VILNIUS GEDIMINAS TECHNICAL UNIVERSITY
INSTITUTE OF MATHEMATICS AND INFORMATICS

Židrina PABARŠKAITĖ

ENHANCEMENTS OF PRE-PROCESSING, ANALYSIS AND PRESENTATION TECHNIQUES IN WEB LOG MINING

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,
INFORMATICS ENGINEERING (07T)

VGTU
LEIDYKLA TECHNIKA
Vilnius 2009

Doctoral dissertation was prepared at the Institute of Mathematics and Informatics in 2003–2009.

Scientific Supervisor
Prof. habil. dr. Šarūnas RAUDYS (Institute of Mathematics and Informatics, Technological Sciences, Informatics Engineering – 07T).

http://leidykla.vgtu.lt
VGTU leidyklos TECHNIKA 1620-M mokslo literatūros knyga

ISBN 978-9955-28-429-1

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS INSTITUTAS

Židrina PABARŠKAITĖ

ŽINIATINKLIO ĮRAŠŲ GAVYBOS PARUOŠIMO, ANALIZĖS IR REZULTATŲ PATEIKIMO NAUDOTOJUI TOBULINIMAS

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,
INFORMATIKOS INŽINERIJA (07T)

VGTU
LEIDYKLA
Vilnius TECHNIKA 2009

Disertacija rengta 2003–2009 metais Matematikos ir informatikos institute.

Mokslinis vadovas

prof. habil. dr. Šarūnas RAUDYS (Matematikos ir informatikos institutas, technologijos mokslai, informatikos inžinerija – 07T).

Abstract

As Internet is becoming an important part of our life, more attention is paid to the information quality and how it is displayed to the user. The research area of this work is web data analysis and methods how to process this data. This knowledge can be extracted by gathering web servers' data – log files, where all users' navigational patters about browsing are recorded.

The research object of the dissertation is web log data mining process. General topics that are related with this object: web log data preparation methods, data mining algorithms for prediction and classification tasks, web text mining. The key target of the thesis is to develop methods how to improve knowledge discovery steps mining web log data that would reveal new opportunities to the data analyst.

While performing web log analysis, it was discovered that insufficient interest has been paid to web log data cleaning process. By reducing the number of redundant records data mining process becomes much more effective and faster. Therefore a new original cleaning framework was introduced which leaves records that only corresponds to the real user clicks.

People tend to understand technical information more if it is similar to a human language. Therefore it is advantageous to use decision trees for mining web log data, as they generate web usage patterns in the form of rules which are understandable to humans. However, it was discovered that users browsing history length is different, therefore specific data preparation needed in order to compose fixed length data vectors required by the algorithm. Methods what data preparations steps necessary to carry out are provided and later classification and prediction tasks were applied to generate web usage models which then could contribute to the web site refinement.

Finally, it was shown that specific part of the text can be a valuable source of information. This part of the text is extracted from the hyperlink text. Method was suggested and steps provided how to use hyperlink text together with other features. Experiments demonstrated more accurate results defining user behaviour by using text as additional feature. In addition hyperlink text can be used in results presentation step as it represents the actual text that users see when clicking hyperlinks.

The main results of this dissertation were presented in 5 scientific publications: two articles in periodical scientific publications from the Master Journal List of Institute for Scientific Information (*Thomson ISI Web of science)*, one in the referred journal by IOS Press, 2 scientific papers were presented and published in the international referred conferences.

Santrauka

Internetui skverbiantis į mūsų gyvenimą, vis didesnis dėmesys kreipiamas į informacijos pateikimo kokybę, bei į tai, kaip informacija yra pateikta. Disertacijos tyrimų sritis yra žiniatinklio serverių kaupiamų duomenų gavyba bei duomenų pateikimo galutiniam naudotojui gerinimo būdai. Tam reikalingos žinios išgaunamos iš žiniatinklio serverio žurnalo įrašų, kuriuose fiksuojama informacija apie išsiųstus vartotojams žiniatinklio puslapius.

Darbo tyrimų objektas yra žiniatinklio įrašų gavyba, o su šiuo objektu susiję dalykai: žiniatinklio duomenų paruošimo etapų tobulinimas, žiniatinklio tekstų analizė, duomenų analizės algoritmai prognozavimo ir klasifikavimo uždaviniams spręsti. Pagrindinis disertacijos tikslas – perprasti svetainių naudotojų elgesio formas, tiriant žiniatinklio įrašus, tobulinti paruošimo, analizės ir rezultatų interpretavimo etapų metodologijas.

Darbo tyrimai atskleidė naujas žiniatinklio duomenų analizės galimybes. Išsiaiškinta, kad internetinių duomenų – žiniatinklio įrašų švarinimui buvo skirtas nepakankamas dėmesys. Parodyta, kad sumažinus nereikšmingų įrašų kiekį, duomenų analizės procesas tampa efektyvesnis. Todėl buvo sukurtas naujas metodas, kurį pritaikius žinių pateikimas atitinka tikruosius vartotojų maršrutus.

Tyrimo metu nustatyta, kad naudotojų naršymo istorija yra skirtingų ilgių, todėl atlikus specifinį duomenų paruošimą – suformavus fiksuoto ilgio vektorius, tikslinga taikyti iki šiol nenaudotus praktikoje sprendimų medžių algoritmus klasifikavimo ir prognozavimo uždaviniams spręsti. Analizės metu rasti naršymo maršrutai leidžia tobulinti žiniatinklio struktūrą, kad labiau atitiktų naudotojų poreikius.

Pasiūlytas teksto, esančio ant nuorodų, panaudojimas. Parodyta, kad prie lankytojų žiūrėtų puslapių pridėjus ir tekstinę informaciją, esančią ant hipernuorodų, galima pasiekti tikslesnius naudotojo elgesį prognozuojančius rezultatus. Pasiūlytas naršymo rezultatų pavaizdavimo etapo patobulinimas, kuomet panaudojus tekstą, esantį ant nuorodų, rezultatai tyrėjui pateikiami suprantamesne forma.

Tyrimų rezultatai publikuoti 5 moksliniuose leidiniuose: paskelbti 3 straipsniai: du – straipsnių rinkinyje, įtrauktame į Mokslinės informacijos instituto pagrindinį (*Thomson ISI Web of Science*) sąrašą, vienas – recenzuojamajame *IOS Press* leidinyje, du – paskelbti tarptautinėse konferencijose.

