Vrije Universiteit Brussels

Faculty of Science

Department of Computer Science

Academic Year 2005-2006

Techniques for Personalization in E-learning

By: Ahmed Abdalaal

Thesis presented as partial fulfillment of the requirements to obtain
the degree of Master after Master of Science in Computer Science

Promotor:

Prof. Dr. Ir. Geert-Jan Houben

June 2006

Acknowledgements

I would like to express my gratitude to my promoter, Prof. Dr. Ir. Geert-Jan Houben, for his valuable guidance and support. Furthermore, I would like to say a big thanks to my family for their patience and supporting me all this time.

Abstract

Creating a personalized web site to facilitating web navigation is a challenge for web site designers. One kind of this system dynamically gives recommendations to website users by learning from web usage mining and users' behavior. In this paper, I am going to use *web usage mining* technique for creating an adaptive recommendation system for navigating a web site. This technique discovers the patterns (association rules) from Web logs data coming from HTTP servers. After a brief explanation of the terms *Web usage* and *Web usage mining*, this document focuses on the processes of data preparation and transaction identification which lead to the discovery of rules and usage patterns. The end-use of these patterns could be for feed back into the design process and for future decisions or, in this case, to automatically adapt aspects of the site based on previous usage patterns. It will deliver different recommendation links for every learner.

This system can be used as possible approach in the e-learning domain to improve learner's satisfaction, by attempting to personalize the learning navigation environment for the learner.

Contents

List of Figures

List of Tables

1 Introduction

With the rapid growth of information on the World Wide Web, navigating a website easily becomes a very important issue. When designers want to implement web site, a lot of them immediately think of graphics and of visual design. However the core design challenges for a web site revolve around information, not visuals. Designers do usually not think of how to present users with the most user-friendly path through the classification so that they find the content they want quickly, how to allow users to move quickly and logically through the web site, or how to highlight for the reader parts of the website that the organization wants to promote.

In this work, I propose a solution to this problem based on an intelligent technique. That assists the user in navigating a particular website by providing recommendations, restructuring the website or highlighting important links. This technique is based on results of mining web log data and observing users' behavior. For the user, a personalized navigation can be built to surf the web and construct logical views that are of interest. For the designer of a web site, It provides a comprehensive analysis to the website structure, which helps designers to reconstruct the website structure in a more efficient, effective and usable way.

1.1 Structure of this document

Chapter 1: This chapter considers the navigation, personalization and e-learning concepts. In addition, it takes into account the optimization and customization issues. This report is structured in the order of development with chapters considering the following content.

Chapter 2: Process of knowledge discovery the previously published work on data-mining algorithms, with specific attention paid to the discipline of web-usage mining. Research results and algorithmic suggestions are presented in the order of a data-mining pipeline, from collection of data to isolation of interesting patterns.

Chapter 3: Design and Implementation the overall implementation process for the web-usage mining with more details for data preparation, pattern discovery and pattern analysis phases. It includes the description of counting technique and Apriori algorithm which are used in pattern discovery phase.

Chapter 4: Conclusion summarizing the work carried out is presented. The implementation process is evaluated, some interesting areas are represented and finally the future direction of the work is considered.

1.2 Motivations

Nowadays, web-based applications are used in various domains such as the learning domain. There is a lot of information with complex structure and various kinds of users. Nevertheless, these complex web-based applications will only succeed in attracting more users if they are easy to interact with. The novice users will go out when the website structure becomes too complex to be explored without appropriate assistance. Furthermore, a number of studies have shown that poor navigation leads users to become rapidly frustrated, which results in a risk of having them leave the website, fail to recognize or visit interesting part of the website, or simply complete their visit with a feeling of not having adequately explored it. For this reason, a number of research groups have been developing ways to assist the user when using a web-based application by providing recommendations for the next link through its navigation, for instance.

1.3 Literature review

This report documents the development of a technique allowing simple websites to exhibit navigational structures that adapt to the needs of their users. In the past, a vast number of technologies would have been required to support the development of such personalized techniques. Using the technique presented in this document, however, a simple website usage log processed with the Apriori algorithm can be used to power adaptive functionality on the static or dynamic pages.

This review chapter introduces the field and the rationale behind the work, and presents the navigational, personalization and e-learning concepts.

1.3.1 Navigation concepts

The visitors surf around a website by following hypertext paths from page to page. If a site has a lot of information with all sorts of attractions, but visitors don't know how to get to it, or if they find the navigation confusing or complicated, they will simply stop exploring the rest of the site, and maybe never return. Therefore, good navigation design is an essential part for any successful web site. A visitor wants to know these things:

- Whose site is he/she looking at?
- Where is he/she in the site?
- What else is available at this site?
- Where should he/she go next?
- How does he/she find what he/she is looking for?
- How can he/she get back to where he/she came from?

Therefore, Navigation should be consistent across the site, consistent with WWW standards, adapted to the target audience, and reflecting the structure of the site.

There are three basic types of navigation [1]:

Hierarchical: which applies to sites that are information-rich and are best organized as a large tree- much like a library.

Global: which applies to sites where it can easily and logically jump among all points; this is best if it can present information in fewer, broader categories.

Local: navigation sits somewhere in between. This applies when you have depth of information within broader areas.

1.3.2 Personalization concepts

On a Website, personalization is the process of gathering and storing information about site visitors, analyzing the information, and, based on the analysis, delivering the right information to each visitor at the right time. It tailors pages to individual visitors' characteristics or preferences. Personalization is a means of meeting the visitor's needs more effectively and efficiently, making interactions faster and easier and, consequently, increasing visitor's satisfaction and the likelihood of repeat visits.

Whereas the personalization process modifies the content or even the structure of a website automatically, by the system based on information concerning the user's stored or even the log files, customization lets users arrange the interface or modify the structure manually. The control of the structure, look or content of the site is explicit and user-driven. i.e., the user is involved actively in the process and has direct control.

The selection of personalization technique should be directed by the Website type in order to provide personalization for real-time applications that will affect the system performance. How personalization is deployed is thus important and needs to be integrated into the overall system design. This is especially true for high-volume Web sites.

There is process to provide a personalized Website. Figure 1 is an overview of personalization process. The major steps may or may not be performed dynamically; part or all of some steps may be performed offline.

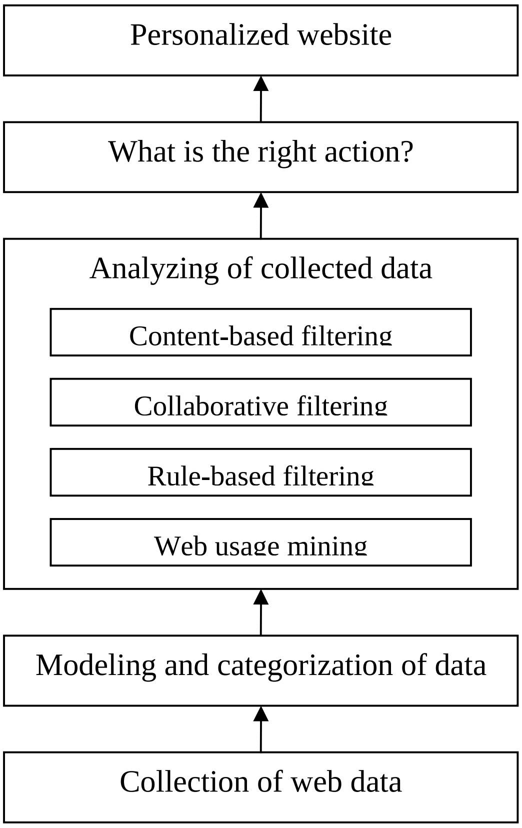


Figure 1: Overview of Personalization process

The objective of collecting visitor information is to build a profile that describes a site visitor's interests, role in an organization, entitlements, purchases, or some other set of descriptors important to the site owner. The most common techniques are explicit profiling, implicit profiling, and using legacy data:

- Explicit profiling: asks each visitor to fill out information or questionnaires by a specific form. This technique has the advantage of letting users tell the site directly what they want to see.
- Implicit profiling: tracks the visitor's behavior. This technique is generally transparent to the visitor. The browsing surfing is usually tracked by saving specific visitor identification and behavior information in log file which keeps the user hits or by a cookie files that is kept at the browser and updated at each visit. For example, Amazon.com logs each customer's buying history and, based on that history, recommends specific purchases.

Analyzing of collected data

When the visitor collected data is available, the next step is to analyze this information in order to present or recommend documents or any actions specific to the visitor. Making such recommendations is the most challenging step. Many ways can be employed in order to analyze the collected data include content-based filtering, collaborative filtering, rule-based filtering, and Web usage mining.

1. Rule-based filtering:

It allows website administrators to specify rules, based on static or dynamic profiles. That then used to affect the information served to a particular user [2]. This requires the administrator, most likely with the help of a consultant, to figure out the appropriate rules. Cross-selling is an e-business example of the rule-based technique. For example, a rule could be specified to offer product X to a customer who has just bought product Y; for example, a customer of a book might be interested in current or previous books by the same author or in books on the same subject. Rule-based techniques can be used with filtering techniques, either before or after the filtering process, to develop the best recommendation.

2. Content-based filtering

It tracks user behavior and preferences, recommending items that are similar to those that users liked in the past [3]. So it works by analyzing the content of the objects to form a representation of the visitor's interests. Generally, the analysis needs to identify a set of key attributes for each object and then fill in the attribute values. One example is a document filtering system that analyzes documents based on keywords. Content-based filtering is most suitable when the objects are easily analyzed by computer and the visitor's decision about object suitability is not subjective.

3. Collaborative filtering

It compares a user’s tastes with those of others in order to develop a picture of like-minded people. The choice of material is then based on the assumption that this particular user will value information that like-minded people also enjoyed [4]. The user’s tastes are either inferred from their previous actions or else measured directly by asking the user to rate products. Collaborative filtering develops recommendations by finding visitors with similar tastes. Recommendations produced by collaborative filtering are based on the peer group's response and are not restricted to a simple profile matching. For product recommendations, collaborative filtering is most suitable for homogeneous, simple products, such as books, CDs, or videos.

4. Web usage mining

It relies on the application of statistical and data-mining methods based on the web server log data, resulting in a set of useful patterns that indicate users’ navigational behaviors. The patterns discovered are then used to provide personalized information to users based on their navigational activity [5].

