December 2004

Clustering Web Sessions Using Extended General Pages

Zhongming Ma
University of Utah

Olivia Sheng
University of Utah

Clustering Web Sessions Using Extended General Pages

Zhongming Ma
University of Utah
David Eccles School of Business
zhongming.ma@business.utah.edu

Olivia R. Liu Sheng
University of Utah
David Eccles School of Business
olivia.sheng@business.utah.edu

Abstract

We study Web sessions clustering in order to find groups of similar sessions and discover user access patterns on a Web site. We extend the general page concept presented in (Fu, Sandhu and Shih 2000) by including partial document names and dynamic pages, and use an extended general page (EGP) to represent many individual page URLs sharing the same EGP. We present two extensions of a hierarchical clustering algorithm, ROCK (Guha, Rastogi and Shim 2000). One is a notion of EGP count that we add to the session similarity calculation. The other is a goodness threshold we adopt to restrict certain clusters from merging with others. Further, we propose a set of measurements for assessing the results from clustering boolean and categorical data and help users to identify their desired clustering results. In our experiments, we applied the ROCK and the extended ROCK (E-ROCK) algorithms to cluster a half-month's Web log from a customer service Web site at HP. The experiment results showed that E-ROCK alleviated a large cluster problem of the ROCK algorithm and improved the performance in intra cluster similarity.

Keywords: clustering, clustering validity, Web session clustering, Web usage mining, general page

1. Introduction

The Web has become a space where people communicate through the Internet without restriction of access time or limitation of geographical location. Companies, organizations, governments and individuals gained and share information and knowledge by visiting Web sites and viewing Web pages and (Menasalvas et al. 2003) claims that it is possible to say that the Web is becoming one of the main communication channels for any kind of transaction.

Web usage mining is the application of data mining techniques to discover usage patterns from Web usage data (Srivastava, Cooley, Deshpande and Tan 2000). In a Web usage log file, one type of Web usage data, log records are generated by a Web server each time a Web site is accessed. A Web session consists of a series of sequentially visited Web page URLs with which a client interacts through a Web server over a period of time (Avedal et al. 2000). In order to identify a Web session, a session timeout is needed. The log file can record certain information during the sessions, such as client IPs and visited page URLs.

Clustering is a useful data mining technique for discovering groups of similar objects and for identifying interesting distinctions and patterns in the underlying data. Ideally, data points within a cluster are more similar to each other than to those in different clusters (Guha, Rastogi and Shim 1998). Clustering Web sessions is a promising approach to discovering Web usage patterns and inferring user interests (Heer and Chi 2002).

Owners of Web sites realize that the usability of a site can substantially influence the success of a business. Identifying and understanding Web usage patterns enables webmasters and content producers to improve Web design and usability so as to tailor sites to user needs and to enable marketers to know user interests more closely in order to post better sale promotions and advertising (Heer and Chi 2002). When dealing directly with individual page URLs, it is hard to find sufficient number of sessions during which users visit common pages because there are many Web pages in a site (Fu, Sandhu and Shih 2000) and during each session the user usually visits only a few pages. Thus these authors present a general page concept in order to find groups of sessions with similar access patterns. We extend the general page concept to include more specific but still high-level concepts. In addition, we present an extended ROCK (E-ROCK) algorithm to cluster Web sessions and identify visit patterns from those clusters based on the extended general page concept.

Our contributions are as follows.

(1) Extended general page
We extend the general page concept (Fu et al. 2000) by including partial document names and dynamic pages, and then use an extended general page (EGP) to represent many individual page URLs that share the same EGP. With these two extensions, the EGPs not only cover original general pages, but also include more specific but still high-level concepts.

(2) Count consideration in similarity function
When computing the similarity between a pair of sessions, we consider the count of an EGP in a session whereas Guha et al. (2000) did not because they deal with market basket data. A session that originally consists of a series of page URLs therefore becomes a set of distinct EGPs and the corresponding counts/weights of the EGPs.

(3) Adding a goodness threshold alleviates one large cluster problem and improves intra similarity
When the data set contains many outliers, ROCK tends to generate a very large cluster that contains most of the data points and a very large number of outlier clusters, each of which is very small. The problem with one large cluster is that its intra similarity is low. With a goodness threshold, the E-ROCK alleviates the one large cluster problem and achieves higher intra cluster similarity.

(4) Cluster evaluation criteria
A good clustering algorithm does not necessarily generate good clustering results because the results depend on input parameters and the users do not know the proper set of parameters beforehand. We propose cluster intra-similarity and inter-dissimilarity measurements for categorical data to evaluate clustering results generated from using different sets of input parameters. Those measurements can help users to identify their desired clustering results.

2. Previous work

The clustering technique has been extensively studied in and applied for automatically identifying groups of similar objects in many areas such as statistics, computer science, marketing and biology. With the explosive growth of the Web, the study of clustering Web usage data has become popular (Wang and Zaïane 2002). Web session information is a kind of categorical attribute (Foss, Wang and Zaïane 2001). Thus many clustering algorithms, such as K-means (MacQueen 1967), are not suited for clustering sessions. One reason is that commonly used distance-based similarity measurements such as

the Euclidean distance become improper because it cannot distinguish the difference between two data points that differ on few attributes and between two data points that differ by small amounts on individual attributes (Guha, Rastogi and Shim 1997).