Contents

Glossary

Abbreviations

AI – Artificial intelligence;

Browse – Navigate the World Wide Web. Synonyms: cruise, surf;

Browser – A client program for viewing HTML documents sent by a server over an HTTP connection;

CLF – Common Log Format generated by Open Market series, Netscape servers, and the Microsoft Internet Information Server;

Cookie – These are parcels of text sent by a server to a Web client (usually a browser) and then sent back unchanged by the client each time it accesses that server;

DNS – Domain name system;

Domain – Group of computers are united physically (through the network) into some unit called domain name or just domain;

FTP – File transfer protocol;

Gopher – A menu driven document retrieval system, was very popular the appearance of Internet;

Hostname – DNS name of the single computer on the Internet, e. g., www.yahoo.com;

HTML – Hypertext Markup Language, the language used to create web pages;

HTTP– Hypertext Transfer Protocol; the client/server protocol for moving hypertext files on the Internet;

IELC – Intersé extended log count;

IP – Is the abbreviation for Internet Protocol;

IRC – Internet relay chat allows real time text based conferencing over the Internet;

k-NN – *k*–Nearest Neighbour algorithm;

Link – A user selectable hypertext or hypermedia jump point, that, when selected, will jump to another text or multimedia object;

MLP – Multilayer perceptron;

NFS – Network file systems used to share files among different hosts;

Newsgroups Discussion lists;

OLAP – On-line analytic processing language;

SQL – Structural query language;

SLP – Single layer perceptron;

Telnet – Allows to log into a host from a remote location;

TCP – Transmission control protocol sends data as an unstructured stream of bytes;

URL – Uniform resource locator; the standard World Wide Web address format;

Usenet – World-wide distributed discussion system;

WWW – world wide web;

XML – Extensible Markup Language, is the universal format for structured documents and data on the Web.

Introduction

Problem under Investigation

The research area of this work is web data analysis and methods how to process this data.

Topicality of the Research Work

Data mining is a very important part of modern business. Mining large-scale transactional databases is considered to be a very important research subject for its obvious commercial potential (Berry et al. 1997). It is also a major challenge due to its complexity. New techniques have been developed over the last decade to solve classification (Duda et al. 2000), prediction (Haykin 1999) and recommendation (Dai et al. 2000; Krishnapuram et al. 2001), problems in these kinds of databases. To find and evaluate valuable and meaningful patterns requires huge quantities of data. Each business can produce amounts of data just by recording all customers' events into the database and it is called data warehousing. Although large quantities of data are typically generated by dotcom web servers (Chen et al. 1996), (Pitkow 1997), (Lin et al. 1999), (Perkowitz et al. 2000), (Han et al. 2000), (Facca et al. 2005), (Shen et al. 2007), (Lee et al. 2008),

(Maruster et al. 2008), (Markov et al. 2007), (Chen et al. 2008), (Li et al. 2006) , (Zorrilla et al. 2008), (Buzikashvili 2007) by monitoring requested packages going to web site visitors. This information is logged into special purpose – web log files. Web servers around the world generate thousands of giga-bytes of such data every day. According to the Internet Software Consortium, the World Wide Web (WWW) since 1994 has grown from two million servers to more that 110 million in 2001 (Internet_Systems_Consortium). The number of home users has increased from 3 million to more than 89 million for the same period in the US, estimated by another Internet research company - Jupiter MM. It also states that almost 33 million Europeans in December 2001 used the Internet to make their Christmas shopping. Forrester analysts in (Schmitt et al. 1999) reported that 84% of interviewed companies received demand for site data to skyrocket by 2001. The online business retail spending has grown from $20.3 billion to $144 billion by 2003. According to the EIAA Mediascope Europe 2008 study (EIAA_Mediascope), Europeans are deepening their experience of the internet by not only increasingly using it for leisure, but actively enhance and manage their daily lifestyle. 179 million Europeans (60%) are online each week. Over half (55%) of European users are online every single day. Three quarters (75%) of internet users are online during their evenings compared to 67% in 2007. 51% of Europeans use the internet at the weekend, an increase of 13% since 2007. According to the figures 54% have booked more holidays than in previous years. Because of the growing confidence, consumers made a huge number of purchases online in 2008 or 9.2 purchases per person versus 7.7 in 2007. By having these figures, we can conclude that market will successfully withstood only by companies taking significant attention into their web data and making serious analysis, others will be choked off (Ansari et al. 2001). Therefore, with no doubt, web log data is the largest source of information concerning human interaction with the www and continue to grow rapidly. This enables knowledge discovery from web logs to recognise practical users, improve marketing strategies and increase web site's retention etc.

Currently, most web sites are designed without taking into account how web logs can be used to tune and evaluate the usefulness of the web site. The success of the web site cannot be measured only by hits and page views (Schmitt et al. 1999). Usually, the need to know how users behave comes later. Ideally, web site design should enclose techniques, which relate web pages and access activity into standard database format and make data preparation process easier. Unfortunately, web site designers and web log analysers do not usually cooperate. This causes problems such as identification unique user's (Pitkow 1997), construction discrete users sessions and collection essential web pages (Faulstich et al. 1999) for analysis. The result of this is that many web log mining tools have been developed and widely exploited to solve these problems.

However, as will be shown in this research, neither commercial (AccrueHitList), (Angoss), (DataMiningSuite), (MINEit), (net.Analysis), (NetGenesis), (SAS_Webhound), (Webtrends) neither free tools (Analog; WUM; Pitkow et al. 1994a; Han et al. 1997) solve adequately these problems.

Several steps called *knowledge discovery* must be passed through in order to observe patterns from data. These steps are (a) data pre-processing which includes such stages as data cleaning, feature selection, transformation, (b) data analysis and (c) finally results visualisation and examination (detailed scheme of the data mining process is presented in Chapter 1) (Fayyad 1996).

Research Object

This thesis focuses on shortcomings and improvements of *knowledge discovery* steps, examined and evaluated in terms of how adequate they are in handling large data sets gathered from internet resources. Analysis on each of these applications can be broken roughly into three phases.

The first one is data preparation and covers web log data irrelevance problem. A new data filtering methodology demonstrated in this research work which overcomes existing approaches.

The second part pertains with various data analysis stage. Web site visitors' classification and browsing patterns can be revealed by performing specific data preparations.

Third part utilises text for two purposes: firstly text is used mining web usage data; reduced classification error confirmed the correctness of the hypothesis about linguistic interference. Secondly text was used to display results. The proposed and developed method is an attractive way to represent user friendly reports to data analyst.