In general, different analyzing techniques are most suitable for different variables, such as type of Web site, Web site component, or product/services. Consider the case of product recommendations. Selling books or CDs requires techniques different from those required to sell computers. A technique that improves on the best of the current techniques and offers additional options could satisfy a wider set of needs. With a flexible architecture that allows for multiple recommendation engines, each engine would use specific analyzing techniques to make its recommendations.

The information provided to the user through any of the above techniques can be adapted at three different levels [6]: Adaptive content, Adaptive navigation and Adaptive presentation.

- *Adaptive content* selection is based mostly on adaptable information retrieval techniques: when the user searches for relevant information the system can adaptively select and prioritize the most relevant items. By doing so, the user can obtain results that are more suitable for their knowledge capabilities.
- *Adaptive navigation* support is founded mainly on browsing-based access to information: when the user navigates from one item to the other the system can manipulate the links to guide the user adaptively to most relevant information items.
- Finally, *adaptive presentation* is based on adaptive explanation and adaptive presence, which were largely developed in the context of intelligent systems: when the user gets to a particular page the system can present its content adaptively [6].

The possibilities of content and presentation adaptability are a relevant element in the reuse of the same resources for different purpose, provided they have been correctly customized in advance. Considering the high cost of personalization, adaptability of resources can also offer an interesting byproduct in term of reuse of the same resources in different contexts, provided that their description is correctly defined through standard metadata applications to allow interoperability of the same service in different environments.

1.3.3 E-learning concepts

The Personalization techniques are domain-independent, meaning that the techniques used are usually general and can be applied to any domain, e.g. any Web Site, without concern about the context of the Web site. In this work I choose the E-learning domain to apply the technique on.

> E-learning can be viewed as an innovative approach for delivering well designed, learner-centered, interactive, and facilitated learning environment to anyone, anyplace, anytime by utilizing the attributes and resources of various digital technologies along with other forms of learning materials suited for open, flexible, and distributed learning environment, [7].

However, e-learning suffers from the lack of learner satisfaction are due to the “one size fits all” approach that most current e-learning course developments follow, delivering the same navigation with same static learning experience to all learners, irrespective of their prior knowledge, experience, preferences or learning goals. Many approaches have been developed to solve this dissatisfaction by concerning on personalization issue for the learning experience for the learner, such as Adaptive Educational Hypermedia (AEH) [8].

1.4 The Nature of Websites

To achieve the goals of this work such as improving website navigation, decreasing the complexity of the structure or providing a useful recommendation, we focus on web personalization as a process of customizing a Web site to the needs of specific users by taking advantage of the knowledge acquired from the analysis of the user’s navigational behavior (log files) in correlation with other information collected in the Web context, namely structure, content and user profile data. There are a variety of techniques (mentioned in Sec 1.3.2) to implement this kind of work. The decision to choose the appropriate technique depends on the nature of website. In the following sections, I will present these types then specify the reasons that empower me to choose certain technique.

To realize adaptive method for the website, designers should be able to use statistics and patterns extracted from the web-usage logs to modify a website. This modification could be performed either statically - adapting the web page files directly - or dynamically - altering the page 'on-the-fly'. The dynamic or static nature of the modification will be based largely on the nature of the website.

1.4.1 Statically Driven Websites

Statically driven websites are based on code that only changes when updated by the Webmaster. The code is in Hypertext Markup Language (HTML) and stored in plaintext files on a web server. Static pages could easily be adapted directly in the HTML source. Freshening the content requires the webmaster to manually visit and update the HTML that makes up each page that needs to be changed. Typically, this is done by editing a set of files on the *web server*, where each file represents a single page. Websites can typically grow to include thousands of files, this can become a difficult task, therefore. If we want to change a specific link in the website, the webmaster has the responsibility of changing this link. If each page in the website includes this link, he/she needs to updates every page. He then needs to make sure that no hyperlinks on other pages point to missing pages.

This becomes a lot of work very quickly. As that could be imagining, with any more than a few pages to update every day, this can become boring. The webmaster also understandably makes mistakes (he is human, after all), and forgets to update or remove critical pages.

1.4.2 Dynamically Driven Websites

A fully dynamically driven website can completely customize the contents of each web-page according to various inputs. These could be, for example, user input, database data or the result of executing a system call. Dynamically driven sites are usually created with the help of a scripting language like ASP, Perl, PHP or JSP, though they could be compiled from a language such as C. Because the HTML code (and therefore possibly the links on the page) are generated during script execution (and perhaps based on some external data or input) it is not possible to tell from the source code what the structure of the page will be. Web pages of this type could, therefore, not be adapted directly, though the languages used are powerful enough to support an API that could interface with statistics and pattern data to create or modify links as required.

1.4.3 Adapting Websites

There are a lot of literatures considering the adaptation of websites and the use of web-usage mining as a tool for optimization. Also the data-mining papers available appear to follow a chronological path through initial ideas, development and then proposed solutions, and the majority of the articles available on adapting websites appear to present suggestions for frameworks or suggest the direction that future development should take.

Adaptive Websites: [9] Websites that automatically improve their organization and presentation by learning from user access patterns. Therefore Adaptive web site can serve our goal because of the following:

- What a visitor seeks in a site depends on who the user is. An adaptive site can recognize typical user types and customize the site presentation appropriately. Instead of customizing to a single user (as many sites do with cookies), an adaptive site aggregates the experiences of many visitors over time to generalize customize the site for different types of users.

- Visitors to a site do not always have the same conceptual model of the material as the site's designer. An adaptive site can recognize when user expectations differ from the site structure.
- Although a web site's structure is usually static, user needs change with time. An adaptive site can learn these patterns and decide what information to present when.

1.5 Optimization and Customization

After discover usage patterns, its possible to dynamically adapting the user model, therefore, a series of optimization transformations can be made to a website are proposed in [10]. They are:

- Promoting and Demoting: promotion being the movement of a link up the site hierarchy towards the front page, and demotion being the movement of a link down the site hierarchy.
- Highlighting: drawing attention to an existing link (changing fonts, colors etc).
- Linking: creating a link between two previously unconnected pages
- Clustering: content clustering pages with the intention of providing index pages linking to similar pages

1.6 What this thesis provides

Knowing of information about users' behaviors and their usage patterns, can lead to interesting results that go over descriptive tasks; such examples are dynamic content Web sites which perform mass customization and personalization by discovering groups of users with similar access patterns and by adding navigational links and hints on the fly. Also, information mined from Web usage data allows restructuring and better management of the website, giving more effectiveness to it; the network system, too, can gain benefits from this discovery process. In this work I try to achieve these goals.

- Decreasing the amount of time required to navigate a site to popular destinations.
- Decreasing confusion caused by unintuitive site hierarchies.
- Suggesting alternative pages to users that may be of interest based on previous visitor patterns

2 Process of knowledge discovery

Web Mining has been proposed as a unifying research area for all methods that apply data mining to Web data. The typical sub categorization of the work in Web mining falls into the following three categories [11]: Web Content Mining, Web Structure Mining and Web Usage Mining. Web Content Mining is concerned with the extraction of useful knowledge from the content of Web pages, with the use of data mining. Web Structure Mining is a new area concerned with the application of data mining to the structure of the Web graph. Web Usage Mining aims at discovering interesting patterns of use, by analyzing Web usage data. Out of the three categories of Web Mining, Web Usage Mining is the one mostly related to personalization. The advantage of viewing Web personalization as an application of Web Usage Mining is that the work on Usage Mining can be a source of ideas and solutions to some of the problems encountered in personalization research.

The term *Web usage mining* was introduced by Cooley et al in 1997 [12] when a first attempt of taxonomy of *Web Mining* was done; in particular they define Web mining "as the discovery and analysis of useful information from the World Wide Web". Also it is defined as "the application of data mining techniques to large Web data repositories."

2.1 Introduction

Data Mining and Knowledge Discovery is an active research discipline involving the study of techniques which search for patterns in large collections of data. Meanwhile, the explosive growth of the World Wide Web in recent years has turned it into the largest source of available online data. Thus, the application of data mining techniques to the web, called *web data mining*, was the natural subsequent step and it is now the focus of an increasing number of researchers. In web data mining there are currently three main research directions [13]:

i. Web Content Mining
ii. Web Structure Mining
iii. Web Usage Mining

Mining for Web Content focuses on the development of techniques to assist users in processing the large amounts of data they face during navigation and to help them find the information they are looking for.

Mining the link structure aims to discover the link structure of the hyperlinks at the inter-document level. Based on the topology of the hyperlinks, Web structure mining will categorize the Web pages and generate the information, such as the similarity and relationship between different Web sites. So it develops techniques to take advantage of the collective judgment of web page quality in the form of hyperlinks, which can be viewed as a mechanism of implicit endorsement. The aim is to identify for a given subject the authoritative and the hub pages. Authoritative pages are those which were conferred authority by the existing links to it, and hubs are pages that contain a collection of links to related authorities.

The other research direction, which is being followed by an increasing number of researchers, is mining for user navigation patterns [14]. This work focuses on techniques which study the users' behavior when navigating within a web site. Understanding the visitors' navigation preferences is an essential step in improving the quality of personalization issue. In fact, the understanding of the most likely access patterns of users allows the service provider to customize and adapt the site's interface for the individual user and to improve the sites static structure within the underlying hypertext system.

When web users interact with a site, data recording their behavior is stored in web server logs, which in a medium sized site can amount to several megabytes per day. Moreover, since the log data is collected in a raw format it is an ideal target for being analyzed by automated tools. Currently several commercial log analysis tools are available; however, these tools have limited analysis capabilities producing only results such as summary statistics and frequency counts of page visits. In the meantime the research community has been studying data mining techniques to take full advantage of information available in the log files.

There have so far been two main approaches to mining for user navigation patterns from log data. In the first approach log data is mapped onto relational tables and an adapted version of standard data mining techniques, such as mining association rules are invoked. In the second approach techniques are developed which can be invoked directly on the log data.

The Web usage mining process can be regarded as a three-phase process [Figure 2], consisting of the data preparation, pattern discovery, and pattern analysis phases [15]. In the first phase, Web log data are preprocessed in order to identify users, sessions, page views and so on. In the second phase, statistical methods, as well as data mining methods (such as association rules, sequential pattern discovery, clustering, and classification) are applied in order to detect interesting patterns. These patterns are stored so that they can be further analyzed in the third phase of the Web usage mining process.