In clustering Web usage log, a type of Web usage data, many studies have dealt directly with individual page URLs when computing a similarity between two sessions (Wang and Zaïane 2002; Joshi and Krishnapuram 2000; Heer and Chi 2002). However, there are some problems in handing individual URLs.

(1) High dimensionality in page URLs and small number of similar sessions
Before computing the similarity of a pair of sessions, we need first to find the similarity between a pair of pages, because a session consists of visited pages. A large Web site usually holds thousands, even millions of pages (Fu et al. 2000), but the average number of page URLs visited in a session is small. Therefore, when representing visited URLs as categorical data in a session-URL matrix, the matrix is extremely sparse. The high dimensional space of URLs and the small number of sessions makes it very difficult to find similar sessions that share certain common URLs. This problem, also called the curse-of-dimensionality, produces either small clusters or clusters with very low intra similarity, thus failing to represent user behavior properly (Fu et al. 2000).

(2) Page similarity problem
Some studies, such as (Wang and Zaïane 2002) and (Joshi and Krishnapuram 2000), do not handle page URLs as categorical data. The similarity between a pair of page URLs is measured by common directories in the paths of the URLs. For example, the path for a URL, *home/Support/PAT/ECS_00017.html*, is *home/Support/PAT/*, and this path consists of three directories, *home*, *Support* and *PAT*. In order to compute page similarity using common directories, Wang and Zaïane assign a weight of 2^n to the first directory in the path, and a weight of 2^0 to the document (n = the number of directories in the longer path). Joshi and Krishnapuram assign a uniform weight to each directory in the path. The problem with using common directories to calculate page similarity is that a partial overlap in paths does not necessarily reflect true similarity between two pages. For example, for two page URLs, *home/Support/PAT/ECS_00017.html* and *home/Support/ENOT/OV-EN011570.html*, according to the similarity function used in (Wang and Zaïane 2002), their similarity is 12/15, and 2/3 according to Joshi and Krishnapuram (2000). However, the first page contains patch information for a software product, and the second page is a technical document for a network product. They are neither similar in content nor similar in use. Thus, sharing partial directories on paths between two page URLs does not accurately represent similarity between the two pages.

Fu et al. (2000) represents a Web site as a page hierarchy that consists of leaf nodes and non-leaf nodes. A leaf node, also called a simple page, represents an individual page URL, such as *http://www.umr.edu/~regwww/ugcr97/ee.html*. For this URL, the non-leaf node, also called a general page, is *http://www.umr.edu/~regwww/ugcr97*. Sessions are then represented not by simple pages, but by general pages and a hierarchical clustering algorithm, BIRCH (Zhang, Ramakrishnan and Livny 1996), is applied to the generalized session space.

Similarly, Banerjee and Ghosh define concept-categories to be first-level branches from the home page of a Web site which they examine. Then all pages under the same category are considered to have the same concept. Their log data contain a total of 453,953 accessed pages, but they do not mention how

many of them are unique (Banerjee and Ghosh 2001). If the number of unique pages is large, a problem is that having a small number of concepts is too coarse and cannot precisely represent a large number of pages considering a the Web site they study has fewer than 20 concept-categories.

Heer and Chi (2002) evaluate session categorization methods by considering different data features and using different weighting schemes. They study several types of data features as follows. The content of a Web page is converted into a TF.IDF vector; a URL is tokenized using delimiters such as "/" and "&" and links on a page are represented as Outlink vector or Inlink vector. Four basic weighting schemes include: uniform, TF.IDF, position and view time. By combining data features and combining weight schemes as well as representing a session by the combined data feature and weight scheme, they studied a total of 320 different scheme combinations based on 104 user sessions on www.xerox.com. Finally, one of their suggestions is that simple schemes, such as raw path + visit time or URL token + visit time give good results in categorization accuracy (Heer and Chi 2002). However, their results are based on a small number of users and sessions.

In this paper, we extend a robust clustering algorithm, ROCK (Guha et al. 2000), to cluster Web sessions and discover visit patterns by cluster. ROCK is an agglomerative hierarchical clustering algorithm for boolean and categorical attributes. A traditional distance measure is not proper for these data types. Guha et al. propose a novel concept of links to measure the similarity between a pair of data points. When the similarity of a pair of points, measured by Jaccard coefficient, exceeds a certain threshold, the points are neighbors. The number of common neighbors of two data points is the number of their links. Therefore, the link concept incorporates global information about other points in the neighborhood of the two points, while a similarity based on a distance between two points alone considers only the two points in question. The algorithm maximizes the sum of links(p_i, p_j) for data points p_i, p_j that belong to the same cluster, and minimizes the sum of links(p_i, p_j) for p_i, p_j in different clusters (Guha et al. 2000).

