Clustering of Web Users Based on Access Patterns

Yongjian Fu Kanwalpreet Sandhu Ming-Yi Shih
Computer Science Department
University of Missouri-Rolla
{yongjian,ksandhu,mingyi}@umr.edu

Abstract

The clustering of the Web users based on their access patterns is studied. Access patterns of the Web users are extracted from Web servers' log files, and then organized into sessions which represent episodes of interaction between Web users and the Web server. Using attributed-oriented induction, the sessions are then generalized according to the page hierarchy which organizes pages according to their generalities. The generalized sessions are finally clustered using a hierarchical clustering method. Our experiments on a large real data set show that the method is efficient and practical for Web mining applications.

1 Introduction

With the rapid development of the World Wide Web (WWW), or the Web, many organizations now put their information on the Web and provide Web-based services such as on-line shopping, user feedback, technical support, etc. Web mining, the knowledge discovery in the Web, has become an important research area [2]. Research in Web Mining can be broadly classified into Web content mining and Web usage mining.

An important topic in Web usage mining is the clustering of the Web users, i.e., grouping the users into clusters based on their common properties. By analyzing the characteristics of the groups, webmasters may understand the users better and may provide more suitable, customized services to the users.

In this paper, the clustering of the Web users based on their browsing activities or access patterns on the Web is studied. Users with similar browsing activities are clustered or grouped into classes (clusters). For example, if a number of customers spend quite a lot time on browsing pages about "baby furniture", "baby toys", and "diapers", they may be clustered into a group which could later be analyzed by webmasters or domain experts as "expecting parents". The webmaster then may, for example, arrange the Web pages so that the above pages are interlinked together. Also, when a user has browsed the "baby furniture" and the "baby toys" pages, a link to "diapers" page can be dynamically created and inserted in the current page.

In our approach, the server log data is first processed to identify sessions of Web usages. A session is basically a unit of interaction between the user and the Web server. The sessions are then generalized using the attribute-oriented induction method [4] which will greatly reduce the dimensionality of data. The generalized data are finally clustered using an efficient hierarchical clustering algorithm, BIRCH [10]. The approach is tested on a real large data set. Our experiments show that the method is efficient and we have found several interesting clusters within the data set.

2 Background

Earlier studies on Web usages, such as access statistics, lack the in-depth understanding of user browsing patterns, including pages traversed and time spent on each page. These user browsing

patterns provide accurate, active, and objective information about the Web usages of the users. Moreover, most Web servers, e.g., NCSA's HTTPD [3] and Microsoft's IIS [5], contain such information in their log of page requests.

A Web server's log will contains records of user accesses. Each record represents a page request from a Web user (called client). A typical record contains the client's IP address, the date and time the request is received, the URL of the page requested, the protocol of the request, the return code of server indicating the status of the request handling, and the size of the page if the request is successful. An example is given below which are excerpted from log of the University of Missouri-Rolla's (UMR) Web server, which runs HTTPD 1.0. The IP addresses are modified for privacy reasons. The URLs of the pages are relative to the UMR's home page address, `http://www.umr.edu`.

```
smith.cs.umr.edu - - [01/Apr/1997:00:03:24 -0600]
"GET /~amigos/Virtual/cosas2.html HTTP/1.0" 200 11746
```

From such a Web server log, user access patterns can be extracted. A user's access pattern consists of the pages she visited and the time she spent on each page. Each user can then be represented by a set of (page-id, time) pairs.

Recently, a clustering algorithm, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [10] is proposed. It usually creates a good clustering in just one scan of the data set, so it scales up well for very large data sets. Moreover, BIRCH is a hierarchical and incremental algorithm which is desirable for our purpose. However, the direct application of BIRCH on the primitive user access data as described above is very inefficient and may not find interesting clusters as explained below.

- A Web server usually contains thousands, even millions of pages. It is obviously impractical to represent each user as a high dimensional vector in which each dimension represents a page. Sparse representation will cause the dimensionality of clusters changes dynamically, i.e, the non-sparse items in the centroid of a cluster may grow when new vectors are added. This imposes a lot of overheads on memory and storage management as well as on data structure handling.
- Our goal is to cluster Web users with similar access patterns. However, it is not easy to find many users who access common pages because of the diversity of Web users. This will lead to either small clusters or clusters of users with little commonness. It is more likely we can find groups who share the common interests in the themes of pages. For example, we may not find a large number of users who visit pages on "Okla Homer Smith Mavado crib - white finish", "Kids II Bright Starts musical mobile", and "12-count cotton Rainbow diapers", but we can probably find lots of users who browse pages on "baby furniture", "baby toys", and "diapers".

