Chapter 3

GENERAL WEB USAGE MINING SYSTEMS

General Web Usage Mining systems focus on the analysis of log files using data mining techniques for discovering general access patterns and trends of users. Majority of the studies under this category aim to discover user navigation paths from the log files.

In General, navigation path of a visitor is mainly the path followed by him/her through out his/her visit to the web site. It should be noted that each different study enlarges this definition by determining some specifications on what can be accepted as a path. Systems also differentiate on how they use the paths found. Some of the general usage mining systems present the user navigation paths without any further processing. Some others provide a query language for analyzing the paths better.

In addition, there are studies on clustering paths or just user sessions with the aim of finding similar interest groups among visitors. The other types of studies under this category proposes to adapt well known data mining techniques such as association rule mining to the problem of web usage mining. In this section, we will explain each of these studies in detail.

3.1. Mining Traversal Paths

One of the existing approaches for mining navigation patterns from log files is to make use of well-known techniques from data mining. An example of such a study aims to find frequently occurring paths, which are named as maximal
reference sequences from the log file by using a methodology similar to the association rule mining, M.-S. Chen et al. [19].

The first step of the proposed solution procedure is to traverse the whole log file for finding maximal forward references for each user. To be able to do that, the log file is divided into user paths where each path contains the accesses belonging to a specific user. Then, each user path is processed to find maximal forward references contained it.

A maximal forward reference is defined as a sequence of pages that are visited consecutively by the visitor in which each page is seen only once. Whenever a backward reference to a page previously visited is seen, the current maximum forward reference path is terminated, added into the database and a new one starts. While traveling through the pages, visitors generally turn back to the previously visited pages and choose other links from them. The pages seen on the way back to the previous pages are visited only because of their location, but not their content. In the light of this observation, the study concentrates only on forward references.

As an example, assume that the traversal sequence of the Visitor A is as follows: (P1, P2, P3, P4, P3, P2, P7). In this example, Visitor A turns back to the page P3 and P2 consecutively after retrieving page P4. Here, the page P3 is retrieved only for being able to retrieve page P7 from page P2. The algorithm forming maximal forward references solves that problem by removing the backward references from the paths. The algorithm produces the following maximum forward references for Visitor A: (P1 P2 P3 P4, P1 P2 P7) Once the maximal forward references for each user are formed, the next step of the solution for mining path traversal patterns is ready for the execution. In this step, the database containing maximal forward references for all users is processed to be able
to form large reference sequences, which are frequently occurring consecutive subsequences among all maximal forward references.

Full Scan (FS) and selective scan (SC) are two different algorithms for finding the large reference sequences. FS algorithm is indicated to be similar to the well-known algorithm called Direct Hashing with Pruning (DHP) for mining association rules with adaptations to the current problem. Because, the problem of finding large reference sequences from the database of maximal forward references has common points with finding large item sets from the database of transactions in association rule mining. The main difference between these two problems is that the order of the items in an item set is not important in association rule mining while it is crucial in mining 10 traversal patterns. So, Full Scan algorithm changes the joining strategy used in the candidate generation phase of the DHP algorithm. Selective Scan algorithm is similar to Full Scan algorithm with optimizations to reduce I/O cost. Maximal reference sequences are the subset of large reference sequences so that no maximal reference sequence is contained in the other one.

For example, if the large reference sequences are AB, AE, AGH, and ABD then maximal reference sequences become AE, AGH, and ABD. The sequences obtained through this way can then be used by web masters in redesigning the links between the pages that are accessed together in making marketing decisions.

3.2. Mining Navigation Patterns with Hypertext Probabilistic Grammars

Another study towards the problem of mining access patterns of visitors proposes to model user navigation sessions as a hypertext probabilistic grammar (HPG), Borges and Levene [12],Li[71].
A user session is defined as a sequence of page requests coming from the same machine where the time passing between each request is less than a certain time limit. After the HPG is formed, the paths followed frequently by the visitors are discovered by applying a special case of depth first search algorithm on it. Each terminal and no terminal symbol of HPG built from user sessions correspond to a web page and there is a one-to-one correspondence between terminal and no terminal symbols. The links between web pages are represented by the production rules of the grammar. Two additional states, S and F are added into the grammar to represent the start and finish of the paths.