An important issue in cluster analysis is the evaluation of the clustering results to find the partitioning that best fits the underlying data (Halkidi, Batistakis and Vazirgiannis 2001). However, good clustering algorithms do not necessarily generate optimal clustering results because the results often depend on proper input parameters, such as the number of clusters. Improper input parameters may result in clustering results that do not represent real groups and patterns, leading to wrong decisions.

Halkidi et al. (2000) propose an approach to validation of clustering schemes in order to find the best number of clusters. They use quality indices, average scattering for clusters and total separation between clusters, to find the best compactness and separation of clusters (Halkidi, Vazirgiannis and Batistakis 2000). Their approach is suited to non-categorical data, and is not proper for us because Web sessions are of categorical (Foss et al. 2001).

3. Our Approaches

For the convenience of readers, we list notation used through this paper in table 1.

P_i – extended general page (EGP)	n_{c_p} - number of sessions in cluster C_p
S_i – session, $S_i = \{P_1: W_1, P_2: W_2, \ldots, P_n: W_n\}$ W_j is the count/weight for EGP P_j, j = 1, …, n, and n is the number of unique EGPs in the session	$n_{c_p}^{large}$ - number of large EGPs in C_p

$Sim(S_i, S_j)$ – similarity between sessions S_i and S_j	$Intra(C_p)$ – intra similarity of cluster C_p
W_{S_i} - weight vector for EGPs in session S_i $W_{S_i} = (W_1, W_2, \ldots, W_n)$	$Inter(C_p, C_q)$ – inter dissimilarity between clusters C_p and C_q
C_p - cluster C_p	C - collection of clusters, $\|C\|$ is total number of clusters
$W(C_p)$ - weight vector for EGPs in C_p	n_c - number of sessions in a collection of clusters
$large(C_p)$ - large EGPs vector in C_p	. is a product operation between two vectors

Table 1 Notation summary

3.1 The Concept of EGP

We extend the general page concept in (Fu et al. 2000) by including partial document names and dynamic pages. If the documents follow a naming scheme and there is a large portion of visited documents under a general page, we include the partial document names into general pages to derive EGPs. In this study, we used a half-month of Web log from a Web site at Hewlett Packard (HP). In our cleaned data set, the top six largest general pages own 72.1% of the total document clicks. Clustering the Web sessions using general pages will discover that the some of the six largest general pages appear frequently in some clusters. The result may not be very helpful for Web designers or an online recommendation system as each of the general pages may represents thousands of different documents and therefore is not specific enough. Since our documents follow a naming scheme, after extending the six largest general pages, we obtain 26 EGPs. For example, under */support/PAT/,* one of the six largest general pages, all documents related to *Network Node Manager* have names beginning with *NNM*. For instance, from a page URL, */Support/PAT/NNM_01008.html*, we can obtain its EGP, */support/PAT/NNM*. In addition, if dynamic pages are used with parameters to retrieve similar Web pages, they are analogous to general pages. So we treat the dynamic pages without use of parameters, such as */home/svi_support_contract.jsp,* as EGPs.

Now a session is represented not by visited individual URLs, but by a set of distinct EGPs and their corresponding counts/frequencies for these EGPs. For example, a session S_i that originally consists of three page URLs, is *{/home/products/network/network_security/*s_1*.htm, /home/products/development/ development_tookkit/*d_1*.htm,* and */home/products/network/network_security/*s_2*.htm}*. With the EGP concept, the session now is expressed as S_i = *{/home/products/network/network_security 2, /home/products/development/development_ toolkit 1}*, where the number after an EGP is the count for how many times the user visited the EGP in the session. Therefore, when similar sessions are grouped in a cluster, we can discover which EGPs tend to be accessed together, and how many times they are accessed during a certain time period. Since dividing documents into separate directories organized in an easily understood hierarchy is a basic rule of Web design and file organization (21), we consider this session representation to be helpful because the EGP can represent a higher-level, but still specific, concept, while individual page URLs cannot. Moreover, this representation dramatically reduces high dimensionality in page URLs.

3.2 EGP Count in Session Similarity

In ROCK (Guha et al. 2000), when computing similarity between two data points with Jaccard coefficient, the count of each attribute is ignored because ROCK studies item co-occurrence in market

basket data where the count of an item can be ignored. For example, for two sessions S_1 = *{a: 2, b: 1, c: 2}* and S_2 = *{a: 1, c: 2, d: 2}*, without considering counts of EGPs, their similarity is $Sim(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} = \frac{2}{4}$. However, when applying it to Web sessions, the count on an EGP or on an individual page should be taken into consideration because the count reflects and affects an access pattern. In our approach, there is a count associated with an EGP in a session, and we treat the count as a weight attached to its associated EGP. So a session consists of a set of EGPs with their associated weights, $S_i = \{P_1: W_1, P_2: W_2, \ldots, P_n: W_n\}$. A corresponding weight vector is $W_{S_i} = (W_1, W_2, \ldots, W_n)$. Thus, we extend the session similarity expression to include the weights and define the similarity between a pair of sessions as $Sim(S_1, S_2) = \frac{|W_{S_1} \cap W_{S_2}|}{|W_{S_1} \cup W_{S_2}|}$. Now the similarity between S_1 and S_2 is

$$Sim(S_1, S_2) = \frac{|W_{S_1} \cap W_{S_2}|}{|W_{S_1} \cup W_{S_2}|} = \frac{1+2}{2+1+2+2} = \frac{3}{7}$$