Based on the above observations, we propose an generalization-based clustering approach which combines attribute-oriented induction [4] and BIRCH to generate a hierarchical clustering of Web users based on their access patterns.

3 Session Identification

A Web user may visit a Web site from time to time and spend arbitrary amount of time between consecutive visits. To deal with the unpredictable nature of Web browsing and make the problem tractable, a concept, *session*, is introduced as the unit of interaction between a user and a Web server. A session consists of pages accessed by a user in a certain amount of time. Clusters are found in the sessions instead of the users' entire histories. The fact we cluster sessions instead of users can be justified that our goal is to understand the usage of the Web and different sessions of a user may correspond to the visits of the user with different purposes on mind. In addition, multiple users on a shared computer can be represented by different sessions. Concepts similar

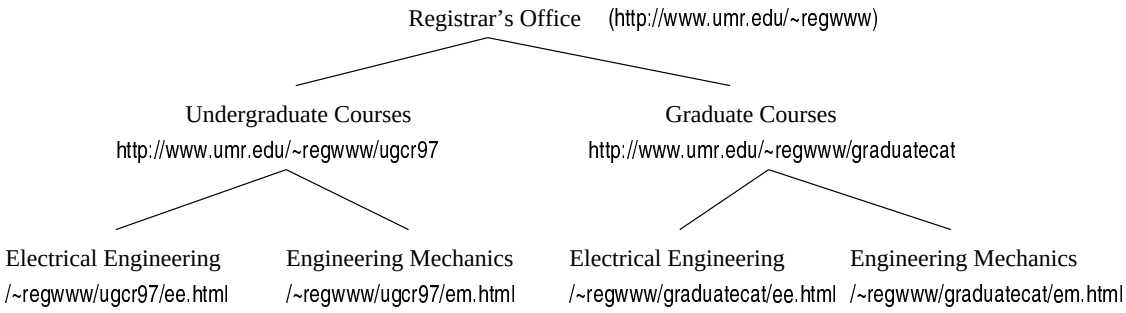

Figure 1: An example of page hierarchy.

to session have also been proposed in other studies [7, 6, 8, 1]. In this paper, we will use session and user interchangeably.

The sessions are identified by grouping consecutive pages requested by the same user together. The data in a Web server log is transformed into a set of sessions in the form of (*session-id,* $\{\langle$*page-id, time*$\rangle\}$), where *session-id* and *page-id* are unique IDs assigned to sessions and pages. A session $(sid_0, p_0, 10, p_1, 30, p_2, 20)$ tells a user spent 10 seconds on page p_0, 30 seconds on page p_1, and 20 seconds on page p_2.

The Web server's log is scanned to identify sessions. A session is created when a new IP address is met in the log. Subsequent requests from the IP address is added to its session as long as the elapse of time between two consecutive requests does not exceed a pre-defined parameter *maximum_idle_time*. Otherwise, the current session is closed and a new session is created.

4 Generalization-based Clustering of Sessions

As mentioned in Section 3, a session is represented as a vector of times spent on pages. Such a representation works well when the number of pages is small. However, when there are a great number of pages, such fine granularity causes degeneration in the quality of clusters as well as the efficiency of the clustering algorithm. Besides, clustering is to find groups with similar access patterns which does not necessary correspond to page level. Groups that are not obvious at page-level may emerge when considered at a higher level.

By analyzing the pages, we realize that pages are not randomly scattered, but they are organized into a hierarchical structure, called *page hierarchy*. A page hierarchy is a partial order of Web pages, in which a leaf node represents a Web page corresponding to a file in the server. A non-leaf node in a page hierarchy represents a Web page corresponding to a directory in the server. A link from a parent node to a child node represents the consisting-of relationship between the corresponding pages. To distinguish the two kinds of pages, a page represented by a leaf node is called a *simple page*; and a page represented by a non-leaf node is called a *general page*. For example, a page hierarchy for some pages in the UMR Web server is shown in Figure 1 which has four simple pages and three general pages.

The page hierarchy can be created automatically based on the URLs of the pages. For example, the page hierarchy in Figure 1 can be constructed from the four simple pages (the mnemonic names of general pages are optional). The root of the hierarchy which is the home page of UMR is not shown in Figure 1 for simplicity.

```
http://www.umr.edu/~regwww/ugcr97/ee.html
http://www.umr.edu/~regwww/ugcr97/em.html
http://www.umr.edu/~regwww/graduatecat/ee.html
http://www.umr.edu/~regwww/graduatecat/em.html
```

The sessions found in Section 3 are generalized using the attribute-oriented induction method [4]. In attribute-oriented induction, the simple page in each session is replaced by its corresponding

general page based on the page hierarchy. Duplicate pages are then removed with their times added together.