In the corresponding automata, states represent no terminal symbols where transitions between states are formed by productions. Each production originating from a state is attached with a value, which is the probability that the link corresponding to a production was chosen from the links on a page represented by that state. In case of a start state, the probabilities of the productions are derived from the rate of the number of times that the page is visited to the overall number of hits.

When navigating through a web site, visitors may concentrate on unrelated topics in a single session. Accordingly, the concept of N-Grammar is suitable to be employed in building HPG. N-Grammar dictates that the link that will be chosen by a visitor on any page is effected only by the last N pages retrieved by him/her.

In HPG that makes use of the concept of N-Grammar, the number of states may increase too much if N is chosen to be very large. This is because each distinct consecutive sequence of N pages visited by any user should have a corresponding state in HPG. After the construction process, user preferred paths are discovered from HPG by applying depth first search like algorithm.
Before mining, the mining expert should specify support and confidence thresholds, which will affect the quality of the paths discovered. Support threshold ensures that the path is frequently visited while confidence threshold ensures that the derivation probability of the corresponding string is high enough. The support value for a particular path is obtained by looking at the probability of the derivation of the first state of this path from the start state. In addition, the confidence value is obtained from the derivation probabilities of other the pages on the path.

By the help of support and confidence thresholds, it becomes possible to discover the paths that describe the common visitor behavior best.

3.3. Analysis of Web Logs through OLAP Mining

A totally different approach to web usage mining is to make use of OLAP (Online Analytical Processing) technology on mining process.

WebLogMiner, O. Zaiane et al. [23], is one of the tools, which aim to incorporate the OLAP technology and the data mining techniques.

OLAP techniques are being used to obtain a portion of the data that is interesting to the analyst who can also determine the abstraction level on which the data will be presented. Then, this data can be used as an input to the data mining algorithms. Results obtained through mining the data can also be presented in different ways by using OLAP techniques. So, the mining process becomes more interactive and flexible.

OLAP technology firstly places the data into a data cube, which is stored in multidimensional array structures or relational databases. Each dimension of the
data cube represents a distinct field of the data, such as URL or domain name. If the data cube has n dimensions, each cell is characterized by having distinct values for the fields represented by these dimensions. Each cell in the data cube stores the number of visitors, which have the same values with the values characterizing the cell.

The advantage of the data cube representation is to make it possible to view from different perspectives and abstraction levels, which are performed by the OLAP operations such as, drill down, roll up and slice.

The following statements are examples for the simple queries that OLAP can answer quickly:

- Hits coming from Turkey, between March 2001 and May 2001
- Hits coming from edu domain on 23/03/2001 with agent Mozilla

In addition to the analysis performed by OLAP technology, WebLogMiner makes use of data mining techniques to analyze the data to answer questions that OLAP cannot. For this, it applies well-known data mining techniques such as association rule mining, clustering or time series analysis on the data stored in the data cube.

For example, by performing time series analysis, it answers the following questions:

- what are the typical page request sequences performed by the visitors? Namely, are there request sequences that are common to most of the visitors?
- What are the event trees belonging to specific time intervals? Here, event trees contain the traversal patterns of the visitors in an aggregated form.
- How the traffic on a web site changes depending on time?
- Are there particular trends on particular times of a day, month etc.? 

26
3.4. Web Utilization Miner

Web Utilization Miner (WUM) is another data mining tool designed for mining user navigation patterns from web logs. The distinguishing facility of this tool is to provide a mining language by which users can dynamically specify constraints on the mining result. WUM is composed of two major modules: Aggregation service and the query processor, Spiliopoulou [18].