The Aim of the Work

The aim of this research is to improve the effectiveness of currently available web log mining systems by proposing innovative cleaning, analysis and results presentation methods. The effectiveness of the cleaning method is measured by how accurately new system removes unnecessary records from web log data compared to the other widely used cleaning methods. Advances proposed in the analysis stage are measured by the error rate which calculates how much records are classified correctly. Suggested improvements in results presentation stage are unique and are based on psychological factor that

semantically understandable result presentation method is much user friendly than technical presentation.

Tasks of the Work

In order to obtain those goals, the following tasks have been performed:

1. Provide a comprehensive web log mining literature review (and related to web design fields). Identify web design peculiarities. Systemize which ones influence the types of files collected by web servers.
2. Propose an efficient data cleaning method with minimum information loss required for subsequent data analysis.
3. Overview techniques for mining web log usage data. Provide a practical study of effects and limitations using decision trees for prediction tasks.
4. Develop integrated web log and web content data framework to model various real world situations and provide a study which could lead to a better prediction and classification task quality.
5. Investigate and develop method to extract text from the hypertext and use it for displaying results to the data analyst in a semantically more understandable layout.

Applied Methods

Theoretical and empirical analysis and comparison of known web log mining methods aimed for web data analysis. Also knowledge from data mining methods, text retrieval was used. Cleaning and text retrieval frameworks, specific data preparations for classification and predictions tasks were implemented using C++ programming language. Microsoft SQL Server 2000 was used to store data, to query the database and generate reports.

Scientific Novelty and its Importance

1. Performed systemized review of methods used for web log mining. Investigated existing web log data pre–processing, analysis and results presentation methods. Methods are classified, systemized and referred to a relevant web log mining analysis steps. On the basis of this theoretical investigation, it was established that data pre-processing takes a majority of time in knowledge discovery process and influences analysis stage by

allowing reducing number of records, speed up analysis time and display significant web pages only to the end–user.

2. In the study about different design structures it was showed that the amount of data gathered by web server depends on web pages design. To remove redundant data, new data cleaning method have been introduced. Proposed cleaning framework enables speed up analysis stage and view only actual visitors clicks.
3. It is demonstrated that decision tree approach can be used with reasonable misclassification error for analysing navigational users' patters and generated sequential pages resulting in browsing termination or continuing browsing.
4. Introduced combined approach which takes users browsing history and text appeared on the links for mining web log data. Proposed methodology increased accuracy of various prediction tasks.
5. Cognitive aspects of web designers' and end users' allowed proposing more understandable way for displaying web log mining analysis result.

Practical Value of the Work Results

Since web log mining research started about a decade ago, the area still needs systemization of the methods, deep analysis and critical overview of the algorithms. Therefore collected comprehensive theoretical material of the field can contribute as a guide to web log mining study. Theoretical material and developed models can contribute to the research community and to web site designers for improving web log data mining analysis tools in the following ways:

- by implementing new data cleaning approach into data pre-processing stage,
- by accomplishing specific data preparations and then applying decision trees to generate patters which can be easy interpreted. From those patterns data analyst/designer can determine what improvements are required,
- by implementing new methods to display the outcome of the analysis in a friendly way.

Statements Presented for Defence

- Proposed cleaning technique reduces number of records and makes analysis process faster. Moreover, "link based" technique imitates real

clicks therefore easier to trace visitors navigational patterns in the results examination phase.
- By following specific data preparation steps, it is feasible using decision trees to generate rules which identify directions for web site improvement.
- Experimental evaluation using not only visitors navigational, but textual information as features increase classification accuracy.
- Perception and interpretation of the results becomes clearer and more attractive because they appear as a text, which users see while browsing the actual web site.

Approval of the Work Results

The main results of this dissertation were presented in 5 scientific publications: two articles in periodical scientific publications from the Master Journal List of Institute for Scientific Information (*Thomson ISI Web of science),* one in the referred journal by IOS Press, 2 scientific papers were presented and published in the international referred conferences.

The Scope of the Scientific Work

This doctoral dissertation consists of five chapters, conclusions, references, publications related to the author and 3 appendixes. There are 136 pages including appendixes, 32 figures and 15 tables and 184 bibliographical records.

The organization of the thesis is the following:

Chapter 1 gives an introduction to the data mining and process to extract knowledge from the data, presents background theory of web logs, their definition, taxonomy and peculiarity of the data, lists data sources available for mining the web. This chapter also analyses knowledge discovery steps mining web log data. Theoretical concepts and variety of different techniques are presented which deals with these kinds of problems.

Chapter 2 describes how data transfer protocol works, introduces to the concepts and processes between the web and users who download web documents. Depending on the type of the web site, different files are recorded into the repository where users' accessed pages are recorded. In order for reader to understand the delicacy of web data preparation process and the need a proper cleaning mechanism, all these issues have to be considered.

Chapter 3 presents an implementation of the new technique for the web log data cleaning. The chapter discusses about problems occurred cleaning web log

data. A comparison study and examples of different web mining techniques gives a detail view on this subject.

Chapter 4 presents an approach how to organise specific data preparation in order to use decision trees in web log mining.

Chapter 5 describes how text taken from the hyperlinks can be used for mining web log data and for results presentation purposes.

1

Web Data Mining Analysis

This chapter provides a systemized theoretical survey most of which was published in (Pabarskaite et al. 2007), (Pabarskaite et al. 2002), (Pabarskaite 2003).

1.1. Knowledge Discovery from Huge Databases

The goal of data mining is to allow corporations, banks and retailers to explore large quantities of data generated in their databases during trading or interaction with customers. This data can contain hidden information about customer's private and professional activities, habits, lifestyle, location and etc. Well processed this data can retrieve handy information and bring commercial gain. Analysis of information stored in these databases requires large computational resources as well as efficient analytic methods to facilitate examination and understanding. Data examination requires resources such as statistical analysis, artificial intelligence as well as domain expertise to produce well suited data mining tasks (Berry et al. 1997).

Moreover, real world data requires special pre-processing. A set of necessary actions have to be integrated in order to make sense out of data. This set is called the Knowledge Discovery in Databases (KDD) process (Fayyad 1996).