3.3 Goodness Threshold

ROCK uses a similarity threshold, θ, to determine whether a pair of data points are considered neighbors. In ROCK the same θ is also used to calculate goodness of a pair of clusters, g = $\frac{links[C_i, C_j]}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}}$ where *links[C_i, C_j]* is number of cross links between clusters C_i and C_j, n_i is the size of cluster C_i and $f(\theta) = \frac{1-\theta}{1+\theta}$. If the goodness of two clusters is larger than zero, ROCK will designate one cluster a candidate to be merged with the other. However, when a data set contains many outliers, a problem is that it tends to generate a very large cluster that contains most of the data points, causing low intra cluster similarity, and a large number of outlier clusters, each of them very small.

Zaïane et al. (2002) noticed the problem in their experiments. When clustering t7.10k.dat and choosing 9 as the number of clusters provided, they obtained one large cluster containing 9,985 of total 10,000 data points, while the remaining 15 data points existed in the eight noise clusters. A cluster is considered noise if its size is less than a threshold. When they set the number of clusters at a thousand, 995 of them were noise. A cluster is considered a noise or outlier if its size is less than a threshold. They concluded that it was because ROCK is noise sensitive (Zaïane, Foss, Lee and Wang 2002). We feel that one reason may be that the similarity between an outlier cluster and a non-outlier cluster or the similarity between a pair of outlier clusters is so small that one non-outlier cluster always has a chance to be merged with another cluster as long as their goodness value is larger than 0, eventually resulting in a very large cluster with low intra-cluster similarity. This is especially true when the data set has various data densities. Therefore, we modified the algorithm by adding an extra criterion, goodness threshold, for cluster merging to overcome a problem noted by Zaïane et al.

3.4 Clustering Validity

Cluster validity involves procedures for evaluating the results of a clustering algorithm (Halkidi et al. 2001). As we mentioned in section 2, a good clustering algorithm does not necessarily generate an optimal clustering result, since it often depends on input parameters. For example, in our experiments,

three of the input parameters (session similarity threshold, number of clusters provided and goodness threshold) affect the clustering results. However, a user has no way of knowing the distribution of a data set beforehand and therefore can hardly provide the optimal set of parameters. Even with many trials, it may be difficult to find an optimal clustering result because clustering is an unsupervised learning and the data set, such as Web sessions, is often too large and not two-or-three dimensional. The user can hardly identify a desired clustering result by directly observing the result.

Most literature on clustering Web usage data does not cover this cluster validity issue and many studies on cluster validity present measurements deal with numerical data, such as using the centroid of a cluster (Halkidi et al. 2001), making them unsuited to Web session analysis. Besides using some evaluation measurements, such as the number of non-outlier sessions and the relative size of the largest cluster, we propose the following intra cluster similarity and inter cluster dissimilarity measurements for boolean and categorical data to help users having different perspectives achieve their desired clustering results. We need to mention that the evaluation measurements are not clustering criteria but are meant to be applied to given clustering results generated from a clustering algorithm to evaluate how good those results are.

3.4.1 Intra Cluster Similarity

The intra cluster similarity measures how data points (sessions in our case) are similar to each other within the same cluster. In the following definitions, intra cluster similarity is defined as a number between 0 and 1.The larger this number, the more similar the data points.

Although our measurements are based on the concept of EGP, the concept can be extended to any attribute for a general case. First we define a concept of *large EGP* for a cluster because we will use a large EGP to represent a pattern of the cluster.

$$large\ EGP = \frac{EGP\ \ frequency}{number\ \ of\ \ sessions} \geq EGP\ threshold \qquad (1)$$

Large EGP is same as the large 1 item in association rule. The same concept is used in (Wang, Xu and Liu 1999), and is called *large item* there.

For a cluster C_p, we define two intra cluster similarities as follows.

$$Intra1(C_p) = \frac{\text{number of large EGPs in } C_p}{\text{number of unique EGPs } C_p},\ Intra1(C_p) \in [0,1] \qquad (2)$$

$$Intra2(C_p) = \frac{\sum_{i=1}^{n_{c_p}^{large}} \text{weight of large EGP(i)}}{\sum_{i=1}^{n_{c_p}} \text{weight of EGP(i)}},\ Intra2(C_p) \in [0,1] \qquad (3)$$

Intra2(C_p) uses weights on EGPs while *Intra1(C_p)* doesn't.