Aggregation service firstly processes the log file and divides it into the visits that are used in constructing the aggregate tree on which the mining will be performed. Adding each path seen on the log file extends the tree. While forming the tree, paths that have a common prefix are merged. So, all paths are represented in the tree at the end. Each node taken together tree contains a URL, occurrence count and the number of visitors reaching that node by following the path starting from the root node. Because there exists visits that contain the same URL more than once, each node is associated with an occurrence count to show which occurrence of the URL this is. The way of storing paths in an aggregate tree was chosen for reducing the space requirements and speed up the mining process. The aggregate tree is built for once and used as input for the other module. Query processor is the module that performs the interactive mining on the aggregate tree constructed by the Aggregation service. The user can specify structural, textual and Statistical constraints on the mining result.

For example, the following query, which is expressed in MINT syntax, will result in a graph showing navigation patterns between B.HTML and any page whose support is larger than 1. The wildcard between the nodes Means that there may be any number of nodes between X and Y. But the order of the template variables should stay the same as given in the query.
Select T nodes as X Y, template X*Y as T and X.name= B.HTML and Y.support!

The algorithm that is used for mining according to the given template and other constraints firstly finds all possible bindings for all template variables.

At first, all possible bindings for the first template variable are found by checking all nodes taken together tree. This is the first and the last time that the query processor processes the whole tree. After that, only the trees rooted at the nodes that contain the URLs bound to the first template variable are processed for finding the possible bindings for the second template variable. Each different binding obtained for the template variables is named as pattern descriptor. That is, pattern descriptors contain identifiers that match to the template variables in given queries and wildcards. Namely, in this step of the algorithm, the structural and textual constraints specified by the user are taken into consideration.

Next, the algorithm obtains Navigation pattern corresponding to each pattern descriptor found in the first step. The designers of the system define a navigation pattern as a graph formed according to the pattern descriptor. For each pattern descriptor, the algorithm firstly finds all branches of the Aggregate Tree that contains the pattern represented in that descriptor.

Then, these branches are merged at their common prefixes and on the identifiers existing in the pattern descriptor. While merging the branches, the counts of the nodes that are merged together are added. After merging, the statistical constraints on the mining result are checked and a descriptor is ignored if these constraints are not satisfied. At the end, remaining navigation Patterns are shown to the user.
3.5. WebSift

WebSift is a web usage mining system, which aims to apply well-known data mining techniques on the usage data obtained through web logs, Cooley et al. [32].

It divides the mining process into three main phases: Preprocessing, Pattern Discovery and Pattern Analysis.

The aim of the Preprocessing step is to turn the raw data in the log file into a form that is suitable for mining.

Then, in the second phase of the usage mining process, well-known data mining techniques such as association rule mining, sequential pattern Mining or clustering is applied on the transactions obtained in the previous phase.

In the third phase, creators of the Web Sift system propose to provide a query language and visualization facilities. In addition, in this phase of the mining process, using the Information Filter filters uninteresting results.

Preprocessing step includes cleaning the data, identifying users and sessions belonging to them, completing the missing references in paths and formatting the data to obtain the appropriate transaction type for the type of the mining operation that will be performed.

Data cleaning is the removal of irrelevant and redundant data in the log file such as requests for graphics. Besides, user identification tries to identify the requests belonging to each user. Authors indicate that an IP address may not be suitable to differentiate between the users, because two visitors may be using the same IP at the same time.
For the solution of this problem, they propose to make use of some heuristics. For example, if two requests come from different types of browsers from the same machine, these requests are accepted to be performed different users. After the users are determined, the accesses belonging to them are divided into sessions. Then, the pages that are not recorded but accessed by the visitor are determined and added to the sessions. In the Information Filter, the interestingness level of the rules is determined by looking at the site structure.

Currently, the system is capable of determining the interestingness level of frequent item sets and association rules with two different techniques: BME (Belief Mined Evidence) and BCE (Beliefs with conflicting Evidence).

BME finds the frequent item sets, which contain pages that are not directly linked. Frequent item sets that contain linked pages are not that interesting because it is already guessed by the site designer who put a link between them. On the other hand, if many visitors retrieve pages that have no link in between together, this may be an interesting result for the site designer who may notice a deficiency in the site design. The pages that are linked, but not in the same frequent item set may also be interesting.

