 16,512 identified sessions, there are 16,090 distinct cookies (i.e. 98%). Thus, the threshold time value really does not matter, and so the default value of 20 minutes is used in this research. Sessions started in the first and last 20 minutes of the trace were ignored to eliminate the session fragments caused by the warm-up and cool-down effects.

Figure 1 shows some general characterization of the sessions obtained from the log. There are a significant portion of short sessions, even after counting the first request, which cannot be identified for each session. About 12% of the sessions have only two requests and about 30% of the sessions have three requests (Figure 1 (a)). Only about 5% of sessions are longer than 20 requests and the average session length is about 7. The distribution of session duration (Figure 1 (b)) again demonstrates the existence of short sessions, about 35% of sessions last less than 40 seconds. Previous research on session-level E-commerce workload characterization [1, 15] showed that: 1) 88% of the sessions are less than 10 requests in length, the distribution of session length is heavy tailed; 2) most sessions last less than 1,000 second. Thus, the general session characteristics observed in this research are quite consistent with previous research.

4. IDENTIFYING SESSION GROUPS

4.1 Merging web pages by functionality

The services an E-commerce site provides to customers are implemented through web pages. Customers visit select web pages for desired services. For an E-commerce site, there may be hundreds or even thousands of web pages, each of which can be considered a "state" of a session, from the customer's point of view. Web pages should be grouped by functionality to reduce the complexity of the group-forming task if the criterion for grouping is services requested or navigation pattern. If every web page is represented by a state, there will be too many states, resulting in misleading details and overwhelming amounts of extra calculations.

Web pages providing the same functionality can be merged to reduce the state space for the clustering algorithm. One merging method examines the URL of a web page. In our trace, two web pages with the same URL prefix have the same general functionality and can be grouped together. In other sites, the path ending or filename may refer to the same page, though the path prefix may be different, as when symbolic links are used.

There are 485 distinct web pages in the reduced log, which can be merged manually into 17 services (Table 1). The

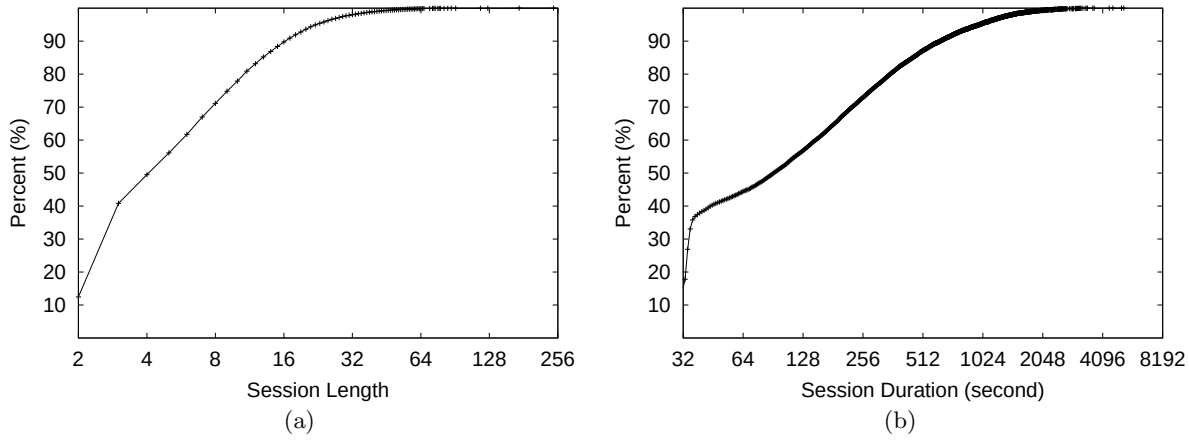


Figure 1: CDFs for session characteristics

merging was easy since the particular design of this website had a structure that matched the functionality of the pages. The organization of web pages on the site had a tree structure. The merging task consisted of deciding what level of nodes to pick on the tree structure as the grouping granularity. An automatic classification of pages into services is possible since most web sites use a tree structure.

This type of classification of pages is highly dependent on the design of the website. Some websites may have a different structure, such as the name of the program used to manipulate the data and access the database. In such an environment, the query string may need to be examined to determine the functionality of a particular request. Even for the websites using simple tree structures, the categorization may not be permanent since websites tend to change from time to time. We believe that a method of merging functionality can be found for each environment with a reasonable amount of effort, but the present structure of the site simplified the task.

Table 1 shows that the E-rental site provides another model of customer interaction from those of on-line bookstores, or general shopping sites. In particular, there are natural restrictions on the interactions that occur. A customer is looking for one item when renting a vehicle, not multiple items as may be typical in a bookstore. There are several parameters for that one item, such as car type, rental dates, pickup locations, payment terms, and other options (child seats, late arrival, airline information). In a bookstore, the item has few attributes. The only choice would be quantities, hardcover/softcover distinction and payment/delivery options. In a clothing store, there may be sizes or colours, as well as details of payment/delivery. The process of browsing in the E-rental site may be more significant.