Next we define an intra cluster similarity as an average of the two.

$$Intra(C_p) = \frac{Intra1(C_p) + Intra2(C_p)}{2},\ Intra(C_p) \in [0,1] \qquad (4)$$

For a group of clusters, the overall cluster intra similarity is defined as a weighted mean of each individual cluster's intra similarity as follows.

$$Intra(C) = \sum_{p=1}^{|C|} \frac{n_{c_p}}{n_c} Intra(C_p), \ Intra(C) \in [0,1] \quad (5)$$

Weight is the size of an individual cluster divided by the size of the all the clusters. The reason we use cluster size as a weight, instead of taking a uniform average is to include the influence of the size of a cluster. For example, given two clusters C_1 and C_2. C_1 has 4 sessions and $Intra(C_1)=1.0$, and C_2 has 96 sessions and $Intra(C_2)= 0.2$. If we take the average of the two, $Intra(C)$ = 0.6 which does not really reflect the intra similarity of the group of clusters. Instead, when we take the size of each cluster as a weight, the new intra cluster similarity is $Intra(C)$ = 0.232 which more truly reflects the intra similarity for the group of clusters.

Next we use an example to illustrate the intra similarity concept. Assume that a cluster C_p contains four sessions as follows. (Here we choose an EGP threshold of 0.5)
S_1={a:1, b:1, c:2}, S_2={a:2, c:1, d:1}, S_3={b:1, c:1, e:1}, S_4={a:1, d:2, f:1}

EGP	**a**	**b**	**c**	**d**	e	f
Frequency	**3**	**2**	**3**	**2**	1	1
Weight	**4**	**2**	**4**	**3**	1	1

Table 2 EGPs in cluster C_p

Based on equation (1) we identify four large EGPs (in bold in table 2), represented as a vector $large(C_p)=(a, b, c, d)$. Then we use equations (2) and (3) to compute intra similarities as below.

$Intra1(C_p) = \frac{4}{6}$

$Intra2(C_p) = \frac{4+2+4+3}{4+2+4+3+1+1} = \frac{13}{15}$

$Intra(C_p) = \frac{\frac{4}{6}+\frac{13}{15}}{2} = \frac{23}{30}$

3.4.2 Inter Cluster Dissimilarity

The inter cluster dissimilarity measures how much clusters are dissimilar. In our following definitions, the inter cluster dissimilarity is a number between 0 and 1. The larger this number, the more dissimilar different clusters are.

First we compute inter cluster similarity, and let 1 minus this similarity be inter cluster dissimilarity. In the following, we define two inter cluster dissimilarities for clusters C_p and C_q.

$$Inter1(C_p, C_q) = 1 - \frac{large(C_p) \cap large(C_q)}{large(C_p) \cup large(C_q)}, \ Inter1(C_p, C_q) \in [0,1] \quad (6)$$

$$Inter2(C_p, C_q) = 1 - \frac{W(C_p).large(C_p) \cap W(C_q).large(C_q)}{W(C_p).large(C_p) \cup W(C_q).large(C_q)}, \ Inter2(C_p, C_q) \in [0,1] \quad (7)$$

Similarly, $Inter2(C_p, C_q)$ uses weights on EGPs while $Inter1(C_p, C_q)$ doesn't. Next we take the average of the two as the inter cluster dissimilarity for the pair of clusters.

$$Inter(C_p, C_q) = \frac{Inter1(C_p, C_q) + Inter2(C_p, C_q)}{2},\ Inter(C_p, C_q) \in [0,1] \quad (8)$$

Finally, we include all clusters and compute a mean value as the inter cluster dissimilarity for the group of clusters.

$$Inter(C) = \frac{2}{|C|(|C|-1)} \sum_{i=1}^{|C|} \sum_{j=i+1}^{|C|} Inter(C_i, C_j),\ Inter(C) \in [0,1] \quad (9)$$

Still using the example in 3.4.1, we add another cluster C_q containing the following four sessions. S_5={c:1, f: 1, g: 1}, S_6={f:2, c:1, m:1}, S_7={g:1, h:1, m:1}, S_8={n:1, e:1, f:1}. The large EGP vector, *Large(C_q), is {c, f, g, m}*. Large EGPs are highlighted in table 3.

EGP	**c**	e	**f**	**g**	h	**m**	n
Frequency	**2**	1	**3**	**2**	1	**2**	1
Weight	**2**	1	**4**	**2**	1	**2**	1

Table 3 EGPs in cluster C_q

According to equations (6), (7) and (8), we obtain the following values.

Inter1(C_p, C_q) = $1 - \frac{1}{7} = \frac{6}{7}$

Inter2(C_p, C_q) = $1 - \frac{2}{4+2+4+3+4+2+2} = \frac{19}{21}$

Inter(C_p, C_q) = $\frac{\frac{6}{7} + \frac{19}{21}}{2} = \frac{37}{42}$

4. Experiments

In our experiments, we apply ROCK (Guha et al. 2000) and our proposed E-ROCK algorithms on clustering Web sessions and compare their clustering results. For both of the algorithms, we apply counts of EGPs in session similarity calculation with Jaccard coefficient. The difference of the two algorithms is that E-ROCK takes a goodness threshold larger than zero, while ROCK always uses a goodness threshold of zero.

4.1 Data

To evaluate the algorithms we used half month of Web log data from a customer service Web site in HP. The data set contains 13,631 distinct visited document URLs and 596 unique EGPs. Unlike a traditional Web log that records users' Web browsing behaviors in one file, HP's Web log consists of three XML documents that record session information, accessed documents and search requests, respectively. The session log file contains session information such as start time, duration, and accessed content page URLs. The accessed document log file records information related to visited content pages, such as content page URLs and visit time. Neither the session nor the document-access file tracks any index pages visited. The search request log file includes search queries issued during a session.