BCE finds that kind of pages. The result shows that the link between these pages is rarely used by the visitors, which may give a clue to the site designer for the removal of this redundant link.

3.6. WAP Mine

Wap(Web Access Pattern) Mine is an efficient data-mining algorithm for discovering web access patterns from the Wap Tree (Web Access Pattern Tree), which is a compact data structure, designed for storing the data obtained from the logs, Perkowitz and Etzioni [17], Das[62].
The result of the algorithm is the set of frequent access patterns, which contain pages requested sequentially by enough number of visitors. Wap Tree is formed by the addition of the frequent access subsequences that take part in the log in hand.

Before the construction of the Wap tree, the log is traversed once for finding frequent 1-sequences, URLs that are seen in efficient number of user sessions. Then, the URLs that are not frequent are filtered from the sessions resulting in frequent access subsequences. Wap Tree is constructed by merging the frequent access subsequences on their common prefixes.

Another feature of the Wap Tree is that all nodes that contain the same URL are linked into a queue and another data structure, header table, contains a pointer to the head of all queues. The foundation of the Wap Mine algorithm is based on a heuristic called Suffix Heuristic.

Suffix Heuristic says that if an event (page reference) e is frequent in the prefixes of sequences that have a suffix, which contains pattern P as a subsequence, and then eP is a pattern. The algorithm for finding frequent sequences based on that heuristic is named as conditional search, which is employed in Wap Mine for mining sequences.

Wap Mine algorithm processes each event one by one. For each event (page reference) e i, it firstly forms the conditional tree for that event. Conditional Wap tree for e i contain the set of prefixes of the subsequences that contain e i as a suffix. After this, the algorithm continues to mine Conditional Wap Tree recursively. Finally, the results obtained from mining conditional Wap Tree are concatenated with e i. Page sequences obtained through this algorithm correspond to the frequent access patterns.
3.7. Clustering User Sessions

Another study towards Web Usage Mining proposes to cluster visitors of a website based on the page requests taking place on the sessions belonging to them.

The aim of this study presented in Fu et al. [42] is to discover the groups of pages that are visited together by many visitors. This information can then be used by the Web master in redesigning the Web Site or updating it with extra links between these pages.

In this study, a log file of the web site is initially divided into user sessions. For clustering, each user session should be represented with a vector of pages in which each entry corresponds to the time spent on that page. However, in a web site that has many pages, the size of the vectors will increase dramatically. Because, the vectors should have an entry for each page in the web site.

To overcome this, session vectors are generalized by using Attribute Oriented Induction method. Entries in the session vector are generalized by looking at the page hierarchy of the web site.

A page hierarchy can be derived from the directory structure of the server and represented in the form of tree. The leaves of the tree correspond to the URLs. The parents of the leaf nodes keep the names of the web pages corresponding to innermost directories containing the URLs represented by their children. The parents of the no leaf nodes are the web pages representing outer directories containing them. The directories are derived directly from the names of the URLs. For example, the parent of the URL http://www.umr.edu/regwww/ugcrc97/ee.html is http://www.umr.edu/regwww/ugcrc97.
By using the page hierarchy obtained by this way, the pages in the sessions are generalized as much as specified by the mining expert by using the tree climbing method from attribute oriented induction. During generalization, each page is replaced with one of the parents depending on the level to which pages are generalized. After generalization of the pages in the session vector, the duplicate general pages are merged into one by adding the times spent on them. By this way, the size of the session vectors is decreased dramatically, because the number of general pages is smaller than the number of URLs. Sessions obtained by this way are then clustered by using BIRCH hierarchical clustering algorithm.

In BIRCH, using a tree structure, which it calls as CF tree, performs clustering. In CF tree, leaf nodes contain the current clusters, which are the collection of session vectors while the no leaf node store CF vectors, which characterize the clusters below them. When a new session vector should be placed into the tree, it goes until a leaf node by choosing the branches that are the closest to him/her. The CF vectors of the parent nodes are updated accordingly. In case there is no matching entry in leaf with given thresholds, the session vector is put into a new entry in the leaf node. If there is no empty entry, the leaf is split into two, which may cause additional splits in the parent nodes.