At the highest level, customer activities at a E-rental site are similar to that at a book store or clothing shop. Customers that eventually purchase items from commercial sites all follow the search-select-buy step, though the search may be optional. The session characterization and grouping results obtained from a E-rental site is relevant and can provide insight into the patterns of other B2C E-commerce sites.

Table 1: Grouped web pages for the site

Abbrev.	Web Pages	Number of Requests	Percentage of Requests
hom	home page	17221	12.6
expL	express lane	741	0.54
gSrv	group service	948	0.69
info	info, help	7141	5.2
loc	available locations	5847	4.3
othr	others	1101	0.80
prmt	promotions/spec. offers	5323	3.9
rChR	check rate	19659	14.4
rCnl	cancel reservation	845	0.62
rHom	reservation home page	23699	17.3
rMkR	make reservations	819	0.59
rMod	modify reservations	1251	0.91
rPpU	popup search info	21420	15.6
rQut	reservation quote	20430	14.9
rVew	View reservations	3794	2.77
trvl	travel information	1842	1.35
vhcl	vehicles to choose	4799	3.50

4.2 Selecting metrics to describe a session

Different metrics have been selected to describe a session [1, 2, 5, 8, 13, 16], which are somewhat application dependent. Three sets of metrics are selected in this research: services requested, navigation pattern and resource usage.

The metrics we have chosen also differ in terms of computational resources needed for the clustering process as the number of states that need to be represented is vastly different. If the number of services is n, the number of states used will be n when grouping by services, but n^2 by navigation pattern. Additionally, our initial version of dimensioning the space by resource usage has a constant number of states. In particular, only two states are used to combine sessions into groups. Further details of the methods used to group services are described in the rest of this section.

4.2.1 Grouping session by services

A website provides many services to customers. A customer typically looks for specific services when visiting a web store. To perform such grouping, a session is represented by

the set of distinct web pages requested. A virtual coordinate space in n-dimensions is used (where n is the number of web page categories identified), in which a session is a point. If the session requests a web page at least once, its coordinate for this dimension is defined as 1, otherwise, it is 0.

4.2.2 Grouping sessions by navigation pattern

When grouping sessions by services (requested web pages), the focus is on what distinct web pages have been requested. Information regarding how many times a service is requested in a session and how a customer browses among web pages to find a service is ignored. Finer granularity regarding customer behaviour and navigation patterns can be analyzed based on this information.

To capture a customer's navigation pattern, the basic unit of browsing is represented by a *move*. A *move* is a single action of browsing and is defined by the starting and destination web pages. It is the transition between a pair of consecutive requests. A session can be represented by a set of distinct *move*s and is a point in the space defined with all distinct *move*s as dimensions. If a session makes a *move*, then its coordinate at the corresponding dimension is the number of times that this particular *move* was made.

4.2.3 Grouping sessions by resource usage

To manage the server resources efficiently (i.e. CPU cycles, I/O bandwidth and memory) or to perform capacity planning, it is important to understand the resource usage pattern. Grouping sessions by resource usage aids in this task. It is difficult to measure and analyze the exact resource consumption for a request from the log data we have obtained, though coarse analysis is possible. Then, further measurement or simulations based on the characteristics obtained can be performed. If an organization wished to know resources used for each type of request in detail, a system could be instrumented and detailed measurements taken on an otherwise idle system.

System Response Time (SRT) for a request is the period of time from the receiving the request to sending the response. It includes all CPU time, queuing time and disk time on web servers, application servers, database servers and payment servers. SRT for a request consists of two parts: i) Minimum SRT (MSRT), which is the SRT when the processing of this request is not interfered by other requests, and, ii) waiting time, which is time the request waits for its turn to use each of the various resources/servers. MSRT is somewhat stable for a web page, while the waiting time is a stochastic process that depends on the service capacity and the queue length at each server in the path. MSRT for a web page is a good indicator for its demands on system resource. However, MSRT is often difficult to determine since the system always processes many requests in parallel.

If the server is not busy, there is very little waiting time and it is reasonable to approximate MSRT by the SRT. In this trace, the server was not in a state of high load[17]. If the server was heavily loaded, response time would be proportional to the arrival rate of requests.

Figure 2 shows the request arrivals for a short excerpt of 10 minutes during the day in 1-second time slots, during the 20th hour of the trace, which was the peak of activity. The traffic is very bursty at this time scale. Even then, we can see that there are some seconds with no arrivals, though the server is not idle. Figure 3 shows the SRT for requests for the web page rQut for the entire day. There are a substantial number of outliers, and we observed that requests were redirected for a time, and some had long waiting times. All other web pages exhibit a similar pattern. Our previous study [17] presents more evidence of the server's light load.