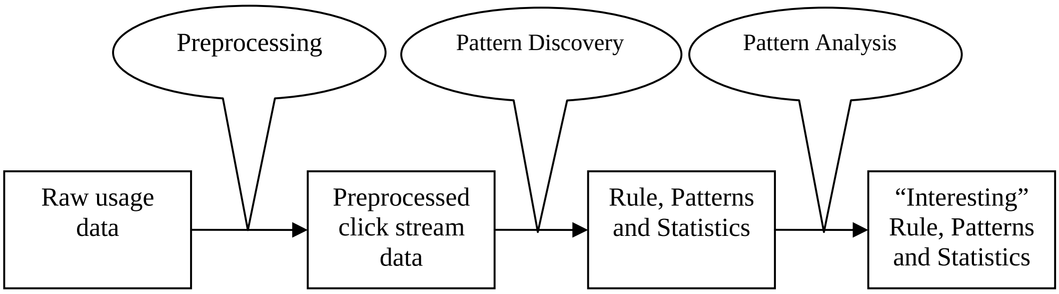

Preprocessing
Pattern Discovery
Pattern Analysis
Raw usage data
Preprocessed click stream data
Rule, Patterns and Statistics
"Interesting" Rule, Patterns and Statistics

Figure 2: Data usage mining process

In the next sections, a description of the fields included in a log entry of a Web usage log follows, along with a set of definitions of Web data abstractions, such as Web site, user, session, page views, and click streams. Technical issues concerning data preparation will be discussed. A more detailed analysis of the methods employed in the Web usage mining process including simple log analysis will be presented.

2.2 Data Preparation

In Preprocessing phase, there are two main stages 1) *Data Collection*. During this stage, data are collected either from Web servers or from clients that visit a Web site. 2) *Data Preprocessing*. This is the stage that involves primarily data cleaning, user identification and user session identification.

2.2.1 Terminology

Some definitions of the specialist terminology will be used in the following chapters. These definitions are based upon those defined by the World Wide Web Consortium (W3C) in [16].

Client: The role adopted by an application when it is retrieving and/or rendering resources or resources manifestations.

Server: The role adopted by an application when it is supplying resources or resource manifestations.

Proxy: A proxy is an intermediary which acts as both a server and a client for the purpose of retrieving resources or resource manifestations on behalf of other clients. Clients using a proxy know the proxy is present and that it is an intermediary. E.g. HTTP firewall proxy

Cache: A Proxy that stores a copy of recently requested pages and can subsequently serve them to other users without having to make another request to the Server.

Request: A message describing an atomic operation to be carried out in the context of a specified resource.

Page view: A series of Requests that together allow the Client to display a complete web-page (including graphics). Represented in this document by a single request - for the file returned containing HTML.

User: The principal using a client to interactively retrieve and render resources or resource manifestations.

User session: A delimited set of user clicks across one or more Web servers, characterized by a Click Stream.

Click Stream: The representation of a user-session as a sequence of page-views. If extracted from a server-side log, Click-streams should be assumed to be deficient because requests satisfied by a local or intermediate cache will not be present.

2.2.2 Data Collection

Web Server access log files, are the main input data. Each entry in the access log file represents a request from the Web server. There are several different standard formats—the three most popular types of log file formats are:

- NCSA Common Log File Format.
- NCSA Extended Log File Format
- W3SVC IIS Log File Format

W3SVC IIS Log File Format

The W3C Extended log file format is the default log file format for IIS. However, the log files used in this work are from this format. Administrator can use IIS Manager to select which fields to include in the log file, which allows you to keep log files as small as possible. The following table describes the fields of this format [17]. Table 1 contains well-explanation for the log file fields, whiles I will use number of it in the preprocessing phase.

Table 1 W3C Extended Log File Fields

Field	Appears As	Description	Default Y / N
Date	Date	The date on which the activity occurred.	Y
Time	Time	The time, in coordinated universal time (UTC), at which the activity occurred.	Y
Client IP Address	c-ip	The IP address of the client that made the request.	Y
User Name	cs-username	The name of the authenticated user who accessed the server. Anonymous users are indicated by a hyphen.	Y
Service Name and Instance Number	s-sitename	The Internet service name and instance number that was running on the client.	N
Server Name	s-computername	The name of the server on which the log file entry was generated.	N
Server IP Address	s-ip	The IP address of the server on which the log file entry was generated.	Y
Server Port	s-port	The server port number that is configured for the service.	Y
Method	cs-method	The requested action, for example, a GET	Y

Table 1 W3C Extended Log File Fields

Field	Appears As	Description	Default Y/N
		method.	
URI Stem	cs-uri-stem	The target of the action, for example, Default.htm.	Y
URI Query	cs-uri-query	The query, if any, that the client was trying to perform. A Universal Resource Identifier (URI) query is necessary only for dynamic pages.	Y
HTTP Status	sc-status	The HTTP status code.	Y
Win32 Status	sc-win32-status	The Windows status code.	N
Bytes Sent	sc-bytes	The number of bytes that the server sent.	N
Bytes Received	cs-bytes	The number of bytes that the server received.	N
Time Taken	time-taken	The length of time that the action took, in milliseconds.	N
Protocol Version	cs-version	The protocol version —HTTP or FTP —that the client used.	N
Host	cs-host	The host header name, if any.	N
User Agent	cs(User-Agent)	The browser type that the client used.	Y
Cookie	cs(Cookie)	The content of the cookie sent or received, if any.	N
Referrer	cs(Referrer)	The site that the user last visited. This site provided a link to the current site.	N
Protocol Substatus	sc-substatus	The substatus error code.	Y

Table 1: lists and describes the available fields. Default fields are noted.

The main extension to the common log format is that a number of fields are added to it. The most important are: *referrer*, which is the URL the client was visiting before requesting that URL, *user agent*, which is the software the client claims to be using, and *cookie*, in the case where the site visited uses cookies. These fields will make the user session identification more accurate as will shown in next sections.

2.2.3 Data cleaning

Most of the information stored in an HTTP server log file is useless for the majority of knowledge discovery techniques. On an average size Web server, access log files easily reach tens of megabytes per day, causing the analysis process to be really slow and inefficient without an initial cleaning task. Every time a Web browser downloads an HTML document on the Internet, the images included are requested as well, causing each of those accesses to be stored in the log file of the server. If our pattern discovery objectives take no notice of images, it is understood that all of these entries have to be excluded from our analysis output files; similar cases of data cleaning regard entries with HTTP status code equals to 404 (resource not found on the server), particular resources with some defined characteristics (CGI applications for counters, URLs with some string patterns in their path that is not desired) and also accesses performed by the so-called spiders or crawlers or robots. These are automatic agents (Robot see Sec 2.2.6) that surf the Web in order to gather and store information; the most common case regards search engine spiders, which crawl the Web and index words and documents in databases to be queried by the end user, and link checkers and site managements tools [[18]]. A cleaning phase is certainly necessary; however its definition varies from case to case, depending on the aims and objectives of the whole analysis process.

So, once the data have been uploaded to the system then the preliminary preprocessing phase of data cleaning commences. This involves the process of applying filters to the log files, in order to remove data that are irrelevant to the specific site’s content and structure. These data are downloaded without a user explicitly requesting them, due to the definition of the HTTP protocol, and thus, they are not considered as part of the user’s actual browsing activity.

2.2.4 Session Identification

That's possible to use the login name as an identifier for users. In fact, the login name is known only for registered users. As a first heuristics, cookies could be used as user identifiers. On the one hand, this is an approximation: once a user deletes his/her cookies, he/she will appear as a distinct user. On the other hand, web browsers can be set not to accept cookies. The former approximation cannot be refined. For the latter, one could consider the pair (cookie, IP) in case the cookie is not present. Considering only the cookie leads to have a single user ("those who do no accept cookies"). Nevertheless, the pair (cookie, IP) can only track a single "visit" of a user and cannot be used to track users along two or more visits because IP can be reassigned, also users can use different computers at home and at work, etc.

Consider now the concept of user session. From a semantically point of view, a user session could be defined as the set of URLs accessed by a user for a particular purpose. By assumption that the amount of time a user spends examining an object is related to the interest of the user for the object contents. On this basis, a model for user sessions is obtained by distinguishing the navigational objects i.e., containing only links appealing to the user, from the content objects i.e., containing the information the user was looking for. The distinction between navigational and content accesses is related to the time period between a request and the next one. If between two accesses A and B there is a time delay greater than a given threshold, then A can be considered as a content URL; otherwise, it is a navigational URL.

On this basis, a user session is a sequence of navigational URLs that from the same IP address and User agent for the content URL. Various methods described in Table 2 which provide a comparison that explains the advantages and disadvantages for each method.

Method	Description	Advantages	Disadvantages
IP Address + User Agent	Assume each unique IP address/Agent pair is a unique user	Always available. No additional technology required.	Not guaranteed to be unique. Defeated by rotating IPs.
Embedded Session Ids	Use dynamically generated pages to associated ID with every hyperlink	Always available. Independent of IP address.	Cannot capture repeat visitors. Additional overhead for dynamic pages
Registration	User explicitly logs in to the site.	Can track individuals not just browsers	Many users will not register. Not available before registration
Cookies	Save ID on the client machine.	Can track repeat visits from same browser	Can be turned off by users
Software Agents	Program loaded into browser and sends back usage data	Accurate usage data for a single site.	Likely to be rejected by users.

Table 2: various methods to specify the user session

2.2.5 The Effect of Caching

When the first stage of processing is complete, a set of user-sessions representing user click-streams through the site are produced. These, however, may not accurately represent the users' actions as a result of interference from caching.

A cache is a feature provided either by a proxy server or browser-software that keeps a local copy of all web requests passing through it. Thus, if a requested page has been viewed recently, the saved copy can be returned rather than having to make a new request to the web-server. An intermediate cache would likely sit on a local network, prior to the internet gateway and exist to respond to common requests quickly, saving incoming bandwidth for other transactions. A local cache sits within browser software and prevents it from having to re-issue a request when for example the "Back" button is pressed. The page is simply loaded and rendered directly from disk. Whilst offering an efficiency saving in terms of both bandwidth and time, caching has a profound impact on server-side logging- a considerable number of page-views are missing as a result of it.

The simplest solution, used by many authors to avoid complex implementation issues is "Cache Busting", which prevents or minimizes browsers or proxies from serving content from their cache. This forces the user or proxy to fetch a fresh copy for each request. This involves setting the expire time of every page returned to zero (a cache will only hold a local copy until the expire time is exceeded). Whilst a workable and simple solution, it negates the purpose of having caches installed, and may not be suitable.