3.8. Clustering of the Paths Followed by the Visitors

Different from the most of the other usage mining systems, the system provides a profiler for obtaining more accurate, reliable and detailed information about the behavior of the visitors of the web site, Shahabi et al. [3], Natascha[59].
The data obtained by this profiler is then used for obtaining the paths followed by the visitor and times spent on each page of the paths. Then, these paths are clustered by using Path-Mining method. The profiler provided by the system works on the client side.

It is a Java applet loaded into the client side with the first page request and staying in the client cache afterwards. A call to this applet is added to each page in the site. The aim of the Java applet is to determine exact viewing time of the pages and catching the page views missing due to the retrieval of them from the client cache instead of a server.

In addition to the profiler, each link on each page is updated to make it transfer more information to the server side when clicked. This information will be used for determining which links the visitor selects. Then, the link names will be added into the paths so that each pair of page requests are separated by the link, which is selected for retrieving the second page.

The authors indicate that link information may provide additional clues on user behavior especially if two links from the same page are pointing to the same page. After the page requests and times spent on them are determined for each visitor, paths found are clustered for obtaining the groups of visitors with similar interests by using the Path-Mining methodology. Via the usage of this methodology, the order of the requests in the path is also taken into consideration on the contrary to the work explained in the previous section.

To be able to this, the system needs a way of measuring the similarity between two paths. The similarity between two paths is measured by finding the angle between them. Briefly, the angles between two paths are calculated by using
the inner product over the feature space where feature space contains all sub paths of these two paths. After the angles between each pair of paths are calculated, the results are fed into k-means algorithm for finding clusters of paths. The resulting clusters are considered to be containing groups of visitors with similar interests.

3.9. System Improvement

Research on System Improvement aims to use web usage mining for improving the web traffic and increasing the speed at which the visitors are responded.

One way to do this is to provide the web server with the capability of guessing the pages that may be retrieved by the visitors next and generate the dynamic content of these pages before user retrieves them.

For guessing the pages, the system proposed by Schecter et al. makes use of the concept of path profile, which is constructed from the data contained in the web logs, Schecter and Smith [16], Liu[66].

Path profile is the set of paths followed by the visitors of a site and the number of people following them. This system provides an efficient technique for generating and storing the path profiles.

The paths are stored in the form of a tree in which the paths are merged on their common prefixes. While constructing the tree, only the paths whose maximal prefix is seen in at least T of the paths are added into the tree for reducing the memory cost.
By following this rule, the algorithm for forming the tree is run on the tree more than once to be able to obtain all paths suiting to the threshold value. A path profile is then used by online working part of the system for guessing the next access of the user.

The system starts with the shortest suffix of the current user path and tries to find a path whose maximal prefix matches with it. As long as a matching path whose maximal prefix equals to the suffix in hand is found, suffix size is increased. Assume that the pages retrieved by the visitor An are as follows: [P1, P2, P3].

In that case, the system initially checks the tree for finding the paths that have a maximal prefix [P3], the smallest suffix of the user path. Then, the suffix size is increased by one and the paths that have [P2 P3] as maximal prefix are found if there exists any.

Assume that [P2 P3 PY] is such a path. Increase in the suffix size continues as long as the corresponding paths are found. If there is a path [P1 P2 P3 PX] in the tree, the system creates the dynamic content for the page PX automatically.

If there exists no path having [P1 P2 P3] as maximal prefix, then the system will create the dynamic content for the page PY.

Another prediction technique is named as point based prediction in which the next page is guessed only by considering the last page retrieved not the whole path.
Experiments with the system show that agreement prediction technique gives the most accurate results. In this technique, dynamic content creation of a particular is done only if both point and path based prediction techniques agree on that page.

3.10. Personalization

Because of the increasing demand to the e-commerce, many companies are eager to make their sites that exhibit their products more serviceable and effective for their visitors to be able to turn them into customers, Oosthuizen[65].

The number of people visiting the web site of a company may be too high whereas only small percentage of the visitors may be turning into a customer. The number of customers gained through web site heavily depends on the success of the site and personalization is critical aspect of this success.