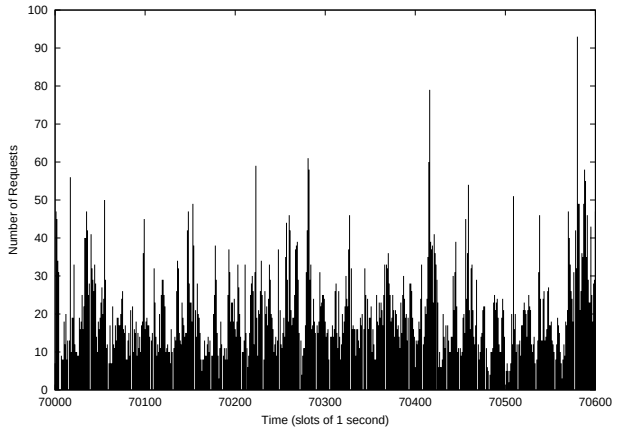


Figure 2: Request arrival process at peak

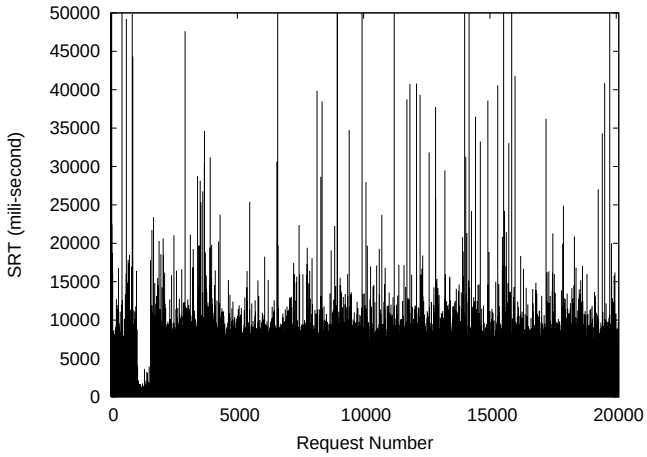


Figure 3: SRT for web page rQut

To evaluate MSRT for a requested service, the distribution of SRT over time obtained and the median of SRT is chosen as MSRT. The median of the SRT value is a reasonable upper bound on the true MSRT. We believe the median is within a factor of 2 of the true minimum for non-directed requests, because by observation, most values fall in a narrow range, though there is a large variance in SRT caused by queueing time and transmission time. It is not possible to obtain precisely the minimum time from a real system trace as these waiting time effects cannot be isolated properly. Some requests must access data on more servers than others. If a request makes a change to a database, it is also the case that time taken to acquire locks to ensure database consistency would be different for different request types.

At the least we can compare the resource usage of two web pages based on MSRT. Once the MSRT for a requested

service is determined, the MSRT for a session is easy to calculate by summing the MSRT for each request. To group sessions by resources used, a session is represented by a total MSRT and average MSRT per request. Then the clustering algorithm is applied to the two dimensional virtual space to perform grouping. Range normalization [11] is applied before clustering to restrict the range of the values to [0,1].

4.3 Clustering algorithm

A clustering algorithm performs item grouping[12]. In our case, an item is a session. An item a can be viewed as a point in a virtual space with dimensions of $(x_1, x_2, ..., x_n)$, where $x_1, x_2, ..., x_n$ are the chosen metrics. A cluster is defined as a set of items which are close to each other within the dimensions of the particular space being considered. The centre of a cluster (hereafter called a **centroid**) is represented by a point whose coordinate is calculated by averaging the values for each component of every item belonging to the cluster. The distance between two clusters in the space is represented by Euclidean distance [11]. Suppose the centroids for two clusters are $(x_{i1}, x_{i2}, ..., x_{in})$ and $(x_{j1}, x_{j2}, ..., x_{jn})$, then the distance d between these two clusters is defined as $d = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$. To develop our algorithm, we describe two popular clustering techniques: minimum spanning tree method and k-means method [11].

The minimum spanning tree method starts with N clusters (each cluster having one item) and then merges clusters with the shortest distance until the desired number of clusters are left. One problem in using the minimum spanning tree method is the demand on main memory and computational power. The computational complexity for this method is $O(N^2)$. N could be quite large for a busy website (or a long trace), making it difficult to use this method. In this case N is in the order of 10,000, but it can be significantly reduced by combining sessions with the same set of dimensions. Since the analysis is done off-line, with the combined sessions generating a reduced input size, the computation time can be tolerated. Another problem encountered using this method is that it tends to result in one large group due to the existence of outliers. It is very difficult to remove these outliers before hand, and as such, it is difficult to obtain the desired number of session groups.

The k-means method selects k centroids and then merges clusters to the nearest centroid. It has a computational complexity of $O(N)$ and makes much less demand on main memory than the minimum spanning tree method. Thus, this method is more suitable when N is large. Previous research has used this method to group sessions [1, 13]. It is difficult to choose the right centroids and the right value for k.

The clustering algorithm implemented in this research combines both methods. First, the minimum spanning tree method is applied to select the particular number of centroids for the next stage, then sessions are grouped according to these centroids. Combining both methods works well for this data set. The minimum spanning tree method identifies centroids representing a small number of group types. A session group will likely form around each selected centroid, preventing the clustering from ending up with one or two large groups.