Following listed steps are provided performing any data mining task (see also Fig 1.1):

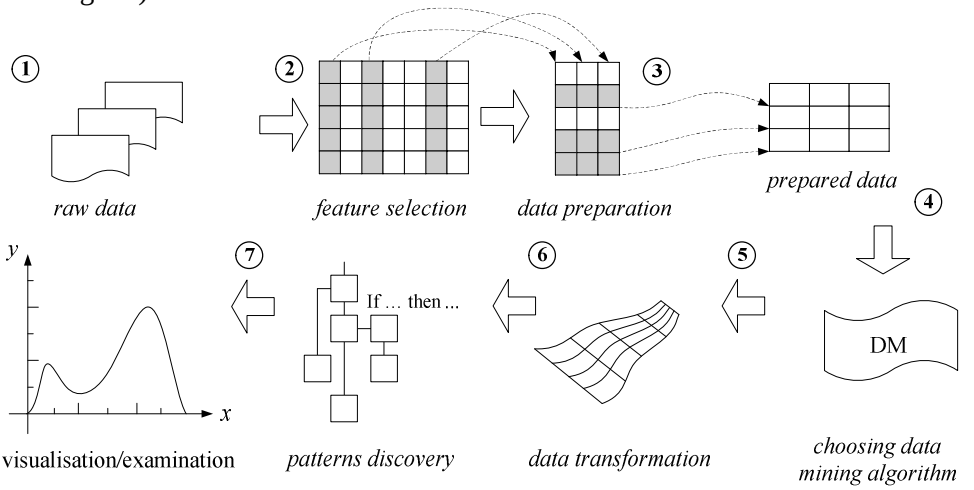

Fig 1.1. KDD steps: gathering raw data, feature selection, data preparation, algorithm selection and transformation processes. The last two stages include running data mining algorithm for pattern discovery and examination of trends

(1) Data collection over a long time period. Large quantities of data are required producing meaningful data mining solutions.

(2) Selecting essential features which define the dimensionality of data.

(3) Data cleaning phase. Removing irrelevant or corrupted records, since this may delay or even crash the analysis process and produce misleading results.

(4) Choosing the proper data mining algorithm according to the task to be undertaken**.** This stage also involves customising and configuring the algorithm according to the task's requirements.

(5) Data transformation involves data preparation what facilitates employment data mining algorithm. For example, data transformation would include grouping data according some criteria, implementing additional features, converting between nominal, binary and continues values according to the requirements.

(6) Hidden pattern extraction. In this stage data mining algorithm extracts patterns embedded in a data.

(7) Data visualisation – results are depicted in a form which can be examined, viewed and is suitable for making the right decisions by humans. **Study** – examination and assimilation of the results while running data mining

algorithm. Originally data mining algorithms generate decisions which are presented in a form of rules, clusters, classes and etc.

After results are examined, some decisions may be accepted and improvements proposed. Therefore **refinement** is the last step, but not a part of KDD process.

Research and development work in data mining over the recent years penetrated into almost every part of human's life covering industrial, business, medicine, biology and other domains. The development of data mining as a new research and application area was driven by the availability of cheap mass storage devices, coupled with highly efficient and relatively inexpensive computer processing power. Moreover, market competition has forced industry and commerce to investigate their data and look for the intelligent solutions to increase business competitiveness. The growing importance of techniques for analysing data can be seen from the growth of data mining companies. For example, some years ago Clementine was bought by SPSS for $7 million, Thinking Machine's Darwin by Oracle for $25 million, HyperParallel by Yahoo for $2.3 million. Recent selling prices of data mining companies are much higher. For example, NeoVista was bought by Accure from $140 million, DataSage by Vignette for $577 million and etc (Ansari et al. 2001). As can be seen, the value of companies extracting knowledge from data growths exponentially.

1.2. Knowledge Discovery from the Web

A decade ago a new research domain based on collecting huge amounts of data from web has appeared which is called web mining. Web mining follows the same knowledge discovery from databases (KDD) process steps as data mining. However, it introduces processes which are unique to this kind of data. Fig 1.2 illustrates the relationship between data mining and mining web data.

With increased competition of retail services, more and more business and market processes have been enabled by the WWW. This can take the form of Business to Customer (B2C) or Business to Business (B2B). Since e-commerce removes all geographical boundaries and time differences, the retailer presents its goods and services across the web. The global consumer browses and makes his decisions. At the same time the growth of web sites from government institutions, libraries and humans increased tremendously over the past few years. The Internet allows two–way communications. This virtual communication between the web sites and users generates huge amounts of Internet data which is either on line, offline or in a form of different electronic form as newsletters and etc.

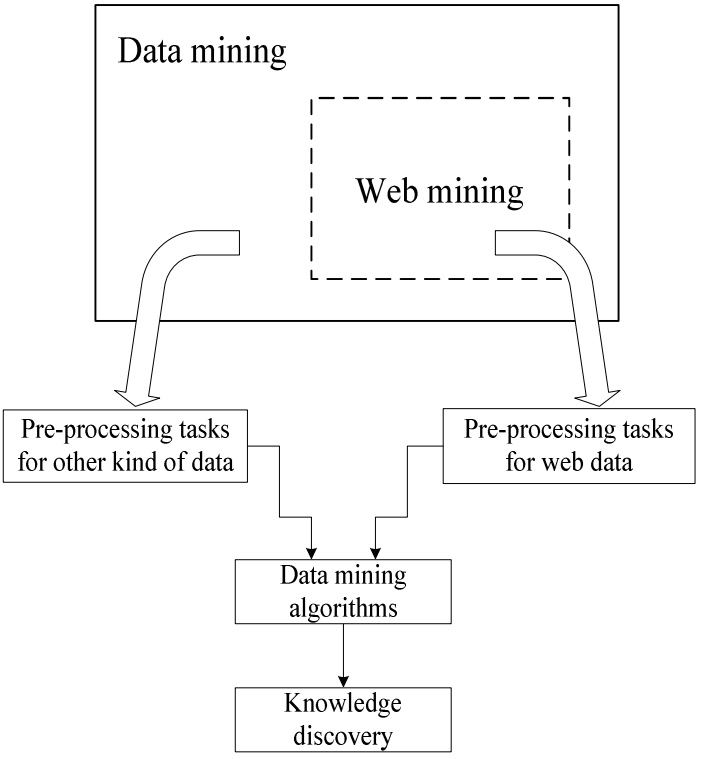

Fig 1.2. Web mining is a part of data mining; however some KDD process steps are distinct

1.3. Web Mining Taxonomy

Most researches (Cooley et al. 1997b; Madria et al. 1999; Srivastava et al. 2000; Pabarskaite et al. 2007) allocate three main directions mining web data (see Fig 1.3). These are:

- *web structure mining*; it uses the topology of the web site structure and examines the strength of associations between pages within and outside the web.
- *web usage mining*; it uses data gathered by web servers as a result of humans interaction with the web site.
- *web contents mining*; it extracts knowledge from text on web pages (unstructured data), HTML documents (semi-structured data), and structured data (XML).