Web Personalization simply means to understand the needs and interests of the visitors of the site and respond accordingly. Such a web site recognizes each visitor and customizes itself by various ways such as determining the information that should be shown to the visitor or automatically changing the site structure in a way that will be useful and attractive for the current user, Adomavicius[76].

Personalization is attractive research topic, because it is critically important for the success of e-commerce companies. Some of the different techniques for personalization will be explained in the subsequent sections.

3.10.1. Content Based Filtering

The main idea of Content Based Filtering is to make use of content similarity between stated user interests and web pages for personalization, Satoğlu [9], tug[67], tam[78], Iijima[77].
WebWatcher is an agent that trusts on content based filtering for personalization of the web sites. It guides visitors during their navigation through the web site according to their interests.

At the beginning of a visit, WebWatcher asks user to enter his interest or the thing that he is looking for in the form of keywords. By using this information, WebWatcher highlights the links that are best suited to the needs of the visitor on each page retrieved by him/her throughout his visit.

WebWatcher accomplishes the task of choosing the best links for that user by using the information learned from the past users. In addition, the actions performed by each visitor are continuously used as training samples for improving the performance of the tool in future recommendations.

Three different learning techniques are tried in WebWatcher: Learning from previous tours, Learning from Hypertext Structure and the combination of first two. First method proposes to store a description for each link in each page in the form of a high dimensional feature vector whose elements are English words.

Interests of the users are also represented by a feature vector. Whenever a visitor follows a link in a page, the interests of the user, which consists of some number of keywords, is added to the description of that link. What are used in choosing the links to highlight in each page is the descriptions of the links determined as a result of this learning mechanism.
While choosing the links that will be recommended to the visitor, WebWatcher calculates the similarity of each link in the page to the interest of the user. The links that will be highlighted are the ones with the highest similarity values with the user interest.

To learn from the hypertext structure, WebWatcher makes use of Reinforcement Learning. If we take an agent moving across states as an example case, the aim of the Reinforcement Learning is to train the agent so that it will reach the final state from the initial state by choosing the best action to take in each state it encounters. Here, the action means choosing a next state to go.

Goodness of choosing an action \( a \) in state \( s \) is represented as \( Q(s; a) \). The optimal strategy is to choose an action that will maximize the \( Q \) value for the current state.

Turning back to hypertext environment, pages are the states and links are the actions. The system learns \( Q(s; a) \) function for each page and word pair, which means that the best action to take is different for different words in the same page. So, in each page the system recommends the hyperlinks, which maximize the total of \( Q \) values, belong the current page and words given as an interest of the user.

The third method, which is detected to be giving the best results, combines the results obtained from first two methods plus two additional methods. The first additional method chooses the links that are mostly preferred while the second method chooses the links whose textual content is most similar to the interest words of the current user.
3.10.2. Usage Based Web Personalization

Most of the recent research on personalization aims to incorporate pattern discovery with personalization, resulting in a usage based Web personalization or customized usage tracking, Cooley [32], Chen[69] and Tsai[75].

In that case, profiles of the visitors are dynamically created according their access patterns. Dynamic creation of profiles is advantageous when compared to the profiles specified by the visitors themselves.

Older personalization tools and techniques rely on that kind of profiles, which are static and most probably biased. As the time passes, user preferences may change although static profiles remain unchanged which decreases the performance of the personalization system.

On the other hand, dynamically created profiles capture the current interests of users. Dynamically created profiles cannot be hundred percent faultless, but they achieve considerable amount of success in helping users without waiting for the user asking for it. Usage based Web Personalization systems generally comprise two major components:

Offline component and Online recommendation engine. Offline component of the system analyzes the log files, which contain the footprints of all visitors visiting the site. First it puts the data in the logs into a form that is amenable for applying data mining techniques. Analysis of log files by various data mining techniques result in aggregate usage profiles, which are common profiles of visitors of the web site.
Then, the online component of the system matches the current user to these profiles based on his navigation pattern up to that point and customizes the current page accordingly.