For many of the sessions, the coordinates for a large number of the dimensions (metrics) used for the grouping algorithm have the value 0. This is especially true for the navigation pattern method, which has 324 dimensions (18×18, to indicate transitions between any 2 distinct web page groups in Table 1, adding the *Home* and *Exit* states.). Thus, a matrix representation of the state space would be extremely sparse. In order to reduce the demand on main memory and computational complexity of the spanning-tree portion of the algorithm, the zero coordinates are not represented explicitly; only non-zero dimensions are recorded. Two lists are used for each item: one for the dimensions and the other for the coordinates. Sessions with the same set of non-zero dimensions were merged beforehand to reduce computation for the minimum spanning-tree method.

The general algorithm is as follows:

Algorithm *groupSession*:
Input: N sessions, s desired number of clusters for spanning tree method, k desired number of final clusters
Output: data representing those k clusters of sessions
Let:
  $(x_1, x_2, ..., x_n)$ be the completed set of selected metrics representing a session
  $(x_{m_1}, x_{m_2}, ...x_{m_m})$ be the set of non-zero metrics for a specific session ($m \leq n$, and, $(x_{m_1}, x_{m_2}, ...x_{m_m}) \subseteq (x_1, x_2, ..., x_n)$)
  $(c_{x_{m_1}}, c_{x_{m_2}}, ..., c_{x_{m_m}})$ be the non-zero coordinate for a specific session, $c_{x_{m_i}}$ is the co-ordinate at the dimension of x_{m_i} (i = 1, 2, ..., m)
1) Represent each session with:
   $(x_{m_1}, x_{m_2}, ...x_{m_m})$ and $(c_{x_{m_1}}, c_{x_{m_2}}, ..., c_{x_{m_m}})$
2) Initiate each session to be a cluster, mark all clusters to be active
3) For each active cluster i
   check all remaining active clusters j
   if cluster j is the same as cluster i
   merge j to i and mark it as inactive
4) Repeat until there are only s desirable numbers of active clusters left
   From all active clusters
   find the pair of active clusters with the smallest distance (*findDistance()*)
   merge them to get a new cluster and mark one of them as inactive (*mergeClusters()*)
5) Manually select k biggest clusters as the centroids
6) For each unattached cluster
   group the clusters to the nearest chosen centroids

Two other algorithms are used in this process of determining session groups: *findDistance* calculates the distance between two clusters, and *mergeClusters* merges two clusters, recalculating the new centroid. These two algorithms are not shown in the paper, due to space considerations. When grouping by resource usage, calculating the distance between clusters and merging clusters are straightforward since there are only two metrics. Thus only the algorithm *groupSession* was used in this case.

5. IDENTIFIED SESSION GROUP TYPES

The number of session groups that are chosen is a matter of subjective evaluation. The algorithm begins with one group per session, but that is an unreasonable number of sessions, and provides no insight into patterns between sessions. The other extreme is to group all sessions into one group, but this also provides no appreciable insight. Previous research has indicated that a small number of groups

Table 2: Services requested in a session

List	hom	gSrv	info	loc	prmt	rChR	rCnl	rHom	rMkR	rPpU	rQut	rVew	trvl	vhcl
srvG-0	0.006	-	0.017	0.003	0.004	1.165	0.001	0.066	-	0.095	1.019	-	0.004	0.01
srvG-1	0.025	0.423	0.111	0.294	0.457	1.671	0.008	1.941	0.05	2.141	1.744	0.045	0.105	0.534
srvG-2	0.074	0.316	0.456	0.603	0.173	0.048	0.117	0.569	0.051	1.418	0.015	0.218	0.069	0.213
srvG-3	0.129	0.827	1.946	0.374	0.702	1.882	0.153	2.441	0.163	2.073	2.406	1.243	0.334	0.5
nvgG-0	-	0.026	0.089	0.014	0.074	1.428	0.006	0.192	0.009	0.14	1.187	0.038	0.026	0.059
nvgG-1	0.068	0.63	0.697	0.047	0.35	0.652	0.045	2.214	0.039	0.6	2.299	0.442	0.169	0.357
nvgG-2	0.087	0.381	0.652	0.739	0.437	0.794	0.129	1.162	0.076	0.511	0.386	0.352	0.133	0.385
nvgG-3	0.029	0.475	0.34	0.284	0.315	1.54	0.019	1.549	0.088	5.26	1.412	0.205	0.09	0.469
resG-0	0.001	-	0.002	-	-	0.888	-	0.002	-	-	1.026	-	0.001	-
resG-1	0.076	0.343	0.293	0.512	0.33	0.636	0.072	0.959	0.062	1.732	0.174	0.144	0.093	0.339
resG-2	0.007	0.207	1.026	0.069	0.157	1.48	0.012	1.946	0.007	0.569	3.334	0.11	0.072	0.164
resG-3	0.035	0.539	0.703	0.235	0.43	1.982	0.071	1.801	0.078	1.846	2.153	0.551	0.169	0.446

(less than 10) is a useful number. This provides some level of discrimination, with the ability of the site to respond to the different needs of a small number of groups. The algorithm was run with values of k = 3, 4, and 5 respectively. It was found that 4 was the ideal number of groups, since fewer groups created additional intra-cluster variance, and more groups resulted in two of the groups having quite similar characteristics. Other E-commerce sites would be different in the number of groups identified, depending on the functionality and usage patterns.