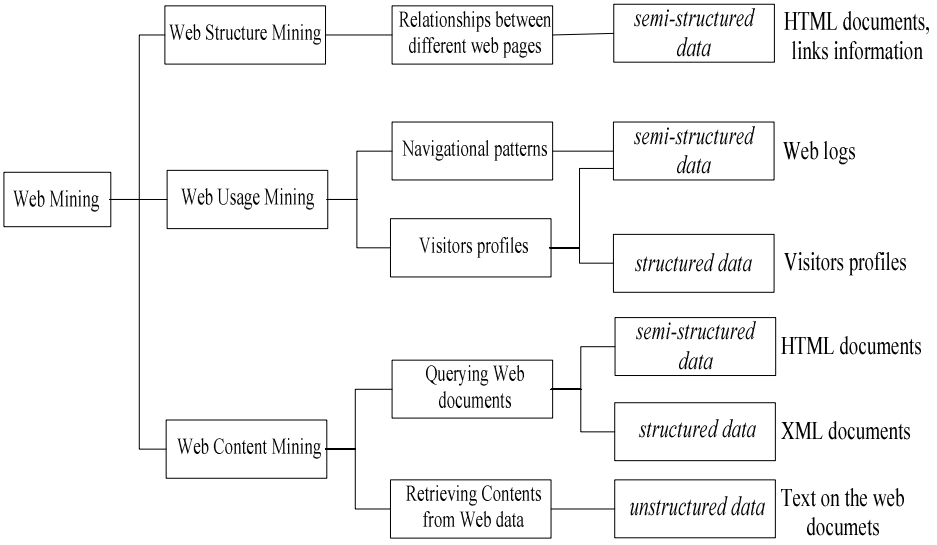

Fig 1.3. Web Mining Taxonomy. Web mining is divided into web structure mining, web usage mining and web content mining depending on the data source used (e. g., web logs, html, xml documents etc.)

1.3.1. Web Structure Mining

Web structure mining is related to the architecture of the web site and examines connectivity of the components within the web site as well as the links within multiple sites in the whole Internet (Madria et al. 1999). The main idea is to assess page quality among others in terms of usability and relevancy. Few examples are presented further which demonstrate advantages gained mining web structure.

(1) One form of web structure mining is measuring the structure of local links existing within the web site. It means, examining how closely related information exist on the same web site. For example, tourism web site will have more local links about available tourist trips.

(2) Measuring how deep links are from the main page. Closer links to the main page have bigger probability to be relevant to the subject. Links of two or three steps from the main page have less probability of being relevant but higher then random selected web pages.

(3) Determining links which bridge different web sites. This analysis involves measuring how easy to find related documents on different web sites. For example, a popular feature on search engines is “more related links”. This

feature shows related web sites having similar information. It ranks pages according to human interest and attention paid to each page. Human interest in a page is measured by number of links pointed to it.

(4) Finally, the web structure mining covers analysing the structure of the web site itself in order to classify or cluster the pages of the site according to the web site structure.

More details on web structure mining can be found in numerous papers (Spertus 1997; Brin et al. 1998; Kleinberg 1998; Madria et al. 1999).

1.3.2. Web Usage Mining

Web usage mining is the most appropriate paradigm to take advantage over navigational patterns that are essential in e-commerce, distance education or just navigating web site (Perkowitz et al. 1999). WWW is made up of pages and links which bridge one page or web site to another. Therefore links represent path from one page to another. The relationship between web pages and links can be illustrated mathematically by a directed graph (see Fig 1.4) where each node represents a web page and edges are links between the web pages.

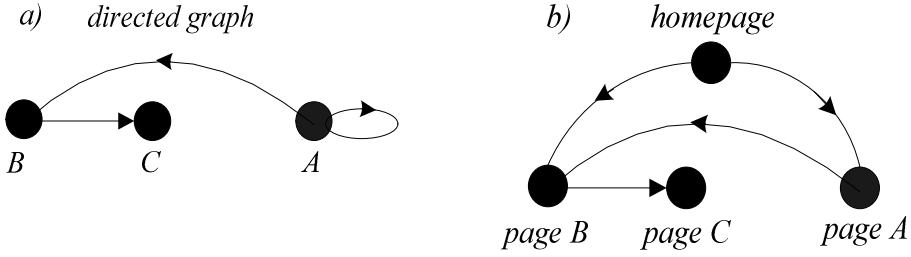

Fig 1.4. a) It is allowed to go through the edges just following directions on the edges as depicted on the picture b) it is available to get from the homepage to pages B and A, but not to C. Page B has a link from page A to page C

According the directed graph, it is not possible to get from one node to another without following direction on the edge (Wilson 1996). Similarly, in the case of web pages, it is not possible to get from one page to another without following links on the web. Customers navigate from one page to another and create stronger or weaker relations between pages. To understand these relations means to understand your customer and his needs. This provides the motivation for web usage mining as becoming a crucial domain for successful e-business.

Data for web usage mining is gathered into web server logs (see section “Web Data” on definition) which register all pages downloaded from the server

to the user computer. Web logs examination outputs include what pages are accessed, in what sequence and combination. Benefits utilizing web logs is the following:

- Discovering associations between related pages, most often accessed together. This can help to place the most valuable information on the frequently accessed pages.
- Discovering sequential arrangements of web pages that are most frequently visited. This can serve for increasing access speed, as related pages can be pre-downloaded in the background. For example, new visitor's browsing is compared with the browsing visitors history in the past and it is assumed that the new visitor behaves similarly or even the same way.
- Clustering users in order to reorganise web site or e-selling according to the patterns of every individual group.
- Classification users into one of the predefined classes. For example, having comprehensive e-trade data, services are proposed to clients whose browsing history matches potential customers. Money and resources are not spent on those who are identified as potential non-customers.
- Personalisation exploits users information (social status, age, earnings, geographical location and etc.) and makes a scenario of valuable visitors profile in order to offer new services depending on who and where they are (Jicheng et al. 1999). Personalisation enables the web site to be adapted better to the market needs and requirements. More issues on personalisation can be found in (Langley 1999).
- Finally, retrieve navigational users information (Masseglia et al. 2002), which sometimes can be combined with other web mining techniques, such as web contents mining (Pabarskaite et al. 2002) or web structure mining (Li et al. 2004) to retrieve interesting relationships between visitors activities and content of web pages.