2.2.6 Robot Removal

WWW Robots are programs that traverse many pages in the World Wide Web by recursively retrieving linked pages (also called wanderers or spiders). These click-streams do not represent the activity of human users and are likely to bias the results of the analysis as a result of their deterministic behavior.

This incident indicated the need for established mechanisms for web servers to stop robots accessed or to remove the robots click-streams before analyzing the navigation behavior, to limit Robots there are two mechanisms [19]:

- The Robots Exclusion Protocol :

  A Web site administrator can indicate which parts of the site should not be visited by a robot, by providing a specially formatted file on their site, in http://.../robots.txt.

- The Robots META tag :

  A Web author can indicate if a page may or may not be indexed, or analyzed for links, through the use of a special HTML META tag.

2.3 Data Analysis

Data analysis of a set of user click-streams involves extracting patterns and rules from the preprocessed data using traditional data-mining techniques (Pattern Discovery), followed by an identification of the subset of "interesting" patterns and rules (Pattern Analysis).The data analysis process usually follows one or more of the following techniques: content-based filtering, collaborative filtering, rule-based filtering and web usage mining mentioned on section 1.3.2 . For the web usage mining we have many mining techniques such as association rules, sequential rules, clustering and classification. In the next sections I will focus on these techniques.

2.3.1 Association Rule mining

It assumes we have some large number of pages (items), e.g., “index.asp”, “students.asp” Visitors navigates some subset of the items, and we get to know what pages visitors visit together, even if we do not know who they are. Designers use this information to improve navigation structure, and control the way a typical visitor traverses the website.

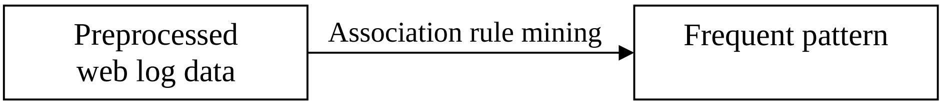


Figure 3: Association rule input/output.

The association rule mining is a technique for finding frequent patterns, associations, and correlations among sets of items [Figure 3] [20]. Association rules are used in order to reveal correlations between pages accessed together during a server session. Such rules indicate the possible relationship between pages that are often viewed together even if they are not directly connected, and can reveal associations between groups of users with specific interests. Aside from being exploited for business applications, such observations also can be used as a guide for Web site restructuring, for example, by adding links that interconnect pages often viewed together, or as a way to improve the system’s performance through pre-fetching Web data.

Association rules are statements of the form {X1,X2,…,Xn}=>Y , meaning that if we find all of X1,X2,…,Xn in the user navigation, then we have a good chance of finding Y .

The probability of finding Y for us to accept this rule is called the confidence of the rule. We normally would search only for rules that had confidence above a certain threshold. We may also ask that the confidence be significantly higher than it would be if items were placed at random into user session. For example, we might find a rule like {index.asp, student.asp} => marks.asp simply because a lot of users visit marks.asp.

I will use the following terminology of association rules in my work [21].

- A *transaction* is a set of items. For example Itemset X = { X1,X2,…,Xn }
- An *association rule* is a rule of the form X => Y, where X and Y are sets of items; X and Y are called respectively the body and the head of the rule. The intended meaning of this rule is that the presence of (all of the items of) X in a transaction implies the presence of (all of the items of) Y in the same transaction with some probability.

Each association rule has two measures relative to a given set of transactions: its confidence and its support.

- *Confidence* is the percentage of transactions that contains Y among transactions that contain X;
- *Support* is the percentage of transactions that contains both X and Y among all transactions in the input data set.

In other words, the confidence of a rule measures the degree of the correlation between itemsets, while the support of a rule measures the significance of the correlation between itemsets.

Association Rule Mining involves two stages.

1. Generating itemsets with the specified minimum support.
2. From each Itemset determining the rules that have the specified minimum confidence.

2.3.2 Sequential Patterns

Sequential pattern discovery [22] [23] is an extension of association rules mining in that it reveals patterns of co-occurrence incorporating the notion of time sequence. In the Web domain such a pattern might be a Web page or a set of pages accessed immediately after another set of pages. Using this approach, useful users' trends can be discovered, and predictions concerning visit patterns can be made.

2.3.3 Clustering

Clustering [23] is used to group together items that have similar characteristics. In the context of Web mining, we can distinguish two cases, user clusters and page clusters. Page clustering identifies groups of pages that seem to be conceptually related according to the users' perception. User clustering results in groups of users that seem to behave similarly when navigating through a Web site. Such knowledge is used in e-commerce in order to perform market segmentation but is also helpful when the objective is to personalize a Web site.

2.3.4 Classification

Classification [24] is a process that maps a data item into one of several predetermined classes. In the Web domain classes usually represent different user profiles and classification is performed using selected features that describe each user's category. The most common classification algorithms are decision trees, naive Bayesian classifier, neural networks, and so on.

2.3.5 Chose the Pattern discovery

One of the main components of a Web personalization system is the usage miner. Web server logs is processed by applying statistical and data mining techniques such as association rules discovery, sequential pattern discovery, clustering and classification which were introduced in the previous sections. Our goals are to enhance website navigation and modify the site structure. The *Sequential pattern discovery* allows Web-based organizations to predict user navigation patterns and helps in targeting advertising aimed at groups of users based on these patterns. By analyzing this information, the Web mining system can determine temporal relationships among data items. Also we can use User clustering which provides groups of users that seem to behave similarly when navigating through a Web site. It is much relevant for Web advertisement purposes. In *Clustering* meaningful clusters of URLs can be created by discovering similar characteristics between them according to users' behaviors. In classification data classified based on the training set and the values in a classifying attribute and uses it in classifying new data [25]. It is used in credit approval applications. The *association rules technique* the main idea is to consider every URL requested by a user in a visit and to discover relationships with a minimum support level between them.

It's possible to apply any of these techniques to achieve the personalization issue; however the association rules technique is the most relevant one to achieve the thesis goals. This work focuses only on studying how to build personalized navigation for website from association rules obtained through data mining on large web log data. The web-log data consists of sequences of URLs requested by different clients bearing different IP addresses. Association rules can find the frequent patterns for the pages visited by users; therefore, it helps designers in restructuring the website. Association rules can be used to decide the next likely web page requests based on significant statistical correlations.

3 Design and Implementation

This chapter focuses on the design and implementation of the product. The design is developed in accordance with aims, and builds upon work referenced in the Literature Review.

The overall structure for the system in actual process is shown in Figure 4: Overall system process. Visitors begin to explore the personalized website while he/she is surfing the web server record each hit in log files. The pattern discovery process will use these files through the database to find the correlation and association between pages that visited. The Database actually reads the raw entries from the log file and saves it, then after the pattern discovery reads this data and performs its process (Association rule technique) it saves the result in the database. The personalized website will retrieve this data and present it to the user as recommendation to the most used pages. On the other hand, the result of pattern discovery assists the designer to take the right decision for improving the structure of the website.

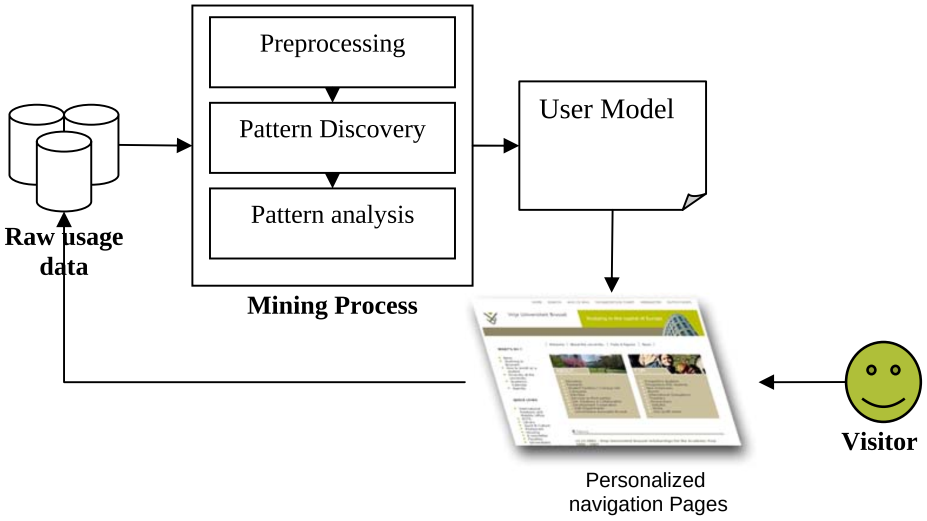


Figure 4: Overall system process.

3.1 General Issues

The next sections describe the design and implementation of the personalized website. The implementation details of implementation language, platform and data storage.

3.1.1 The over all structure of the implementation process

I will use the general architecture for Web usage mining which is presented in Chapter 2. The architecture divides the Web usage mining process into two main parts. The first part includes the domain dependent processes of transforming the Web data into suitable transaction form. This includes *preprocessing* and *transaction identification*. The second part includes the largely domain independent application of generic data mining and pattern matching techniques, the discovery of *association rules* and *statistical counting* method. Data collection is the first step performed in the Web usage mining process then Data cleaning, Data selection, session Identification and repot removal step. Once the domain dependent data transformation phase is completed, the resulting transaction data must be formatted to conform to the data model of the appropriate data mining task. For instance, the format of the data for the association rule discovery task is different from the format necessary for counting technique. These techniques are implemented in a way to take into account the order of elements in a sequence. The counting technique will implement a simple statistical way to counting how many times will each two pages will occurs, while the Apriori algorithm will used to find the relationship between pages in based on the association rule technique. Finally, a query mechanism will allow the user (analyst) to provide more control over the discovery process by specifying various constraints.

3.1.2 Log files source

The Community College (CCAST) web site comprises many pages accessible from the primary domain http://www.ccast.edu.ps . The site consists of only one domain maintained by computer center unit. Like many other education institutes, a significant amount of CCAST's enrollment, teaching, research and administrative functions are now occurring online, it offers training courses for the community, too.

Consider the Web log data from the CCAST Web server shown in Table 3. Typically, these web server logs contain millions of records, where each record refers to a visit by a user to a certain web served by a web server. The log was collected from 00:00:00 April 10, 2006 through 23:59:59 and April 29, 2006. In the last day there were 180802 requests. There are a total of 62 unique IP addresses. A total of 3058 unique pages were requested. These real log files will provide accurate results for the experiment. It has a medium size of records and the web site is educational website with high rate access.