4.2 Data Preparation

In theory, content pages recorded in the accessed document file should match those recorded in the session file for the same session. However, we found that 12.9% of the content pages recorded in the document access file never appeared in the session file. This indicates loss of data during the session generation process. To solve this problem we applied a simple heuristic rule: We connected an "extra" content page in the document-access file with a session in the session file when (1) both shared the same IP, and (2) the visit time of the content page occurred between the session start-time and the user's next session start-time. (Users were identified by IPs) Finally, we updated the session's duration if the original session duration did not cover the visit time of the newly added document.

We then extracted desired data fields, such as visited page URLs, from the updated session log file. We filtered out "pure" sessions containing only one EGP. This procedure was similar to the practice of using a minimum page threshold in (Fu et al. 2000). After cleaning, the data contained 9,122 sessions.

4.3 Parameter Settings and Measurement Metrics

We defined a cluster to be an outlier if the size of the cluster was less than a certain threshold. Because of the larger size of our data set, we set the outlier threshold in our experiments at 20 instead of 3 as used by Foss et al. (2001). When computing cluster intra similarity and inter dissimilarity with our measurements, we ignored outlier clusters in order to obtain a meaningful cluster quality measurement (Foss et al. 2001).

To evaluate the performance of different algorithms, we use the following metrics:

a. *Intra cluster similarity* defined in section 3.4.1

b. *Inter cluster dissimilarity* defined in section 3.4.2

c. *Number of non-outlier sessions*

Non-outlier sessions were sessions in non-outlier clusters. This metric reflects how many sessions from the original data set were clustered after we ignored the outlier clusters.

d. *Relative size of the largest cluster*

$$\text{Relative size of the largest cluster} = \frac{\text{number of sessions in the largest cluster}}{\text{number of non-outlier sessions in all clusters}}$$

This metric reflects how large the largest cluster was, relative to all non-outlier sessions in the clustering result.

4.4 Experiments

4.4.1 Experiment I

Guha et al. (1997) point out that when θ is larger than 0.7, ROCK generally results in good clustering. So we set the similarity threshold, θ, at 0.7 and changed the number of clusters provided. Having goodness threshold *g* at 0.2 for E-ROCK, large EGP threshold of 0.15 for both algorithms, we obtained the results from the two algorithms shown in table 4. (We see similar patterns when choosing a large EGP threshold at 0.1 or 0.2)

Number of clusters provided	Intra(C)		Inter(C)		Number of non-outlier sessions		Relative size of the largest cluster	
	ROCK	E-ROCK	ROCK	E-ROCK	ROCK	E-ROCK	ROCK	E-ROCK
<= CNT	0.479	0.538	0.870	0.878	4,031	3,573	78.4%	69.4%
CNT + 1	0.499	0.539	0.878	0.878	4,031	3,570	77.8%	69.4%

CNT +10	0.512	0.569	0.874	0.875	3,999	3,558	76.3%	66.2%
CNT +20	0.521	0.557	0.878	0.871	3,911	3,545	73.6%	65.0%
CNT +30	0.546	0.609	0.885	0.872	3,853	3,486	70.6%	56.6%
CNT +50	0.554	0.610	0.885	0.872	3,785	3,450	69.1%	56.4%
CNT +100	0.539	0.611	0.878	0.862	3,560	3,395	69.3%	56.5%

Table 4 ROCK (CNT=3,918) vs. E-ROCK (CNT=4,011), when provided number of clusters changes

First, we observe that for both algorithms, the clustering results are the same when the number of clusters provided is under a certain value. We call this value a *cluster number threshold (CNT)*. This situation is caused by an additional stop condition in both algorithms that terminates the clustering process when there is no cluster having a candidate cluster to merge with. The CNT is data- and algorithm-dependent. For our data set, when θ = 0.7, it is 3,918 for the ROCK and 4,011 for E-ROCK. In experiment I, we focused on the two algorithms' performance when the number of clusters provided was above CNT.

Since the two algorithms have different CNTs, we feel that to compare their performances under the same number of clusters provided is not an optimal evaluation method. Instead, we believe that it is better to use difference between 'the number of clusters provided' and 'CNT' as measurement units, e.g., we compared the ROCK's performance at its CNT + n as the number of clusters provided with the E-ROCK's performance at its CNT + n as the number of clusters provided.

4.4.2 Experiment II

In this experiment, we studied a situation in which the number of clusters provided is smaller than CNT. We set the number of clusters provided to be 1000, a number less than CNT, and changed similarity threshold, θ. We choose the large EGP threshold at 0.15 for both algorithms and a goodness threshold at 0.2 for E-ROCK. We show the results from the two algorithms in table 5.