1.3.3. Web Content Mining

Web content mining retrieves knowledge from the contents of web documents (Boley et al. 1997; Boley et al. 1999; Craven et al. 2000). This allows improving information access on the web.

Information in the web can cover a very wide range of data. At the beginning the Internet consisted of different services and data sources such as ftp, gopher, Usenet. However the volume of the Internet data is raised tremendously in the last decade. Virtually every business or social structure presents itself on the web: digital libraries, government and learning institutions, e-commerce, private homepages for individuals (Kosala et al. 2000). At the same time, this data

became accessible through the web. Visitors, seeking services or products can access all this data through the web interface. Thus web data covers a wide range of data which can be divided into certain groups. The first group is unstructured data such as free texts, e. g. news portals. The second group is semi-structured data such as HTML documents, and the last group is structured data which forms XML.

1.4. Web Data Collection Sources

This section discusses types of web data which can be used for web analysis. Web/Internet data comes from different sources (Mulvenna et al. 1998; Bdchner et al. 1999): marketing, web servers and web site maintainers' resources.

This data can have various structures. For example, marketing data contains information about the clients in relational databases; web server data have three types of data formats: structured, semi-structured and unstructured. Generated log files contain semi-structured data view, some server data can be stored in relational databases (DB) and text from web pages is in unstructured format. Topological information is semi-structured data about the structure (links between pages) of the web site. All these areas (see Fig 1.5) have to synergize together with the data analyst who conducts the pattern discovery process (Bdchner et al. 1999).

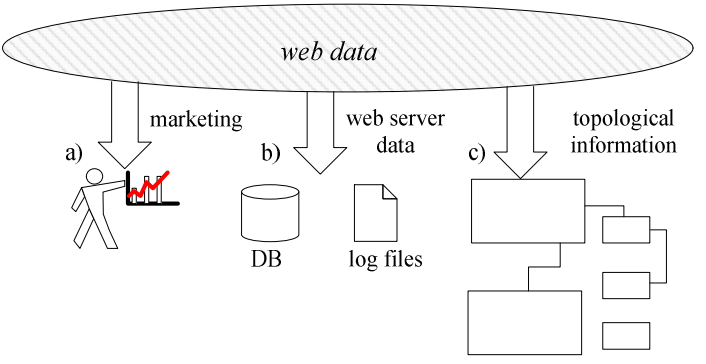

Fig 1.5. Web data collection sources include marketing, web server's data and topological information about the web site

1.4.1. Marketing Data

Data manager collects *marketing data* about selling/navigational results and statistics about visitor's interest in the provided services or goods. This kind of data can also contain knowledge about visitors/clients personal information like profession, social, geographical status and leisure activities and later serve creating valuable customer profiles.

1.4.2. Web Server Data

Web site usage data is collected by web servers and stored in log files. This data is a result of human interactions with the web site. Every human's access to the web site is recorded. Beside common log files, it exists other data storages with valuable information created while the visitor accesses the site (Bdchner et al. 1999). For example, if during the visit, the web page cannot be downloaded, a record is placed into special error log file. In addition to the error files, cookie files are created by web servers and are placed on the client's site. Cookies are files containing information about customers and used for unique users identification purposes. So when the next time user connects to the same web site, web site system recognises the visitor by obtaining and identifying cookie information.

1.4.3. Topological Information

While creating or updating the web site, special data is recorded (Bdchner et al. 1999). This data focuses on logical and physical relationships between the pages outside and associations inside the web site. This data is mostly valuable for administrative purposes since it presents the topology about the web site and other technical characteristics.

1.4.4. Unstructured Web Data

Most of web data is unstructured (Ahonen et al. 1999), (Chakrabarti 2000) and contains documents such as information web sites (e. g. news portals). Information retrieval is performed on vectors constructed from *bag-of-words* (Salton et al. 1983). *Bag-of-words* representation means that all the words from the document are used for analysis. Thus, every document can be represented by a vector $\vec{V}_d = (t_1, w_{d1}; ...; t_i, w_{di}; ...; t_n, w_{dn})$, where t_i is a word or term in the document collection and w_{di} is the weight of t_i in d (document). Sometimes this research area is referred to as text data mining (Hearst 1999) or just text mining

(Soderland 1997; Freitag 1998; Arimura Hiroki et al. 2000; Nahm et al. 2000). The information retrieval process used here is based on linguistic peculiarity of features. There are several important stages in text mining. First step is removing irrelevant tokens such as punctuations or words used very rarely and having no influence on the whole process. Also words having very high occurrence but low degree of importance are removed as well. For example, articles “a” and ”the”, as well as prepositions would be removed (Pabarskaite et al. 2002). Additionally, correlated words can be filtered. This means that the original document word vectors can be transformed to lower dimensional vectors by merging words having the same root (Deerwester et al. 1990; Kosala et al. 2000). For example, words like **techn**ical, **techn**ician, **techn**ocrat and **techn**ologist have the same root “techn”. Therefore, the dimensionality of these words is reduced and is assumed to be “techn”.

Techniques to analyse unstructured data involves tasks such as text categorisation (Tan 1999; Weiss et al. 1999), searching words positions in the document (Ahonen et al. 1998; Frank et al. 1999), looking for morphological roots (Kargupta et al. 1997), phrases (Dumais et al. 1998; Scott et al. 1999), terms as “inland revenue” (Feldman et al. 1998) and hypernyms[1] (OnlineDictionary) (e. g. “car” is a hypernym of the “Volvo” (Riloff 1995)).

Authors in (Jicheng et al. 1999) analyzed relationships between information mining and retrieval on the Web. Authors designed a prototype system WebTMS for mining Web text which is a multi-agent system. It combines text mining and multidimensional document analysis in order to help user in mining HTML documents on the Web effectively. WebTMS analyses documents from different point of views, including the properties of document and the relationships of documents using OLAP technology. OLAP technology enables to apply various analytical operations such as slicing, dicing, rotation, drilling-down and etc.

1.4.5. Semi-Structured Web Data

Research in this area is similar to that in unstructured data. Although due to the additional structural information (since this is HTML data/documents), this type of data has some structure unique for this data type. Since the web is viewed as a large, graph like database, it is natural and possible to run queries for information retrieval. At present querying is supported by all search engines (Florescu et al. 1998). WebLog (Lakshmanan et al. 1996), W3QL (Konopnicki et al. 1995a), WebSQL (Mendelzon et al. 1996), ARANEUS (Mecca et al. 1998), TSIMMIS (Agrawal et al. 1995), Genvl and WWWW (McBryan 1994), Lycos

[1] Linquistic explanation – hypernym is a word that is more generic than a given word.