3.1.3 Implementation Language

There are several computer languages suitable for implementing a software system of this type. A broad description of the system operation is that it will have to open a large text file, parse its contents, make some changes, and analyze its contents. C, C++ and Java . High-level languages all heavily used in industry, are all capable of undertaking such a task with considerable evidence of success in the past. Another language family can be considering like ASP, VB Script or Perl. It is interpreted rather then compiled (as C, C++ and Java are), and so can boast a shorter implement! Test! Debug cycle. For instance, Perl is specifically designed for interpreting large volumes of textual data. It is particularly efficient at implementing applications containing complex string manipulation, and handles all memory issues internally (unlike C and C++). As a result of the decision if which process will be online or offline will emphases the chosen language, however my previous experience will also affect the decision, asp with VB scrip are the language of choice for development.

3.1.4 Data Storage

Previous implementations of web usage mining applications have adopted a data storage system using a Database Management System (DBMS). In order to keep the processing time as low as possible, this system is designed to keep part of the data that it requires in volatile-memory, rather than introducing a database support layer with its associated costs, another part I use Microsoft Access as data storage. The daily log files around 20mb and it depends on the usage frequency of the website. As such no problems were anticipated with holding part of data in memory during processing.

3.1.5 Implementation Platform

With the decision having previously been made to implement in asp, the system is inherently platform independent. Asp's native environment can be windows based, and also available for Linux, however, in order to produce fully working installable system, platform specific operations will be required. The development platform chosen is windows as a result of its compatibility with the server systems running the client website and its maturity as a development platform for asp. Little work would be required to port the application to a Linux environment.

3.2 Stage1: Data Preparation

3.2.1 Data Collection

Data that can be used for Web usage mining can be collected at one of these three parts:

- **Server level collection**: the server stores data regarding requests performed by the client, thus data regard generally just one source;
- **Client level collection**: it is the client itself which sends to a repository information regarding the user's behavior (this can be done either with an ad-hoc browsing application or through client-side applications running on standard browsers);
- **Proxy level collection**: information is stored at the proxy side, thus Web data regards several Websites, but only users whose Web clients pass through the proxy.

In my experiment, I cover only the case of a Web server (HTTP server) data.

3.2.2 Data Cleaning

Two filtering methods used. The first is in the form of multimedia file extensions. Another filtering method is in form of certain HTTP error codes at the status field of the entry, in order to remove records corresponding to bad requests, or unauthorized access. This process is to remove noise and inconsistent data.

- Filter Multimedia file extensions, such as .jpg, .jpeg, .avi, etc, and script file extensions, such JavaScript files and Java classes.

By Appling SQL query statements on the log file entries. With more concerned to the URI Stem field (cs-uri-stem) which represents the target of the action, for example, Default.htm, so by applying the query for only the URI stem with wanted extensions like .asp, .htm, .html, .php, etc. After that saves the result of the query statement in MS Access database table for later retrieve. The type of allowed extensions depends on the structure of the web site.

- HTTP error codes at the status field of the entry, in order to remove records corresponding to bad requests, or unauthorized access.

Within the previous SQL query statement. And by using the HTTP Status (sc-status) Which indicate whether a particular request is successful or unsuccessful and can also reveal the exact reason why a request is unsuccessful, for example, 200 means “OK. The client request has succeeded”, and 404.0 means (None) – “File or directory not found”, so by applying the query for only the HTTP Status with appropriate codes like *1xx – Informational*, *2xx – Success* and *3xx – Redirection*

3.2.3 Data Selection

Retrieving the data relevant to the analysis task, more concentrating on these fields:

- Time: The time, at which the activity occurred.

Which will be used to specify the time interval, in order to decide if this page is navigational page or content page. In the experiment I assume that if the time interval is more than 5 seconds and less than 5 minutes it will be content page or else it will be assumed as navigational page.

- Client IP Address (c-ip): The IP address of the client that made the request.

For specifying the user session, In the first experiment I assume that all pages from the same Client IP address will recognize as a user session.

- URI Stem (cs-uri-stem): The target of the action.

It is the main input for the algorithm process, the over all process to discover the usage pattern will depend upon this field.

- User Agent cs(User-Agent): The browser type that the client used.

Will be used to improve the constructing of user session. In order to make the result more accurate.

3.2.4 Session Identification

In data-processing first of all the user session identification process should perform. The module handles the opening of the data-file, traverses it placing the entries into appropriate sessions, performs some simple attribute calculation and returns a data-structure representing the user sessions, their attributes and their raw log entries. Several user session identification ideas were mentioned in section 2.2.4.

For the experiment; User sessions are the main input to the pattern discovery phase, and are extracted using the following procedure:

- Filtered logs are saved in database and can be retrieved by date and time.
- A time-frame is selected within which two hits from the same IP address and user agent can be considered to belong in the same access session.
- Pages accessed by the same IP address and the same user-agent within the selected time-frame are grouped to form a User session.

The user session will be applied for the content pages by using SQL query statement to retrieve the three fields, then applying the process on the URI Stem.

3.2.5 Robot Removal

In the experiment, I asked the Web Server Administrator to use the Robots Exclusion Protocol in order to prevent Robots from access the site.

When a compliant Web Robot visits a site, it first checks for a "/robots.txt" URL on the site. If this URL exists, the Robot parses its contents for directives that instruct the robot not to visit certain parts of the site. So, he put the "/robots.txt" in the top-level of URL space for the selected files date which I used later in the analysis. Simply To exclude all robots from the entire server we put the following code in the robots.txt file:

```
User-agent: *
Disallow: /
```

3.3 Stage2: Pattern Discovery

3.3.1 Data Input

After data preparation stage which transfer data from a web-usage log to a series of individual user sessions. Each session can be saved in array that can be utilized by the Data Mining Algorithms.

For the experiment I used web log files contains 2 days' HTTP requests to the IIS with W3C Extended log file format from Educational web site http://www.ccast.edu.ps (section 3.1.2):

	First Log file	**Second log file**
The log was collected in	April 10, 2006	April 29, 2006
Requests	36864	180802
Unique visiting IP addresses	56	62
Requested pages	2157	3058

Table 3: Details of used log files.

This detail was checked by the analog 6.0 tool.

Here we show a typical web log entry.

```
2006-04-29 00:45:26 83.244.64.38 - 10.10.40.11 80 GET /staff/staffweb/index.asp
id=1055 302 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1)
```

3.3.2 Applying simple counting Technique

For the first trial I proposed a simple counting technique in order to discover the usage pattern as an example, for this user session

{ index.asp, student.asp, marks.asp, index.asp, student.asp, advert.asp, index.asp}

I used two nested loops that will traverse the array and count the frequency for each two pages. By this algorithm I can find the association between every two adjacent links, But this will be so difficult to work if we need to discover the association between three or more links. Also there is no mining flavor because every time it should traverse all links even if it was traversed.

Experimental Results

By using the log files (mentioned in Table 3) after applying pre-processing phase, we have the user session in Table 4. After applying the counting technique we have the result in Table 5 which contains the top two links found together. The Count column specifies how many times that the first and second link was visited together.

User ID	Pages Itemsets
1	{/default.asp, /nportal/default.asp, /elan.htm, /nportal/dologin.asp,..,}
2	{/default.asp, /ceu/default.asp, /default.asp,/English/Default.asp,..,}
3	{/default.asp, /nportal/default.asp, /default.asp, /staff/index.asp,..}
…	…

Table 4: Sample user session

First URI	Second URI	Count
/default.asp	/nportal/default.asp	73
/nportal/default.asp	/nportal/dologin.asp	101
/nportal/dologin.asp	/nportal/INDEX.asp	173
/nportal/INDEX.asp	/default.asp	68
/default.asp	/nportal/INDEX.asp	45
/english/default.asp	/default.asp	5
…	…	…

Table 5: The output generated by the counting technique

The result reveals the cohesion strength between every two pages in the website. Designers can use this information to review the navigation structure while it can be used to suggest the next link for the visitor.