Each of the three session grouping methods independently identified a set of four session groups. Thus, there are three sets and a total of 12 different overlapping groups (Table 2). Groups in the same set were named a common prefix, which indicates the metric used for grouping. The prefixes srvG, nvgG and resG are corresponding to metric of services requested, navigation pattern, and resource usage (MSRT), respectively. Table 2 lists what services a session group requests and the average frequency for a session to request a service. Only the more frequently visited services are listed in this table, as several of the services had no frequency greater than 0.12. From this table we can see the focus of a session group and unique characteristics.

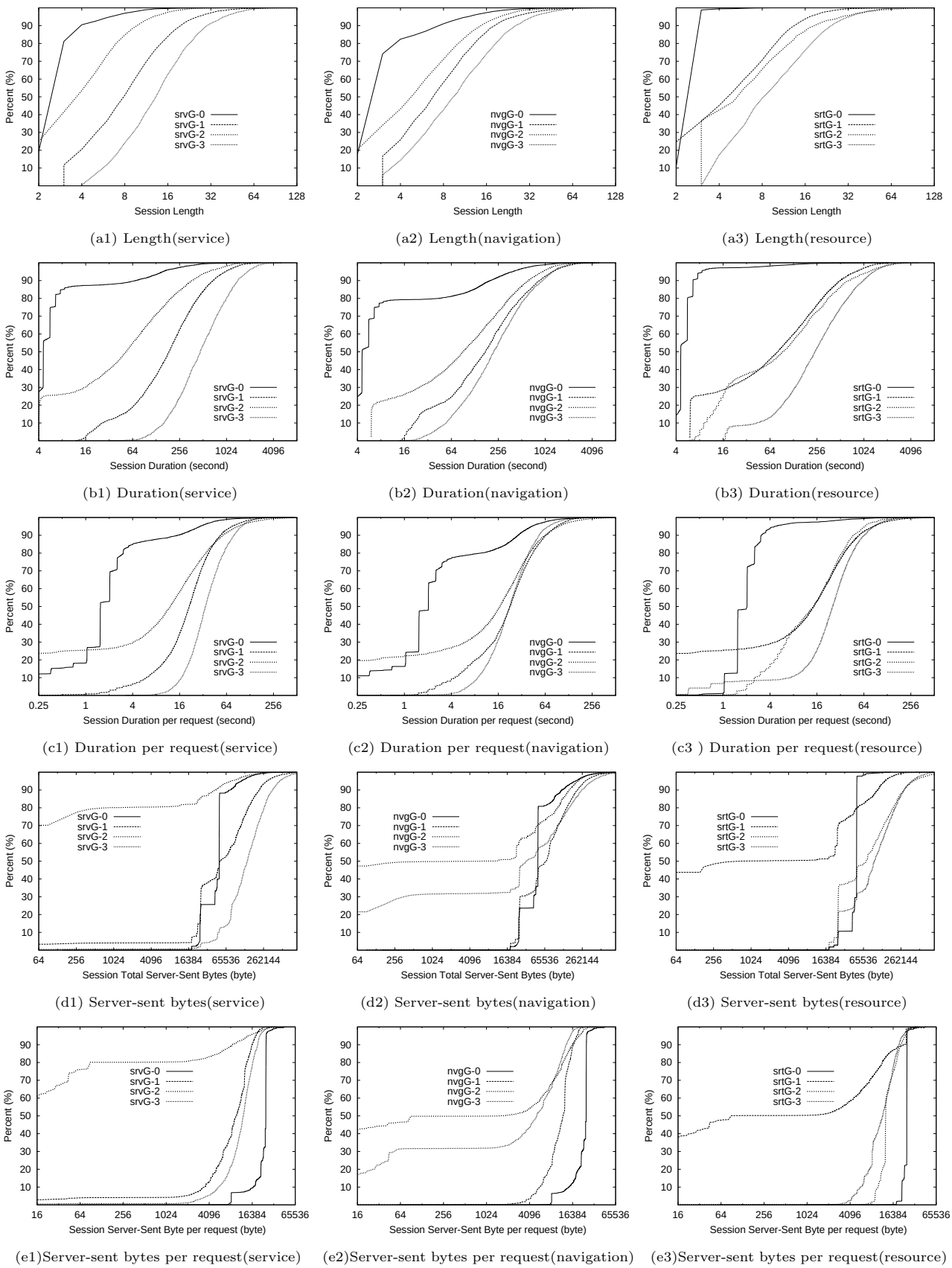
Further details which provide quantitative support for these groupings are given by the Cumulative Distribution Functions (CDF) of session length, duration, and server-sent bytes (Figure 4). The CDFs for the total and per-request client-sent bytes are not shown in these figures since per-request client-sent bytes distributes in the narrow range of 350 to 1024 bytes for all session groups. Total client-sent bytes for a session is proportional to session length.