(Mauldin et al. 1994), Hy+ (Hasan et al. 1995), MultiSurf (Hasan et al. 1995), WebOQL (Arocena et al. 1998), STRUDEL/STRUQL (Fernandez et al. 1997), WebDB (Li et al. 1998) and etc. are systems for querying the contents from the web. These systems retrieve data whose contents meets certain search criteria (Krishnapuram et al. 2001; Lup Low et al. 2001). The syntax and structure of such query languages vary but is similar to SQL since uses the same clauses and structure, e. g.:

```
SELECT word/lists of itmes
FROM document SUCH THAT condition
WHERE condition
```

There are two specialised types of query systems: according to linguistic content of the documents and according to the structure of links (Florescu et al. 1998). At the earlier stages of developing querying web document techniques, searching was performed on the bases of the searched words. This is a linguistic approach. Additionally, extracting more complex information from the web, can involve using sets of tuples in querying systems. Making sense of tuples of words on web documents can be difficult. Therefore, wrapping techniques are utilized (Kushmerick et al. 1997; Gruser et al. 1998; Bright et al. 1999). These wrappers enclose data in a format which can be understood by query systems. Although, querying HTML documents, problems come up because of Internet resources characteristic make changes frequently. Hence, wrapping techniques should be re-designed and maintained according to these changes. In addition, different wrapping techniques should be developed for various types of web sites according to their structure (Florescu et al. 1998). At present, research in this area is trying to integrate more sophisticated queries in order to use the entire keywords or phrases and not just solely words in complex search engines.

In addition, new features were introduced in query processing systems recently that are becoming more advanced. This means that it becomes possible to perform information extraction according to link structures (Carriere et al. 1997; Brin et al. 1998; Kleinberg 1998) which determine how items searched for are related inside and outside web documents.

1.4.6. Structured Data

Structured data have keys (attributes, features) associated with each data item that reflect its content, meaning or usage. A typical example of structured data is a relational table in a database (Markov et al. 2007).

Structured data includes spreadsheets, address books, financial transactions, configuration parameters, drawings and is usually in textual format or binary. Therefore, humans can easily read it without special editors and understanding it does not require expert's participation. The Extensible Markup Language (XML) is a universal format representing structured documents/data on the web. XML is similar to HTML because it also uses tags, although for other purposes. Tags in HTML define how objects should be placed in the document. In XML tags are used for delimiter purposes. XML itself is not a programming language. It contains a set of rules – guidelines on how to design, structure and present data. XML makes computer easy to generate and read structured data.

1.5. KDD Steps Using Web Log Data

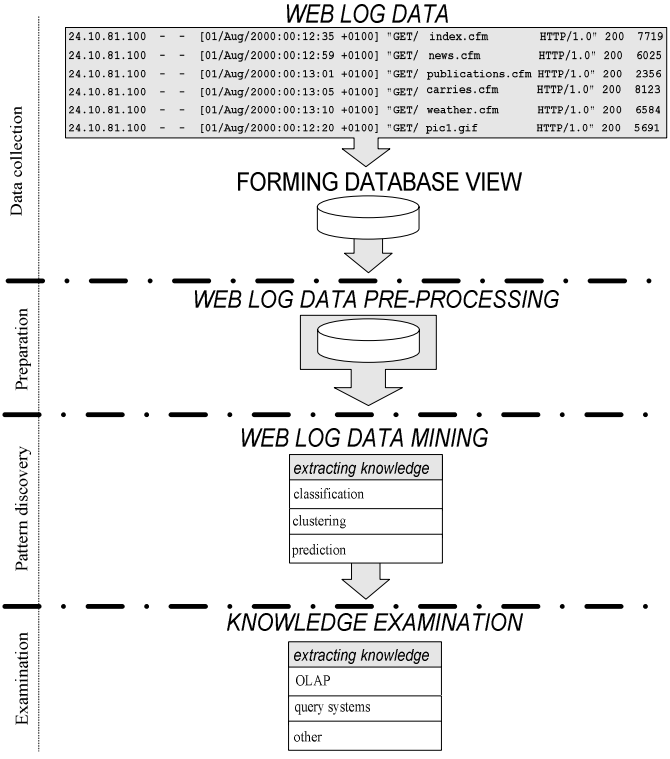

Fig 1.6. Knowledge discovery steps using web log data

Knowledge discovery from databases is applicable to any type of the data (see Fig 1.6), although due to the peculiarity of web log data, some KDD steps are unique. The majority of KDD process time takes data preparation (Pyle 1999) during which meaningful set of features suitable for knowledge extraction is created. However, due to the way web log data is collected, exist many unclear issues like what records are relevant, how unique user must be defined, how users episodes called sessions must be identified and etc.

Thus, theoretical concepts and variety of different techniques are presented in this chapter which deals with these kinds of problems. Finally, web usage analysis and visualisation/examination techniques are presented with a brief references to the other research works.

1.6. Web Log Data Collection

Data gathered from web servers is placed into special files called logs and can be used for web usage mining. Usually this data is called web log data as all visitors activities are logged into this file (Kosala et al. 2000). In real life web log files are huge source of information. For example, the web log file generated running online information site[2] produces log with the size of 30 – 40 MB in one month time, another advertising company[3] collects the file of the size of 6 MB during one day.

There are many commercial web log analysis tools (Angoss; Clementine; MINEit; NetGenesis). Most of them focus on statistical information such as the largest number of users per time period, business type of users visiting the web site (.edu, .com) or geographical location (.uk, .be), pages popularity by calculating number of times they have been visited and etc. However, statistics without describing relationships between visited pages consequently leave much valuable information undiscovered (Pitkow et al. 1994a; Cooley et al. 1997a). This lack of depth of analytic scope has stimulated web log research area expand to an individual research field beneficial and vital to e-business components.

The main goals which might be achieved mining web log data are the following:

- Web log examination enables to restructure the web site to let clients access the desired pages with the minimum delay. The problem of how to identify usable structures on the WWW related with understanding what

[2] http://www.munichfound.de/
[3] http://www.ipa.co.uk

facilities are available for dealing with this problem and how to utilize them (Pirolli et al. 1996).