Similarity threshold (θ)	Intra(C)		Inter(C)		Number of non-outlier sessions		Relative size of the largest cluster	
	ROCK	E-ROCK	ROCK	E-ROCK	ROCK	E-ROCK	ROCK	E-ROCK
0.4	0.189	0.328	1.000	0.966	7,839	5,763	100.0%	81.6%
0.5	0.206	0.321	1.000	0.976	7,216	5,714	99.6%	86.9%
0.6	0.276	0.362	0.986	0.961	5,995	4,800	98.4%	86.0%
0.7	0.479	0.538	0.870	0.878	4,031	3,573	78.4%	69.4%
0.8	0.636	0.700	0.821	0.827	2,631	2,604	59.8%	51.0%

Table 5 ROCK vs. E-ROCK when similarity threshold changes

5. Result analysis and patterns

5.1 Result analysis

From tables 4 and 5, we obtained the following results.

R1: The E-ROCK algorithm improved intra cluster similarity.

From table 4, for seven different provided numbers of clusters, the intra cluster similarities with E-ROCK were 6.9% to 13.4% higher than those from the original ROCK algorithm. From table 5, compared with results from the ROCK algorithm, the relative increases for intra cluster similarity from

E-ROCK ranged from 10.1% to 73.5%. The reason for the improvement is attributed to a goodness threshold that restrains clusters from merging when their goodness value is lower than the threshold.

R2: Inter cluster dissimilarity values are very close for both algorithms
From tables 4 and 5, the inter cluster dissimilarity values for both algorithms are very close, the maximum difference is only 1.9%, and all those dissimilarity values are above 0.86.

R3: With the E-ROCK algorithm, the number of non-outlier sessions decreased.
From tables 4 and 5, the ROCK algorithm covered more non-outlier sessions than the E-ROCK algorithm. In table 4, the differences in numbers of non-outlier sessions for the two algorithms for seven different numbers of clusters provided were between 4.9% and 12.9%. From table 5, when the similarity threshold, θ, was changed from 0.4 to 0.8, the numbers of non-outlier sessions produced from the ROCK algorithm were 36.0%, 26.3%, 24.9%, 12.8% and 1.0%, respectively, higher than those from the E-ROCK algorithm.

R4: TheE-ROCK algorithm alleviated one large cluster problem.
From table 4, with the E-ROCK algorithm, the decreases in relative sizes for the largest cluster ranged from 10.8% to 19.8%. From table 5, compared with results from the ROCK algorithm, the drop rates were from 11.5% to 17.7%.

R5: Once the similarity threshold, θ, reached 0.7, the intra cluster similarity started to increase dramatically, and the relative size of the largest cluster started to decrease sharply.
From table 5 for the ROCK algorithm, when θ was changed from 0.4 to 0.8 with a step size of 0.1, the increases of intra similarity were 0.017, 0.070, 0.203, and 0.157 respectively. The relative sizes of the largest cluster dropped 0.4%, 1.2%, 20% and 18.6% respectively. The highest drop rates occurred when θ was changed from 0.6 to 0.7. Similar trends appeared in the results for the E-ROCK algorithm. As high intra cluster similarity is often an important measure for good clustering, our above findings support Guha et al.'s (1997) argument that when θ is larger than 0.7, ROCK often generates good clustering.

5.2 Patterns Discovered

We used a large EGP to represent a pattern of a cluster. The following table lists patterns discovered from the five largest clusters using the E-ROCK algorithm and suggests more meaningful interpretations for those patterns. The input parameters to generate data in this table were: similarity threshold θ=0.7, number of clusters provided = CNT+30 (4,041), goodness threshold = 0.2 and large EGP threshold = 0.15.

Cluster index	Size of cluster	Patterns	Interpretation of patterns
1	1,974	/Support/RCEN/A /Support/KNO/ /Support/PAT/PHSS /Support/ENOT/OV-EN /Support/RCEN/	/Support/Response center engineering note/A /Support/Known problem/ /Support/Product patch/Patch subsystems /Support/Engineering note/Openview engineering /Support/Response center engineering note/
4	318	/Support/PAT/SDSK /Support/ENOT/OV-EN /Support/KNO/ITSM	/Support/Product patch/Service desk /Support/Engineering note/Openview engineering /Support/Known problem/IT service manager
11	178	/Support/ENOT/OV-EN	/Support/Engineering note/Openview engineering

		/Support/KNO/	/Support/Known problem/
5	107	/Support/ENOT/OV-EN /Support/RCEN/A /Support/KNO/ /Support/MAN/J	/Support/Engineering note/Openview engineering /Support/Response center engineering note/A /Support/Known problem/ /Support/Manual/J
3	65	/Support/PAT/NNM /Support/PAT/ /Support/KNO/	/Support/Product patch/Network node manager /Support/Product patch/ /Support/Known problem/

Table 6 Patterns discovered from clusters

6. Conclusions

Our objective in clustering Web sessions is to identify groups of similar sessions and find user visit patterns through clusters. We extend the general page concept (Fu et al. 2000) by including partial document names and dynamic pages and use the extended general page (EGP) to represent many individual page URLs that share the same EGP. As a result, high dimensionality in page URLs is dramatically reduced. In our data set, the number of unique URLs user accessed was 13,631 while the distinct EGPs totaled 596. We extended the ROCK *(Guha et al. 2000)* clustering algorithm by adding the EGP count into the session similarity function, and including a goodness threshold to determine whether a cluster will become a candidate to be merged with another. We have tested the extended ROCK (E-ROCK) algorithms on 9,122 Web sessions. Finally, we propose a set of measurements for boolean and categorical data to assess clustering results, and compare Web session clustering results between the ROCK and the E-ROCK algorithms. We have found that the E-ROCK generated clusters with higher intra cluster similarity and alleviated the large cluster problem.