From the total of 12 different overlapping groups, four session group types are identified. The characteristics of these groups are described as follow:

1. Rate-Checkers. This group is best represented by session group resG-0. Customers in this group did a quick rate-checking and left. Their activities involve mainly two states, checking rate (rChR) and getting the result (rQut). These users pay very little attention to other relevant information (Table 2). Thus sessions are very short, 88% of all sessions in this group are exactly 3 requests in length (average length: 2.9). (Figure 4 (a3)). The duration per request is very small (average 2.5 seconds; with 90% less than 3 seconds) (Figure 4 (c3)) since customers were very clear on what they need and how to do achieve their goals. Thus, 97% of the sessions are less than 9 seconds (Figure 4 (b3)). The server-sent bytes per request for this group types is the highest since check-rate results are sent in such a short session(Figure 4 (e3)) . A careful examination of this group revealed that the same organization was responsible for nearly all of the sessions. This organization was a search facility, which examines many similar sites to compare prices for the same item. These sessions did not request images from the server and entered at the reservations home page, which is step 4 of the 5-step process in some rental sites.

2. Browsers. These customers are best represented by srvG-2. Customers in this group did not show a focused objective. They mainly did searching, checking out relevant information (states: rPpU, loc, info, etc.)(Table 2). Compared to the other groups, customers in this group did very little rate-checking, with each session visiting the state rChR only 0.048 times on average. Many customers who wanted to cancel their reservations were also in this group. About 40% of the sessions in this group are not longer than 3 requests (Figure 4 (a1)), but 10% of the sessions are longer than 10 requests (average session length: 5.6). This group also has the smallest server-sent bytes in total and per request, as can be seen in Figure 4 (d1) and (e1), due to the lack of rate-checking, which involves a large number of server-sent bytes.

3. Confirmers. This group is best represented by resG-2. Customers in this group showed interest in car reservation. They stayed much longer, about 20% of the sessions are longer than 12 requests (average session length: 10.2) (Figure 4 (a3)). They searched for more detailed information, checking out available vehicles and sales promotion, and checking rate (most visited states: info, rchR, rQut, vhcl, etc.)(Table 2). However, the percentage of customers who made reservations is very low, the average frequency of making reservation (visiting state: rMkR) in a session is 0.007. About 36% of the sessions have exactly 3 requests and exactly the same number of server-sent bytes (Figure 4 (a3) and (d3)). Further examination shows that these sessions identically requested for the states rHom and rQut.



(a1) Length(service) (a2) Length(navigation) (a3) Length(resource)

(b1) Duration(service) (b2) Duration(navigation) (b3) Duration(resource)

(c1) Duration per request(service) (c2) Duration per request(navigation) (c3) Duration per request(resource)

(d1) Server-sent bytes(service) (d2) Server-sent bytes(navigation) (d3) Server-sent bytes(resource)

(e1)Server-sent bytes per request(service) (e2)Server-sent bytes per request(navigation) (e3)Server-sent bytes per request(resource)

Figure 4: CDFs for session group characteristics

4. Buyers. They are best represented by srvG-3. Customers take their time in getting detailed information, searching, checking rate and making reservation (states: info, prmt, gSrv, rchR, PupU, rView, etc.) as seen in Table 2. There are many relatively long sessions. The median for session length is about 9. The average session length is 16.7. The duration per request for this group is the longest, with a median of 32 seconds (Figure 4 (c1). The combination of long session length and long duration per request produces a long session duration (Figure 4 (b1)). The percentage of sessions making a reservation is much higher than other session groups, with an average frequency of 0.163 visiting the state rMkR in a session, as shown in Table 2.

6. SESSION GROUPING METHODS

Table 3 demonstrates the group sizes and the association between session groups of different criteria. The smallest group (resG-2) has 7.3% of the total number of sessions, and the largest one (resG-1) has 40.3% number of sessions. The association between two groups is demonstrated by the degree they are overlapped. The more two groups are overlapped, the higher degree they are associated. An entry in the group association part of Table 3 shows what percentage of the session group in the same row are also contained in session group in the same column. For example: 95.5% of sessions in group srvG-0 are also contained by group svgG-0, and 88.2% of svgG-0 is in srvG-0.

Based on Table 3, we can see how well a grouping method works in identifying the four group types, and how the results from different grouping methods are related.