The generalization of the sessions involves two steps: page hierarchy construction and attribute-oriented induction of sessions, which are explained as follows.

1. Construction of page hierarchy.

   A page hierarchy is initialized with only the root which represents the home page of the Web server. For each URL in the URL table generated in Section 3, if it does not already exist in the page hierarchy, a node for the page is created. Next, the URL is parsed and for each prefix which is a legal URL, a node for the general page is created if it does not exist in the page hierarchy. For every pair of nodes in which one's URL is a closest prefix of the other, a links is added. For example, for the URL `http://www.umr.edu/~regwww/ugcr97/ee.html`, nodes for itself, its first prefix `http://www.umr.edu/~regwww/ugcr97/`, and its second prefix `http://www.umr.edu/~regwww/` may be created and a link is added between itself and its first prefix, between its first prefix and its second prefix, and between its second prefix and the root.

2. Attribute-oriented induction of sessions.

   For each session found Section 3, its pages are replaced by their corresponding general pages in the page hierarchy, called tree climbing in attribute-oriented induction. The level of the general pages to climb is decided by the user or inferred from a user-given threshold which specifies the maximum number of general pages in results. If two pages in a session have the same higher-level general page, one of them is removed and its time is added to the other's. The session is said to be generalized and a session so obtained is called a generalized session. For example, for a session of simple pages in Figure 1, `((Undergraduate Electrical Engineering Courses, 25), (Undergraduate Engineering Mechanics Courses, 48), (Graduate Electrical Engineering Courses, 32), (Graduate Engineering Mechanics Courses, 19))`, the attribute-oriented induction method will generalize it to a generalized session `((Undergraduate Courses, 73), (Graduate Courses, 51))` at level 2.

Since the number of general page is much less than that of simple pages, the generalization of session greatly reduce the dimensionality. As a result, a generalized session can then be represented by a regular vector, (*session-id*, $t_1, t_2, \cdots, t_n$), where t_i is the total time the user spent on the i-th general page and its descendents. Note the page IDs of these general pages are not included in the vector because all session are on the same set of general pages.

Another advantage of the generalization of sessions using page hierarchy is that the resulting data representation can accommodate updates of Web pages, such as addition and deletion of pages, as long as the higher level structure is stable.

The generalized sessions are clustered using the BIRCH algorithm [10]. BIRCH builds a Clustering Feature (CF) tree as the result of clustering by incrementally inserting objects (represented by vectors) into the CF tree. A CF tree is a multidimensional structure like B+ tree in which a nonleaf node stores entries of (CF_i, *pointer_to_child_i*), and a leaf node stores entries of (CF_i), where CF_i is a CF vector. A CF vector is a triple containing the number, the linear sum, and the square sum, of the vectors in the subtree rooted at a child.

When a new vector is inserted into the CF tree, it goes down from the root to a leaf node by choosing the closest child according to a distance measure, such as the Euclidean distance. If any entry of the leaf node can incorporate the new object within a diameter threshold T, that entry's CF vector is updated. Otherwise, the object is put into an empty entry in the leaf node. In the later case, if there is no empty entry left in the leaf node, it is split into two. In case there is a split, an empty entry in the parent node is used to record the new leaf node. The parent node is split in a similar way if there is no empty entry and if this goes up to the root, the tree is one level deeper. When the tree grows too large to be held in memory, the threshold T is enlarged and the current tree is converted into a new tree by inserting all leaf node entries of the current tree into the new tree, which is guaranteed to be smaller.

5 Experiments

The algorithms have been implemented and tested on a data set collected from the UMR's Web server log (`http://www.umr.edu`). It contains more than 2.5 million records with a total size of 270MB. The experiments are carried out on a Sun SparcStation Ultra 1 with 64MB of memory running Solaris 2.5.

We tested the algorithms on several testing sets which are subsets of the data set, which contain from 50,000 to 500,000 records. The *max_idle_time* is set to 30 minutes. The number of sessions identified in the testing sets is shown in Table 1, which also shows the number of distinct pages and the number of distinct hosts in the testing sets. From the figure, it is clear that the number of sessions is linear to the number of records in the testing sets.

testing set	no_of_records	no_of_distinct_pages	no_of_distinct_hosts	no_of_sessions
50k	50,000	5,731	3,694	18,184
100k	100,000	8,168	6,304	35,665
200k	200,000	12,191	11,473	70,702
300k	300,000	15,201	16,498	107,322
400k	400,000	17,574	21,153	142,457
500k	500,000	21,308	26,107	178,655

Table 1: Number of sessions in testing sets

The sessions are then generalized to level 2 which is the level just below the root. This is chosen because we want to see the effect of attribute-oriented induction on dimensionality reduction. The generalized sessions have a dimensionality of 1,628 except for the 50k testing set which is 1,194. Basically, the generalization has greatly reduced the dimensionally, by two thirds in the 50k testing set and by nine tenths in the 500k testing set.