Customization can be done through recommending some links or putting advertisements or product news that may interest the customer. Throughout the following sections, we will explain current Web Personalization systems and tools in more detail.

3.10.3. Analog

Analog, Yan et al. [38] is one of the first usage based Web Personalization systems. Its offline module clusters the users of a web site according to their access patterns.

Offline module first processes the log file of the target site to find out user sessions, which are represented as n dimensional vectors where n is the number of distinct pages in the site. The weights of the entries corresponding to the pages visited by the visitor are larger than 0, while the weights for no visited pages are zero in the session vector. Therefore, the system does not take into account the order in which pages are retrieved. After all session vectors are obtained in this way, LEADER algorithm is applied to find clusters.

LEADER is a simple clustering algorithm, which has some drawbacks. After clustering is completed, median vector of each cluster is computed as a representative of the cluster. The online module of the system recommends some links to the active visitors by looking at the pages that they retrieve before.
Active user sessions are represented as n dimensional vectors as in the offline module. Whenever user retrieves a new page, the session vector belonging to that user is updated accordingly. Online module of the system tries to match the active user session to existing clusters. User session is accepted to be matching to a particular cluster if the number of common pages between user session vector and cluster median is larger than some threshold value. The pages in the median vectors of matching clusters are then recommended to the user if they are not already retrieved by him/her.

3.10.4. Web Personalizer

Web Personalizer, Cooley [32], is one of the other systems that make use of the explained framework for usage based Web Personalization.

The main aim of the offline module of the system is to obtain aggregate usage profiles, which are represented as weighted collection of URLs. The reason for preferring this representation style is to be able to make use of classical vector operations that are used in clustering.

Two different methods for forming the aggregate usage profiles are presented in. The first method is to cluster the user sessions by using standard clustering algorithms for grouping the visitors that have similar interests together as in Analog.

This method proposes to represent each user session, as an n-dimensional vector where n is the number of distinct URLs that exist in the user sessions. The values kept in each entry of the user session vector can be chosen to be binary to indicate the existence or nonexistence of that URL in that session. After putting
them into the vector form, user sessions are clustered by using classical clustering
techniques from data mining to obtain session clusters.

The next step to form Aggregate Usage Profiles is to find the mean vector of
each cluster. Entries in the mean vector of a cluster are calculated by finding the
ratio of the number of user sessions that contain the URL that is represented by that
entry to the total number of sessions in that cluster. Because of this calculation,
some of the URLs are filtered out because of having very low support, which means
that only minority of the user sessions in the cluster contain them. The resulting
mean vectors are representative aggregate usage profiles for the log data processed.
The other method proposed in clusters URLs instead of sessions. It is indicated that
users that have very different sessions may have common interest to a group of
URLs. This information will remain undiscovered with the previous method. At the
end of this method, each cluster will contain a set of URLs which tend be together
in majority of the sessions. Standard clustering algorithms are difficult to be applied
in that case because of the nature and the size of the feature space, which consists of
the sessions.

Therefore, another clustering technique, which is named as Association Rule
Hypergraph Partitioning (ARHP), is employed by this method. The hyper graph to
be clustered by this technique composed of URLs as vertices and the frequent item
sets as the hyper edges, which connect the vertices representing the URLs in that
item set. As known, frequent item sets are formed by a well-known technique from
association rule mining.

Application of the ARHP technique on the hyper graph obtained by this way
results in a set of clusters which contain a set of URLs that are frequently accessed
together. Usage profile for each session is obtained by associating a connectivity
value of the vertex as a weight for the corresponding URL.
As it is in the other usage based personalization systems, online component of the system keeps track of the active user sessions to recommend some links attached with significance scores to the users.

The way to do this is to find aggregate usage profiles that match to the current user session best.

The matching scores are calculated by standard distance and similarity measures between vectors. Besides, site structure becomes effective in calculating the matching scores by increasing the score of the pages that are farther away from the current page. History depth is an important concept employed by the online component of the system for obtaining more successful recommendations. It determines the number of previous pages that will be effective on the recommendation. It is indicated that user sessions are mostly composed of some number of episodes, which are paths, followed for reaching different kinds of information. The length of episodes is indicated to be 2 or 3 in general.