7. Future work

As the time complexity for ROCK is $O(n^2 + nm_m m_a + n^2 \log n)$, ($m_m$ is the maximum number of neighbors for a data point and m_a is average number of neighbors for a data point) when handling a large data set, in order to reduce the clustering process time, we can first draw a random sample set to generate clusters, and then assign remaining data points to the appropriate clusters as described in (Guha et al. 1997) and (Guha et al. 2000). The current clustering on Web sessions does not distinguish users. If needed, we can use IP addresses in sessions to identify users. Based on clustering results, we can build a Web recommendation system and examine how the system can help users find their desired contents in the Web site with fewer clicks.

References

Avedal, K. et al. *Professional JSP*, Wrox Press, 2000, p116

Banerjee, A. and Ghosh, J. "Clickstream Clustering using Weighted Longest Common Subsequences", *Proceedings of SIAM Conference on Data Mining*, Chicago, USA, April 2001

Estivill-Castro, V. and Yang, J. "Categorizing Visitors Dynamically by Fast and Robust Clustering of Access Logs", *Lecture Notes in Computer Science*, 2001

Fu, Y., Sandhu, K. and Shih, M.Y. "A Generalization-based Approach to Clustering of Web Usage Sessions", *Lecture Notes in Artificial Intelligence*, Vol. 1836, Springer, 2000, pp. 21-38

Halkidi, M., Vazirgiannis, M. and Batistakis, Y. "Quality Scheme Assessment in the Clustering Process", *Proceedings of PKDD 2000*, Lyon, France, 2000

Halkidi, M., Batistakis, Y., Vazirgiannis, M. "On Clustering Validation Techniques", *Journal of Intellegent Information Systems*, 2001

Heer, J. and Chi, E. "Separating the Swarm: Categorization Methods for User Sessions on the Web",

Proceedings of CHI 2002, Minneapolis, Minnesota, USA, April 2002
Foss, A., Wang, W. and Zaïane, O.R. "A Non-Parametric Approach to Web Log Analysis", *Proceedings of Workshop on Web Mining in First International SIAM Conference on Data Mining*, Chicago, IL, April 2001, pp. 41-50
Guha, S., Rastogi, R., and Shim, K. "A Clustering Algorithm for Categorical Attributes", *Technical report*, Bell Laboratories, Murray Hill, 1997
Guha, S., Rastogi, R., and Shim, K. "CURE: An Efficient Clustering Algorithm for Large Databases", *Proceedings of ACM SIGMOD Conference*, 1998
Guha, S., Rastogi, R. and Shim, K. "ROCK: A robust clustering algorithm for categorical attributes", *Information Systems*, 2000
Joshi, A., and Krishnapuram, R. "On Mining Web Access Logs", *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2000
MacQueen, J. "Some methods for Classification and Analysis of Multivariate Observations", *Proceedings of 5th Berkeley Symposium on Mathematics, Statistics and Probability*, Vol I: Statistics 1967, pp. 281 - 298
Menasalvas, E., Millán, S., Pérez, M., Hochsztain, E., Robles, V. and Marbán, O., Peña, J., Tasistro, A. "Beyond user clicks: an algorithm and an agent-based architecture to discover user behavior", *Proceedings of ECML/PKDD-2003*, Cavtat-Dubrovnik, Croatia, September 2003
Shahabi, C., Zarkesh, A., Adibi, J. and Shah, V. "Knowledge Discovery from Users Web-Page Navigation", *Proceedings of the 7th International Workshop on Research Issues in Data Engineering*, 1997
Srivastava, J., Cooley, R., Deshpande, M., and Tan, P.N. "Web Usage Mining Discovery and Applications of Usage Patterns from Web Data", *ACM SIGKDD Explorations*, January 2000
Wang K., Xu, C., and Liu, B. "Clustering transactions using large items", *Proceedings of CIKM99*, Kansas City, MO, USA, 1999, pp. 483-490
Wang, W. and Zaïane, O.R. "Clustering Web Sessions by Sequence Alignment", *Proceedings of DEXA'02*, Aix-en-Provence, France, September 2002
Zaïane, O.R. and Foss, A., Lee, C.H. and Wang, W. "On Data Clustering Analysis: Scalability, Constraints, and Validation", *Proceedings of PAKDD 2002*, April 2002, pp. 28-39
Zhang, T., Ramakrishnan, R. and Livny, M. "BIRCH: An Efficient Data Clustering Method for Very Large", *Proceedings of ACM SIGMOD International Conference Management on Data*, Montreal, Canada, June 1996
(21) http://depts.washington.edu/trio/comp/howto/site/organize/organizefiles.shtml