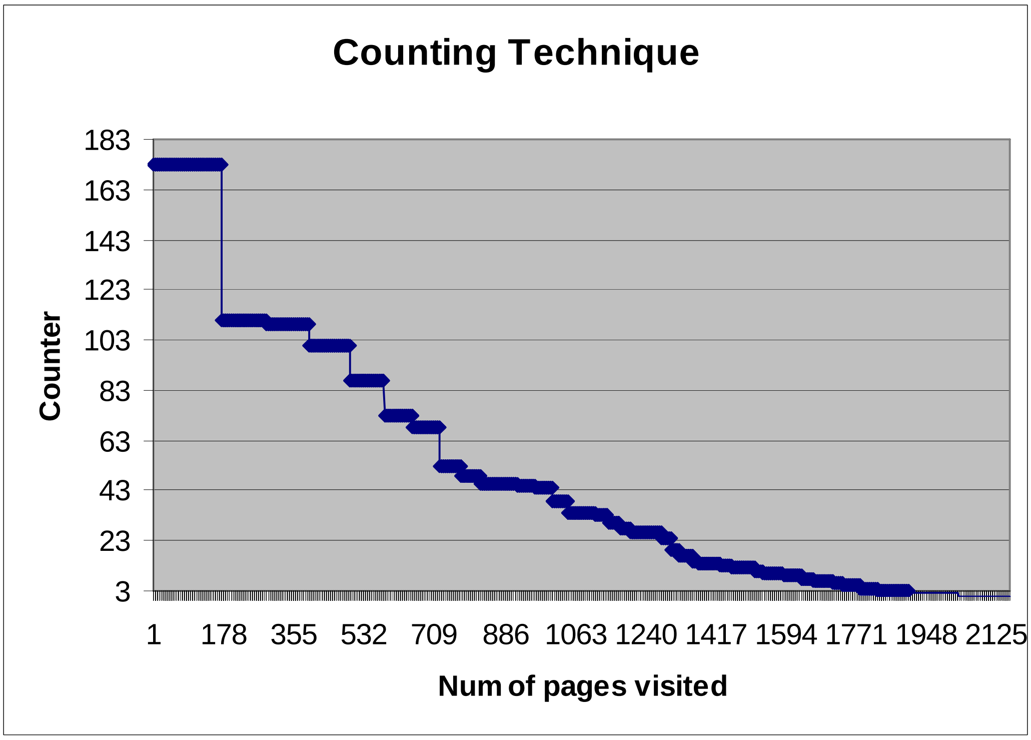


Figure 5: Counter technique results

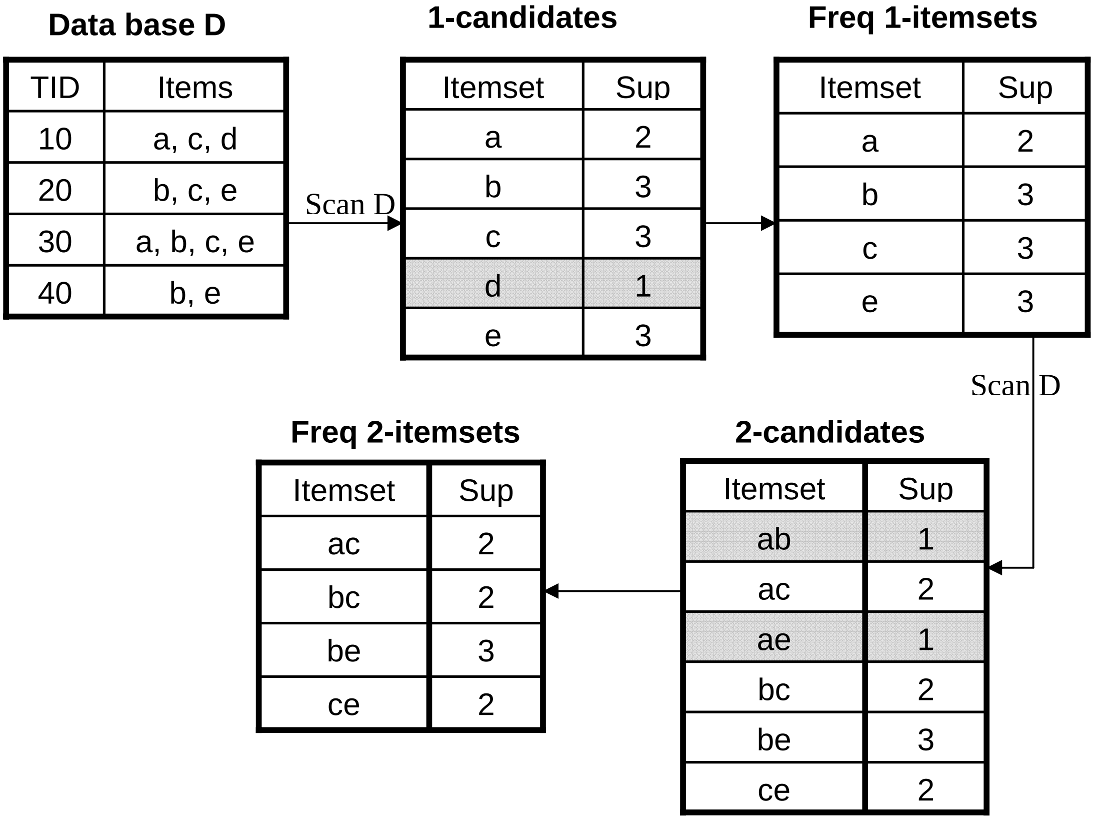
Unfortunately, this technique simply used two nested loops to check the association between every two URI the entire loop will traverse all URIs sequence also the external loop will do the same, therefore to discover the association between four URIs for instance we need four nested loops which will be time and memory consume.

3.3.3 Applying Data usage Mining Technique

Association rules capture the relationships among items based on their patterns of co-occurrence across transactions. In my case, association rules capture relationships among page views based on the navigational patterns of users as mentioned in section 2.3.1.It is common practice to generate only those rules that have high support and confidence. Algorithms that mine for association rules are doing what is conceptually simple, but the clever thing is managing a huge search space to make the mining of interesting association rules tractable. One such algorithm is Apriori which I will use in my work.

The association rules are generated from the itemsets. It is the generation of the itemsets that is the key to the efficient search in Apriori. Most of the time I will only be interested in large itemsets and therefore a minimum support value (minSupport) is specified. All itemsets whose support is below this value can be discarded. Apriori starts by generating the large 1-candidates with a single item then generate Freq 1-itemsets whose support is equal to or higher than minSupport. The 1-itemsets are then used to generate 2-candidates (itemsets with 2 items) which in turn are used to generate candidate 3-itemsets, and so on. The most interesting part of the algorithm concerns the generation of the candidate itemsets [[26]].

For instance if the minSupport = 2



Data base D

TID	Items
10	a, c, d
20	b, c, e
30	a, b, c, e
40	b, e

1-candidates

Itemset	Sup
a	2
b	3
c	3
d	1
e	3

Freq 1-itemsets

Itemset	Sup
a	2
b	3
c	3
e	3

2-candidates

Itemset	Sup
ab	1
ac	2
ae	1
bc	2
be	3
ce	2

Freq 2-itemsets

Itemset	Sup
ac	2
bc	2
be	3
ce	2

Figure 6: Example for Apriori algorithm

The Apriori algorithm as mentioned by Wei Wang.

```
*Ck*: Candidate itemset of size k
*Lk* : frequent itemset of size k

*L1* = {frequent items};
for (k = 1; *Lk* !=∅; k++) do
    *Ck+1* = candidates generated from *Lk*;
    for each transaction t in database do increment the count of all candidates in
    *Ck+1* that are contained in t
    *Lk+1* = candidates in *Ck*+1 with min_support
return ∪*k* *Lk*;
```

For the generation of candidates we have two steps:

- Step 1: self-joining Lk
- Step 2: pruning
    - For each itemset *c* in *Ck* do
    - For each (*k*-*1*)-subsets *s* of *c* do if (*s* is not in *Lk*-*1*) then delete *c* from *Ck*

The Major roles of Apriori algorithm:

- Any subset of a frequent itemset must be also frequent — an anti-monotone property
- No superset of any infrequent itemset should be generated or tested. So many item combinations can be pruned
- Generate length (k+1) candidate itemsets from length k frequent itemsets, and
- Test the candidates against DB

Experimental Results

By using the same log files (mentioned in Table 3) after applying the Apriori algorithm we have the result in Table 6 which contains the first and second Frequent Itemsets with minsupport = 5 . The supp column specifies support of the rule which measures the significance of the correlation between itemsets.

First Freq Itemset		Second Freq Itemset	
1-Itemset	**Supp**	**2-Itemset**	**Supp**
/default.asp	1197	{/default.asp , /nportal/INDEX.asp}	30
/nportal/INDEX.asp	543	{/nportal/INDEX.asp , /default.asp}	30
/nportal/dologin.asp	293	{/default.asp , /nportal/dologin.asp}	26
/staff/staffweb/index.asp	367	{/Default.asp , /ceu/default.asp}	21
/taqarer.asp	8	{/default.asp , /child6/default.asp}	17
/ceu/default.asp	40	{/ceu/default.asp , /staff/staffweb/index.asp}	15
...	...	...	...

Table 6: Frequent Itemsets generated by the Apriori algorithm

From Table 6 we can obtain that 30% of users who accessed the Web page with URL /default.asp also accessed /nportal/INDEX.asp, so these pages are highly tide.

We can choose a rule with the highest confidence among all the applicable association rules, among all rules whose support values are above the minimum support threshold, therefore we can present a recommendation system for users or designer can use this rules to restructure the website.

The mechanism to choose a proper minsupport for a given minconfidence will affect the number of rules, because if minsupport and minconfidence too high we cannot obtain enough rules on the other hand if they are too low the runtime may be unacceptable long.

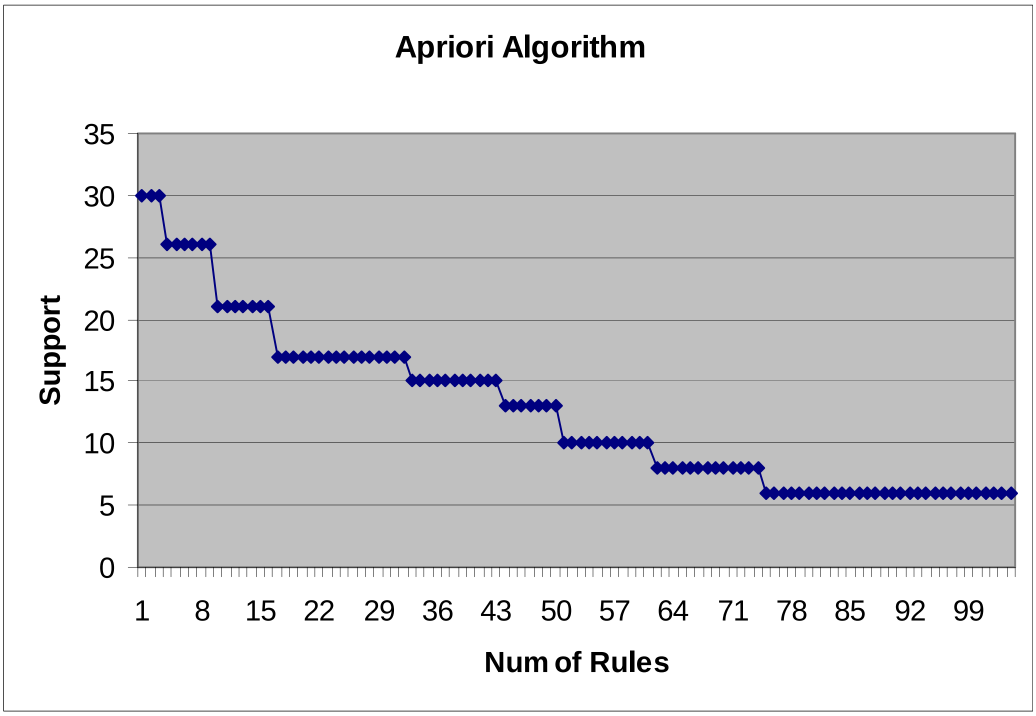


Figure 7: Apriori Algorithm results

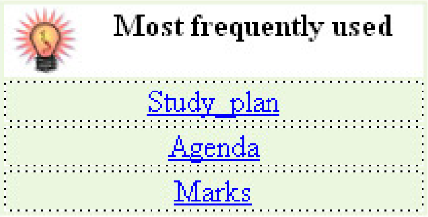
While in Apriori algorithm we can find the association between two or more pages, in the counting technique its time and memory expensive. In addition the mining in Apriori will decrease the number of links that should be traversed, while in counting it should traverse all links in every loop.