- Services: Browsers and Buyers are identified the best (i.e. group srvG-2 and srvG-3). As well, Rate-checkers are contained in srvG-0 (99.6% of resG-0 is contained within srvG-0, 69.8% of srvG-0 are sessions of resG-0). A large portion of Confirmers (70.6%) is in srvG-1. The four groups, srvG-0, 1, 2 and 3 are matched, one to one, with the four identified group types.

- navigation: Rate-checkers are mainly contained in nvgG-0 (98.3% of resG-0 are contained by nvgG-0, 63.5% of srvG-0 are sessions of resG-0); A big portion of Browsers (73.8% of srvG-2) are contained by nvgG-2; A big portion of Confirmers (72.1% of resG-2) are contained by nvgG-1. The Buyers group distributes mainly in nvgG-1, 2 and 3. Thus, this clustering is not as good as the one done by services requested.

- resource usage: Rate-checkers and Confirmers are best identified (i.e. resG-0 and resG-2). Almost all of the Browsers group (97.1%) are contained in group resG-1, and a large portion of Buyers (71.2%) are in group resG-3. Again, the four groups, resG-0, 1, 2 and 3 are matched, one to one, with the four previously identified group types.

All three session grouping methods identify, to some extent, the four kinds of groups described above, except that the serious-buyer group is not well identified by navigation pattern. Figure 4 also demonstrates that grouping by these three methods yield very similar results. CDFs for a set of groups (srvG-0, nvgG-0, resG-0) are similar in shapes, indicating that these three groups are highly overlapped. It is the same case for session group sets (srvG-1, nvgG-1, resG-2), (srvG-2, nvgG-2, resG-1) and (srvG-3, nvgG-3, resG-3).

These three methods are, however, different in some ways. Grouping by navigation is complicated since there are too many coordinates involved (324 coordinates in this case). The computation complexity may affect the clustering results since clustering is based on the distance between sessions and the distance is determined by coordinates. This may explain why the serious buyer group was not well identified by this method.

The issue in grouping session by resource usage is that it is difficult to obtain resource usage data. In this case the resource usage data are approximated based on http logs. A reasonable estimate of MSRT in this case since the server was not busy, but it is hard to verify its accuracy.

Grouping by service requested is much simpler than grouping by navigation. Unlike grouping by resource usage, there are no uncertain approximations involved. Grouping by services works well in identifying session groups. The serious-buyer group is best isolated by this method.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we analyzed the customer behaviour of an E-rental business with respect to user sessions. The goal was to determine appropriate characteristics for an analyst to consider in order to distinguish between customer groups.

We introduced a hybrid clustering algorithm, which combines the features of the k-means clustering and minimum spanning tree methods. This algorithm is more efficient in time and space than the minimum spanning tree and provides the same results as k-means clustering, with appropriate values of k.

We found that session groups obtained by the three selected criteria are highly associated. and yield similar results. This is reasonable since services, navigation pattern and resource consumption are highly associated. Sessions with the same navigation pattern want the same service and consume similar amounts of server resources; users requesting the same services tend to have similar navigation pattern and resource consumption; similar resource consumption correlates with similar service and navigation pattern. The performance implication is that one can group sessions in one of these ways and use the result to analyze issues related to service, navigation pattern and resource usage, such as QoS, server resource management and scalability. Grouping by services requested is recommended since this method is relative simple and data on requested services are easy to obtain.

Grouping customers and analyzing session groups can be viewed as an advance workload characterization for an E-commerce site. Dividing sessions into groups is the first step. Session groups obtained using methods discussed in this paper can be further characterized. For example, a CBMG can be obtained for each session group. Ultimately, workload characterization is used for analyzing system performance-related issues.

Several assumptions were made about the nature of the interaction that a user makes with the website. The specific characteristics of the structure of the system under test enabled the analysis of web page classification to be done rather easily. This does not restrict the usefulness of this

Table 3: Degree of Association among session groups

	Group Size		Group Association											
Group	NumOf session	Percent (%)	srvG-0	srvG-1	srvG-2	srvG-3	nvgG-0	nvgG-1	nvgG-2	nvgG-3	resG-0	resG-1	resG-2	resG-3
srvG-0	5605	33.9	100	-	-	-	**95.5**	2.1	1.43	0.7	**69.8**	12.5	3.8	13.8
srvG-1	4898	29.7	-	100	-	-	9.47	34.1	26.0	30.4	0.3	34.4	17.5	**47.8**
srvG-2	3901	23.6	-	-	100	-	0.2	0.3	**73.8**	25.8	0.1	**97.1**	0.4	2.5
srvG-3	2107	12.8	-	-	-	100	11.7	39.1	25.2	24.0	0	22.7	6.0	**71.2**
nvgG-0	6071	36.8	**88.2**	7.6	6.1	4.1	100	-	-	-	**63.6**	12.3	3.7	20.4
nvgG-1	2623	15.9	4.4	**63.7**	0.4	31.4	-	100	-	-	0.3	4.7	33.3	**61.7**
nvgG-2	4763	28.8	1.7	26.8	**60.4**	11.1	-	-	100	-	1.3	**84.4**	1.1	13.2
nvgG-3	3054	18.5	1.8	**48.7**	32.9	16.6	-	-	-	100	0	**57.6**	20.0	40.4
resG-0	3929	23.8	**99.6**	0.4	0	0	**98.3**	0.2	1.5	0	100	-	-	-
resG-1	6652	40.3	10.6	25.3	**56.9**	7.2	11.2	1.8	**60.4**	26.5	-	100	-	-
resG-2	1212	7.3	17.6	**70.6**	1.2	10.5	18.4	**72.1**	4.5	5.0	-	-	100	-
resG-3	4718	28.6	16.4	**49.6**	2.1	31.8	26.2	34.3	13.3	26.1	-	-	-	100