Therefore, by the help of the concept of history depth, it is aimed to make recommendations based on the pages retrieved only in the current episode.

3.11. Site Modification

Another way of benefiting from the usage data discovered from the logs is to use it to improve the design of the web site.

In personalization, web sites are dynamically customizing themselves differently for each visitor. On the other hand, site modification systems offer static changes in the structure and content of the web sites to meet the needs of all visitors, Perkowitz and Etzioni [17].
IndexFinder is one example for that kind of tools. It aims to discover index pages whose addition is very likely to improve the site design. These pages, which are created offline, consist of links to the conceptually related, but currently unlinked pages, which coexist in most of the user sessions. The addition of automatically created index pages to the site is performed with the authorization of the Web Master. Index page creation in IndexFinder is performed in three phases: processing logs, cluster mining and conceptual clustering.

In the first stage of the algorithm, a log is processed to be divided into visits. What comes next is the calculation of the co-occurrence frequencies between each pair of pages to determine to what extend these pages are related. Co occurrence frequency between two pages is simply calculated by taking the minimum of the two probabilities, probability of the existence of first page in the visit given the fact that second page is in the visit and visa versa. The concurrence frequency between linked pages is taken as zero to avoid uninteresting clusters.

After the concurrence frequencies are calculated, a similarity matrix is constructed which is then converted into a graph form by taking the pages as nodes and concurrence values as edges between these nodes. Naturally, the nodes will be unlinked if the concurrence frequency between the pages denoted by them found to be 0 from the similarity matrix. The connected components in the graph built in this way are accepted as clusters. The pages corresponding to the nodes of a connected component found by this way are put into one cluster.

The Cluster Mining algorithm is PageGather, which differs from the other clustering algorithms because of not insisting on putting every instance in one and only one cluster. Instead, the algorithm discovers small number of high quality
clusters. The clusters obtained by this way may contain pages that are conceptually unrelated.

Yet, the aim of the IndexFinder system is to produce index pages that contain links which are conceptually related in addition to be visited together by the majority of the visitors. This constraint is satisfied by applying a concept-learning algorithm on the clusters found in the previous step. To be able to apply that algorithm, first each page should be tagged manually with the correct values for predefined enumerated concepts. Concept learning algorithm finds the most common and basic concept that summarizes the pages in the cluster.

Then, the noisy pages that conflict with the concept found are removed from the cluster while the nonexistent pages that conform to the given concept are joined to it. Each cluster obtained by this way is used to form one index page, which is composed of the links to the pages that are in that cluster. The candidate pages are presented to the web master who will give the final decision on the addition of these pages to the site and the location and the title of them.

3.12. Limitations of the available Web Usage Mining Miners:-

Web usage mining is the application used in data mining to analyze and discover required hidden patterns of user’s usage data on the web. The usage data records the user’s interactions when the user browses or makes transactions on the web site. An activity involves the automatic discovery of patterns from one or more Web servers. Organizations often generate and collect large volumes of data; most of this information is usually generated automatically by Web servers and collected in the server log. Analyzing such data can help these organizations to determine the importance/value of particular customers, cross marketing strategies across products
and the effectiveness of promotional campaigns, etc. Some of the limitations of existing may be are as follows.

- Most of the existing tools capture the data after it is stored in log file, which may contain error messages and noises.

- There is a great delay in generating the reports and reports generated are with less accuracy since they pick the data form a huge log (bulk file).

- Many of the tools requires complicated configuration for settings.

- Majority of the existing miners collect all the user behavior including the technical data into the log and Data mining repository is a group of all activities, which are collected from the web pages including unwanted technical information like http objects, cookies etc.

- Large number of tools starts their work after the user behavior is collected and stored in one log file.

- Most of the existing miners are standalone tools and data transformation is done with static transformation settings.

In the light of above limitations, we have designed a framework which can cover most of the activities of web usage mining with a tool known as "Online Miner".