3.4 Stage 3: Pattern Analysis and Optimization Transformations

These observations can be used as a guide for Web site restructuring, for example, by adding links that interconnect pages often viewed together, or as a way to improve the system's performance through pre-fetching Web data. As mention in section 1.5 . There are several options to optimize the link structure Promotion & Demotion, Highlighting, Linking and Clustering. Furthermore, we can use Adaptive hypermedia technology [27] which provide direct guidance, adaptive link sorting, link hiding, link annotation, link generation or map adaptation. All of these techniques can be used to realize the pattern discovery results.

The CCAST College - whose log files I take to apply my techniques - maintains a website for its introductory courses, stuff and a portal for student services. Enough information is available, however important documents can be hard to find or entirely lost in the confusion. After applying the personalization technique with Apriori algorithm, the website designers become able to determine what is important and makes that information easier to find. Important pages would be available from the site's front page. Important links would appear at the top of the page or be highlighted. Timely information emphasized, and obsolete information moved out of the way. For instance the technique showing the tightly coupled between */default.asp* and */nportal/INDEX.asp*, in the real situation users should click four links to go from */default.asp* to */nportal/INDEX.asp* which is time-consuming. By these suggestions the website designers decide to put */nportal/INDEX.asp* link directly in the */default.asp* page.

By the recommendation, system users may perceive a connection between sections of the site that the web designer never intended; linking these sections may facilitate user navigation. Users may also find an irrelevant connection the web designer considered important. Therefore the website designers decide to apply the technique with log files collected in April, 2006. Firstly, they concern on the pages belongs to the student portal which contains information for student's registration process, student affair, Bank and finance services etc. As a result they decide to reconstruct the student portal structure based on the association rules they got from the technique. For instance with the new website release they provide a new recommendation system within the student portal. The recommendation system presents for each page tree suggestion links which have the highest support and confidence amount [Figure 8].

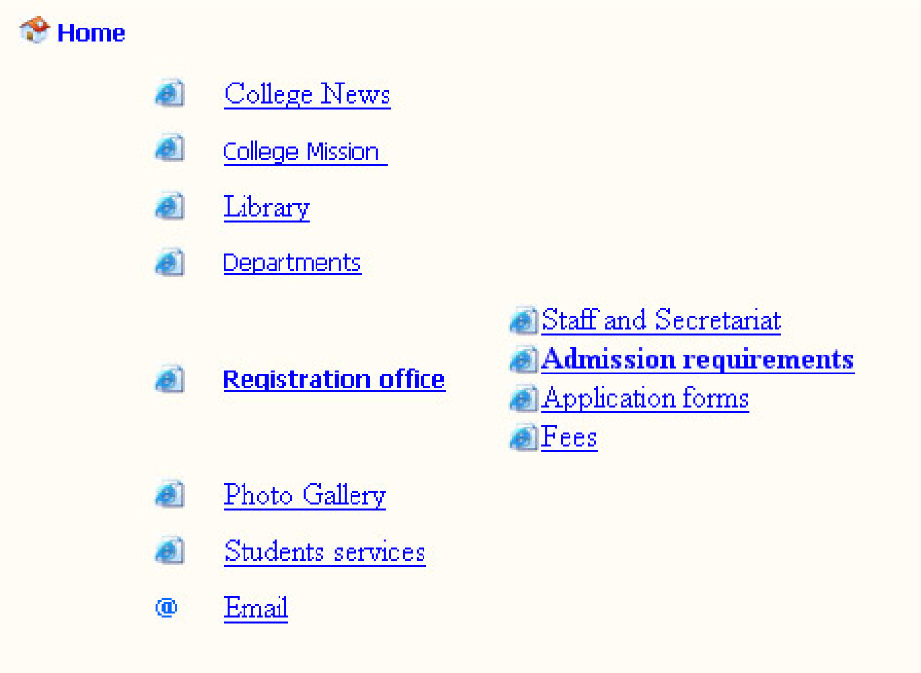


Most frequently used

- Study_plan
- Agenda
- Marks

Figure 8: Suggestion links

Secondly, this technique could be used to improve the website map by highlighting (section 1.5) the frequently accessed links. They decide to make ten links with most support and confidence is bold see [Figure 9].



Home

- College News
- College Mission
- Library
- Departments
- **Registration office**
  - Staff and Secretariat
  - **Admission requirements**
  - Application forms
  - Fees
- Photo Gallery
- Students services
- Email

Figure 9: Part of CCAST site map

Finally, the overall structure of the website was changed by means of adding new links for each page with its most relevant page, removing another link because of its poor coupling with current page.

3.5 Apriori drawbacks

The Apriori technique generates a very large number of frequency itemsets and rules, which reduces not only efficiency but also effectiveness of mining rules to find useful ones. The Apriori heuristic achieves good performance gain by (possibly significantly) reducing the size of candidate sets. However, in situations with prolific frequent patterns, long patterns, or quite low minimum support thresholds, an Apriori algorithm may still suffer from the following two nontrivial costs:

- It is costly to handle a huge number of candidate sets. For instance, with (3000) frequent 1-itemsets, the Apriori algorithm generates more than (4501500) length-2 candidates and accumulate and test their occurrence frequencies. Moreover, to discover a frequent pattern of size 100, such as {X1,X2,…,X_{100}} it must generate more than $(2)^{100}$ candidates in total. This is the inherent cost of candidate generation.
- It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns.

After some careful examination, the bottleneck of the Apriori algorithm is at the candidate set generation and test. If one can avoid generating a huge set of candidates, the mining performance can be substantially improved.

4 Conclusion

4.1 Rationale and Brief

Plainly, personalization has become a required, expected feature of many web-base domains specially an e-learning Website. The presence and quality of site personalization determine whether visitors find the site attractive and return to it with an intention to stay. The real question is not whether to personalize, but how and what, and how to implement personalization for maximum performance. In this thesis, I have tried to study current personalization techniques and the significance of intelligent navigation issue and other techniques to maximize site performance. On other hand I have focused my study in different association-rule based methods for web request prediction. My analysis is based on the whole personalization process from data collection phase to pattern analysis however the pattern discovery was implemented through the simple counting technique and the Apriori Algorithm.

4.2 Application in Related Areas

Recently Web Usage Mining has focused on its use for personalization. There are number of studies that have looked at the construction of operational knowledge, to be used by the personalization module of a Web-based system. Mobasher et al [28] present a system that employs Web Usage Mining techniques to identify users and recommend dynamically and in real time Web pages to them. Perkowitz and Etzioni [29] proposed the idea of adaptive Web sites, i.e., sites that "…automatically improve their organization and presentation by mining visitor access data collected in the Web server logs". The information extracted is used to generate Web pages, based on templates, and present them to the users dynamically. Similarly, Daniel Siaw et al [30] present SiteHelper, which is a local agent, i.e., it operates on a specific Web site. Site-Helper exploits Web Usage Mining techniques to build a set of rules that represent the user's interests. Having discovered these rules the system can recommend new or updated Web pages to the users according to their interests.

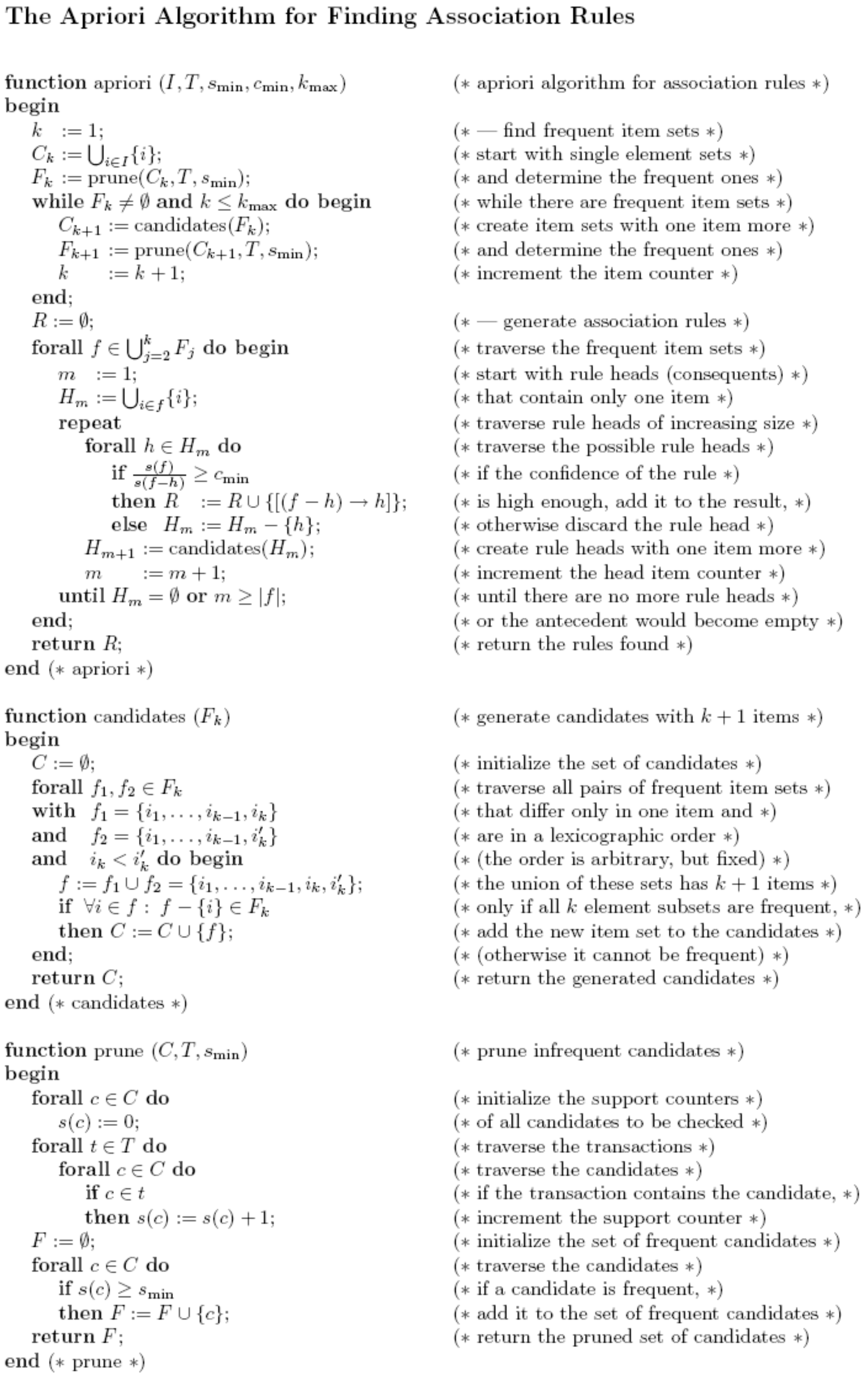
There are some of the most popular Web sites that provide personalized or customized Web pages. Popular Web sites such as Google, Yahoo!, Excite, or Microsoft Network [31] allow users to customize home pages based on their selections of available content, using information supplied by the users and cookies thereafter. Rule-based filtering is used from online retailers such as Dell and Apple Computer, giving users the ability to easily customize product configurations before ordering. As far as recommendation systems are concerned, the most popular example is Amazon.com. The system analyzes past purchases and posts suggestions on the shopper's customized recommendations page.