It has also been found that the total execution time in session identification and session generalization is almost linear to the number of records in the testing sets.

The resulting clusters are analyzed by examining the general pages in the clusters. Most leaf clusters in the hierarchical clustering contain only a few general pages which are considered as the general interest of the group. For example, there is a group who are interested in the Web pages of the registrar's office. Sometimes, the pages in a cluster are too general and we have to look at the simple pages in the data set to evaluate the validity of the cluster. For example, there is a cluster containing the home page of the mechanical engineering department. A detailed analysis reveals a group of users who are interested in mechanical engineering professors' home pages.

It can be summarized that the method we proposed in this paper scales up well for large data sets. Moreover, meaningful clusters may be found in the generalized sessions.

6 Conclusion and Future Work

The clustering of the Web users based on their access patterns has been studied. A generalization-based clustering method is proposed which employs the attributed-oriented induction method in clustering to reduce the large dimensionality of data. Our experiments on a large real data set show that the method is efficient and practical for Web mining applications.

In the generalization of sessions, a level is given to determine the general pages which are going to replace the simple pages. This level plays an important role in forming the generalized sessions as well as the final clustering. If the level is set too high, over-generalization may occur in which too much details are lost and the validity of the clusters may be in question. On the other hand, if the level is set too low, it may not reduce the dimensionality by much. More experiments need to be conducted to gain more insights on the issue.

It is found that even after attribute-oriented induction, the dimensionality of generalized sessions is still very large. A possible way to deal with this is first clustering the Web pages [9] and then organizing them into the page hierarchy.

The construction of the page hierarchy based on the URLs of pages implies that the underlying page organization reflects the semantics of the pages. In case this cannot be assumed, the page hierarchy should be constructed according to the semantics of the pages, e.g., by using a document clustering or categorization method.

Lots of Web sites require their users to register. A natural extension of our method is to combine user's registration information, such as age, income level, address, etc, with their access patterns in clustering.

Acknowledgments

This work is supported by University of Missouri Research Board Grant R-3-42434. We thank Dr. Tian Zhang for the source code of BIRCH and Meg Brady for the testing data set.

References

[1] M. S. Chen, J. S. Park, and P.S. Yu. Efficient data mining for path traversal patterns in distributed systems. *Proc. 1996 Int'l Conf. on Distributed Computing Systems*, 385, May 1996.

[2] O. Etzioni. The world-wide web: Quangmire or gold mine? *Communications of ACM*, 39:65–68, 1996.

[3] National Center for Supercomputing Applications. *NCSA httpd.* http://hoohoo.ncsa.uiuc.edu/docs/Overview.html, 1995.

[4] J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: An attribute-oriented approach. In *Proc. 18th Int. Conf. Very Large Data Bases*, pages 547–559, Vancouver, Canada, August 1992.

[5] Mircosoft Inc. *Internet Information Server.* http://www.microsoft.com/ntserver/web/exec/overview/overview.asp, 1999.

[6] B. Mobasher, N. Jain, S. Han, and J. Srivastava. *Web Mining: Pattern Discovery from World Wide Web Transcations.* Technical Report, University of Minnesota, avialable at ftp://ftp.cs.umn.edu/users/kumar/webmining.ps., 1996.

[7] J. Moore, S. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher. *Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering.* Workshop on Information Technologies and Systems, avialable at ftp://ftp.cs.umn.edu/users/kumar/web-wits.ps., 1997.

[8] C. Shahabi, A. Z. Zarkesh, J. Adibi, and V. Shah. Knowledge discovery from users web-page navigation. In *Proc. of 1997 Int. Workshop on Research Issues on Data Engineering (RIDE'97)*, Birmingham, England, April 1997.

[9] O. Zamir, O. Etzioni, O. Madani, and R. Karp. Fast and intuitive clustering of web documents. In *Proc. 1997 Int'l Conf. on Data Mining and Knowledge Discovery (KDD'97)*, pages 287–290, Newport Beach, CA, August 1997.

[10] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data*, pages 103–114, Montreal, Canada, June 1996.