approach in general, but obtaining the data in proper form may be more difficult.

This study compares metrics used in clustering user sessions and is based on analysis of traces from only one E-commerce website. Currently, we are engaged in similar analysis on additional E-commerce sites, and more general purpose sites to verify our findings and methodology.

8. REFERENCES

[1] M. Arlitt, D. Krishnamurthy, and J. Rolia. Characterizing the Scalability of a Large Web-based Shopping System. *ACM Transactions on Internet Technology (TOIT)*, 1(1):44–69, 2001.

[2] A. Banerjee and J. Ghosh. Clickstream Clustering Using Weighted Longest Common Subsequences. In *Proceedings of the Web Mining Workshop at the 1st SIAM Conf. on Data Mining*, pages 34–40, Chicago, IL, April 2001.

[3] Refsnes Data. ASP Sessions Object. www.w3schools.com/asp/asp_sessions.asp.

[4] V. Estivill-Castro and J. Yang. Categorizing Visitors Dynamically by Fast and Robust Clustering of Access Logs. *Lecture Notes in Computer Science*, 2198:498+, 2001.

[5] Y. Fu, K. Sandhu, and M. Shih. Fast Clustering of Web Users Based on Navigation Patterns. In *World Multiconference on Systemics, Cybernetics and Informatics (SCI/ISAS'99)*, pages 560–567, Orlando, FL, August 1999.

[6] Y. Fu, K. Sandhu, and M. Shih. A Generalization-Based Approach to Clustering of Web Usage Sessions. In *Intl. Workshop on Web Usage Analysis and User Profiling (WEBKDD'99)*, pages 21–38, San Diego CA, August 1999.

[7] Y. Fu and M. Shih. A Framework for Personal Web Usage Mining. In *Intl Conf. on Internet Computing (IC'2002)*, pages 595–600, Las Vegas, NV, June 2002.

[8] J. Heer and E. Chi. Identification of Web User Traffic Composition Using Multi-Modal Clustering and Information Scent. In *Proc. of the Workshop on Web Mining, SIAM Conf. on Data Mining*, pages 51–58, Chicago, IL, April 2001.

[9] J. Heer and E. Chi. Mining the Structure of User Activity using Cluster Stability. In *Proc. of the Workshop on Web Analytics, SIAM Conf. on Data Mining*, Arlington, VA, April 2002.

[10] J. Heer and E. Chi. Separating the Swarm: Categorization Methods for User Sessions on the Web. In *Proc. of the SIGCHI Conf. on Human factors in Computing Systems*, pages 243–250, Minneapolis, MN, April 2002.

[11] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc, 1992.

[12] L. Kaufman and P. Rousseeuw. *Finding Groups in Data*. John Wiley & Sons, Inc, 1990.

[13] D. Menascé, V. Almeida, R. Fonseca, and M. Mendes. A Methodology for Workload Characterization of E-commerce Sites. In *Proc. of the 1st ACM Conf. on Electronic Commerce*, pages 119–128, Denver, CO, November 1999.

[14] D. Menascé, V. Almeida, R. Fonseca, and M. Mendes. Business-Oriented Resource Management Policies for E-commerce Servers. *Performance Evaluation*, 42:223–239, May 2000.

[15] D. Menascé, V. Almeida, R. Riedi, F. Ribeiro, R. Fonseca, and W. Meira Jr. In Search of Invariants for E-business Workloads. In *Proc. of the 2nd ACM Conf. on Electronic Commerce*, pages 56–65, Minneapolis, MN, October 2000.

[16] C. Shahabi, A. Zarkesh, J. Adibi, and V. Shah. What Knowledge Discovery from User's Web-page Navigation. In *the IEEE Intl. Workshop on Research Issues in Data Engineering (RIDE)*, pages 20–29, Birmingham, UK, April 1997.

[17] Q. Wang, D. Makaroff, K. Edwards, and R. Thompson. Workload Characterization for an E-commerce Web Site. In *CASCON 2003*, pages 68–82, Toronto, Canada, October 2003.

[18] J. Xiao, Y. Zhang, X. Jia, and T. Li. Measuring Similarity of Interests for Clustering Web-Users. In *Proc. of the 12th Australasian Conf. on Database Technologies*, pages 107–114, Queensland, Australia, 2001.