This emphasizes the important of personalization issue. The methods that are applied for implicit collection of user profile data vary from the use of cookies or similar technologies to the analysis of the users' navigational behavior that can be performed using Web usage mining techniques. However, all the techniques that are used for this purpose have some drawbacks. The users' privacy violation is the most important issue that should be addressed. The extraction of information concerning the navigational behavior of Web site visitors is the objective of Web usage mining.

4.3 Future Work

In the future, I am planning to implement other algorithms for pattern discovery and more rule selection methods which helps in evaluating the results. In addition, I wish to consider other types of domains knowledge to include in the rule representation and examine more sophisticated techniques for data preprocessing and the identification of access sessions in order to reduce common problems of Web Usage Mining.

4.4 Algorithm



The Apriori Algorithm for Finding Association Rules

```
function apriori ($I, T, s_{\min}, c_{\min}, k_{\max}$)      (* apriori algorithm for association rules *)
begin
   $k$   := 1;                                          (* — find frequent item sets *)
   $C_k := \bigcup_{i \in I}\{i\}$;                     (* start with single element sets *)
   $F_k :=$ prune($C_k, T, s_{\min}$);                  (* and determine the frequent ones *)
   while $F_k \neq \emptyset$ and $k \leq k_{\max}$ do begin   (* while there are frequent item sets *)
      $C_{k+1} :=$ candidates($F_k$);                   (* create item sets with one item more *)
      $F_{k+1} :=$ prune($C_{k+1}, T, s_{\min}$);       (* and determine the frequent ones *)
      $k$      $:= k + 1$;                              (* increment the item counter *)
   end;
   $R := \emptyset$;                                    (* — generate association rules *)
   forall $f \in \bigcup_{j=2}^{k} F_j$ do begin        (* traverse the frequent item sets *)
      $m$   := 1;                                       (* start with rule heads (consequents) *)
      $H_m := \bigcup_{i \in f}\{i\}$;                  (* that contain only one item *)
      repeat                                            (* traverse rule heads of increasing size *)
         forall $h \in H_m$ do                          (* traverse the possible rule heads *)
            if $\frac{s(f)}{s(f-h)} \geq c_{\min}$      (* if the confidence of the rule *)
            then $R$   $:= R \cup \{[(f - h) \to h]\}$; (* is high enough, add it to the result, *)
            else  $H_m := H_m - \{h\}$;                 (* otherwise discard the rule head *)
         $H_{m+1} :=$ candidates($H_m$);                (* create rule heads with one item more *)
         $m$      $:= m + 1$;                           (* increment the head item counter *)
      until $H_m = \emptyset$ or $m \geq |f|$;          (* until there are no more rule heads *)
   end;                                                 (* or the antecedent would become empty *)
   return $R$;                                          (* return the rules found *)
end (* apriori *)

function candidates ($F_k$)                             (* generate candidates with $k + 1$ items *)
begin
   $C := \emptyset$;                                    (* initialize the set of candidates *)
   forall $f_1, f_2 \in F_k$                            (* traverse all pairs of frequent item sets *)
   with  $f_1 = \{i_1, \ldots, i_{k-1}, i_k\}$          (* that differ only in one item and *)
   and   $f_2 = \{i_1, \ldots, i_{k-1}, i'_k\}$         (* are in a lexicographic order *)
   and   $i_k < i'_k$ do begin                          (* (the order is arbitrary, but fixed) *)
      $f := f_1 \cup f_2 = \{i_1, \ldots, i_{k-1}, i_k, i'_k\}$;   (* the union of these sets has $k + 1$ items *)
      if $\forall i \in f : f - \{i\} \in F_k$          (* only if all $k$ element subsets are frequent, *)
      then $C := C \cup \{f\}$;                         (* add the new item set to the candidates *)
   end;                                                 (* (otherwise it cannot be frequent) *)
   return $C$;                                          (* return the generated candidates *)
end (* candidates *)

function prune ($C, T, s_{\min}$)                       (* prune infrequent candidates *)
begin
   forall $c \in C$ do                                  (* initialize the support counters *)
      $s(c) := 0$;                                      (* of all candidates to be checked *)
   forall $t \in T$ do                                  (* traverse the transactions *)
      forall $c \in C$ do                               (* traverse the candidates *)
         if $c \in t$                                   (* if the transaction contains the candidate, *)
         then $s(c) := s(c) + 1$;                       (* increment the support counter *)
   $F := \emptyset$;                                    (* initialize the set of frequent candidates *)
   forall $c \in C$ do                                  (* traverse the candidates *)
      if $s(c) \geq s_{\min}$                           (* if a candidate is frequent, *)
      then $F := F \cup \{c\}$;                         (* add it to the set of frequent candidates *)
   return $F$;                                          (* return the pruned set of candidates *)
end (* prune *)
```

5 References

[1] Sean Timberlake, The Basics of Navigation, Website:
http://www.efuse.com/Design/navigation.html

[2] R. Cooley, B. Mobasher and J. Srivastava. Web Mining: Information and Pattern Discovery on the World Wide Web, USA, 1997.

[3] Khan, B. (2001). Managing E-Learning Strategies: Design, Delivery, Implementation and Evaluation, Hershey, PA, USA: Idea Group Inc.

[4] Bonnet, M. (2002). Personalization of web services: opportunities and challenges. Ariadne, Issue 28, June. http://www.ariadne.ac.uk/issue28/personalization/

[5] M. Eirinaki, M. Vazirgianis, "Web Mining for Web Personalization", ACM Transactions on Internet Tehnology (TOIT), volume 3, issue 1, 2003.

[6] Peter Brusilovsky (2003), From Adaptive Hypermedia to the Adaptive Web. USA.

[7] Khan, B. (2001). Managing E-Learning Strategies: Design, Delivery, Implementation and Evaluation, Hershey, PA, USA: Idea Group Inc.

[8] Brusilovsky, P. (2001). Adaptive hypermedia. Methods and techniques of adaptive hypermedia. International Journal of User Modeling and User-Adapted Interaction.

[9] Mike Perkowitz and Oren Etzioni. Adaptive Sites: Automatically Learning from User Access Patterns.

[10] Mike Perkowitz and Oren Etzioni. Adaptive Sites: Automatically Learning from User Access Patterns.

[11] Cooley, R., Srivastava, J., & Mobasher, B.. (1997) Web Mining: Information and Pattern Discovery on the World Wide Web. In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97).

[12] R. Cooley, B. Mobasher and J. Srivastava. Web Mining: Information and Pattern Discovery on the World Wide Web, USA, 1997.

[13] Jose Borges and Mark Levene , Data Mining of User Navigation Patterns, U.K.

[14] Bamshad Mobasher , Web Mining : Information and Pattern Discovery on the World Wide Web Website:
http://maya.cs.depaul.edu/~mobasher/webminer/survey/node2.html

[15] Srivastava, J. Colley, R. Deshpande and Tan ,(2000) Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations.

[16] W3C, Web Characterization Terminology & Definitions Sheet: Website, http://www.w3.org/1999/05/WCA-terms/

[17] Microsoft, The W3C Extended log file format: Website http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/bea506fd-38bc-4850-a4fb-e3a0379d321f.mspx?mfr=true

[18] Gabriele Bartolini.(2001)Web usage mining and discovery of association rules from HTTP servers logs.

[19] TheWeb Robot Database. Website: http://www.robotstxt.org , Jan 2005.

[20] Magdalini Erinak and Michalis Vazirgiannis , Web Mining for Web Personalization.

[21] Weiyang Lin, Sergio A. Alvarez and Carolina Ruiz, Collaborative Recommendation via Adaptive Association Rule Mining.

[22] Rakesh Agrawal Ramakrishnan Srikant. Mining Sequential Patterns, IBM Almaden Research Center.

[23] B. Mobasher. Web Usage Mining and Personalization, In Practical Handbook of Internet Computing Munindar P. Singh (ed.), CRC Press, 2005.

[24] Kostis Sagonas. Data Mining course - lecture note, Uppsala university , 2002.

[25] Wei Wang, lecture slides (spring 2006) Data Mining: Concepts, Algorithms, and Applications, website: http://www.cs.unc.edu/Courses/comp290-090-s06/

[26] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. and Verkamo, A. I. (1995). Fast discovery of association rules, Advances in Knowledge and Discovery and Data Mining. AAAI/MIT Press, Cambridge, MA.

[27] Hongjing Wu, Geert-Jan Houben, Paul De Bra, AHAM: A Reference Model to Support Adaptive Hypermedia Authoring. Netherlands

[28] Mobasher, B., Cooley, R., & Srivastava, J. (1999). Creating Adaptive Web Sites Through Usage-Based Clustering of URLs.

[29] Perkowitz, M. & Etzioni, O. (1998). Adaptive sites: Automatically synthesizing web pages.

[30] Daniel Siaw, Weng Ngu and Xindong Wu. SiteHelper: A Localized Agent that Helps Incremental Exploration of the World Wide Web, Website: http://www.cs.uvm.edu/~xwu/Publication/SiteHelper.html

[31] Websites: Google: www.google.com

Yahoo!: www.yahoo.com
Excite : www.excite.com
MSN : www.msn.com
Dell : www.dell.com
Apple : www.apple.com
Amazon: www.amazon.com

[35] Gabriele Bartolini.(2001)Web usage mining and discovery of association rules from HTTP servers logs.

[36] *Wei Wang*, lecture slides (spring 2006) Data Mining: Concepts, Algorithms, and Applications, website: **http://www.cs.unc.edu/Courses/comp290-090-s06/**