- Web log inspection allows improving navigation. This can manifest itself by organizing important information into the right places, managing links to other pages in the correct sequence, pre-loading frequently used pages.
- Attracting more advertisement capital by placing adverts into the most frequently accessed pages.
- Interesting patterns of customer behaviour can be identified. For example, valuable information can be gained by discovering the most popular paths to the specific web pages and paths users take upon leaving these pages. These findings can be used effectively for redesigning the web site to better channel users to specific web pages.
- Turning non-customers into customers increasing the profit (Faulstich et al. 1999). Analysis should be provided on both groups: customers and non-customers in order to identify characteristic patterns. Such findings would help to review customers' habits and help site maintainers to incorporate these observed patterns into the site architecture and thereby assist turning the non-customer into a customer.
- From empirical findings (Tauscher et al. 1997) is observed that people tend to revisit pages just visited and access only a few pages frequently. Humans browse in small clusters of related pages and generate only short sequences of repeated URLs. This shows that there is no need to increase number of information pages on the web site. More important is to concentrate to the efficiency of the material placed and accessibility of these clusters of pages.

General benefits obtained from analysing Web logs are allocating resources more efficiently, finding new growth opportunities, improving marketing campaigns, new product planning, increasing customer retention, discovering cross selling opportunities and better forecasting.

The common log format

Various web servers generate different formatted logs: CERF Net, Cisco PIX, Gauntlet, IIS standard/Extended, NCSA Common/Combined, Netscape Flexible, Open Market Extended, Raptor Eagle. Nevertheless, the most common log format is common log format (CLF) and appears exactly as follows (see Fig 1.7):

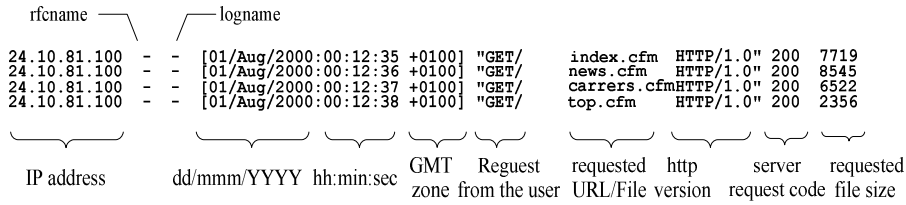

Fig 1.7. Example of the Common Log Format: IP address, authentication (rfcname and logname), date, time, GTM zone, request method, page name, HTTP version, status of the page retrieving process and number of bytes transferred

```
host/ip rfcname logname [DD/MMM/YYYY:HH:MM:SS -
0000]"METHOD /PATH HTTP/1.0" code bytes
```

- host/ip - is visitor's hostname or IP address.
- rfcname - returns user's authentication. Operates by looking up specific TCP/IP connections and returns the user name of the process owning the connection. If no value is present, a "-" is assumed.
- logname - using local authentication and registration, the user's log name will appear; otherwise, if no value is present, a "-" is assumed.
- DD/MMM/YYYY:HH:MM:SS - 0000 this part defines date consisted of the day (DD), month (MMM), years (YYYY). Time stamp is defined by hours (HH), minutes (MM), seconds (SS). Since web sites can be retrieved any time of the day and server logs user's time, the last symbol stands for the difference from Greenwich Mean Time (for example, Pacific Standard Time is -0800).
- method - methods found in log files are PUT, GET, POST, HEAD (Savola et al. 1996). PUT allows user to transfer/send a file to the web server. By default, PUT is used by web site maintainers having administrator's privileges. For example, this method allows uploading files through the given form on the web. Access others then site maintaining is forbidden. GET transfers the whole content of the web document to the user. POST sends information to the web server that a new object is created and linked. The content of the new object is enclosed as the body of the request and is transferred to the user. Post information usually goes as an input to the Common Gateway Interface (CGI) programs. HEAD demonstrates the header of the "page body". Usually it is used to check the availability of the page.
- path stands for the path and files retrieved from the web server.
- HTTP/1.0 defines the version of the protocol used by the user to retrieve information from the web server.

- code identifies the success status. For example, 200 means that the file has been retrieved successfully, 404 - the file was not found, 304 - the file was reloaded from cache, 204 indicates that upload was completed normally and etc.
- bytes number of bytes transferred from the web server to another machine.

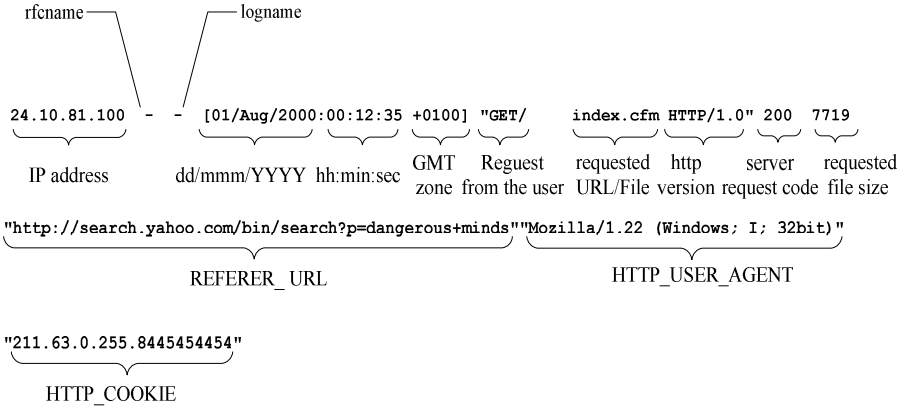

Fig 1.8. CLF followed by additional data fields: web page name visitor gets from - referrer page, browser type and cookie information

It is possible to adjust web server's options to collect additional information such as REFERER_URL, HTTP_USER_AGENT and HTTP_COOKIE (Fleishman 1996). REFERER_URL defines URL names where from visitors came. HTTP_USER_AGENT identifies browser's version the visitors use. HTTP_COOKIE variable is a persistent token which defines visitors identification number during browsing sessions. Then CLF is a form depicted in Fig 1.8.

1.7. Web Log Data Pre-Processing Steps

Web log data pre-processing step is a complex process. It can take up to 80% of the total KDD time (Ansari et al. 2001) and consists of stages presented in Fig 1.9. The aim of data pre-processing is to select essential features, clean data from irrelevant records and finally transform raw data into sessions. The latter step is unique, since session creation is appropriate just for web log datasets and involves additional work caused by user identification problem and various non-settled standpoints how sessions must be identified.

All these stages will be analysed in more detail in order to understand why pre-processing plays an important role in KDD process mining complex web log data.

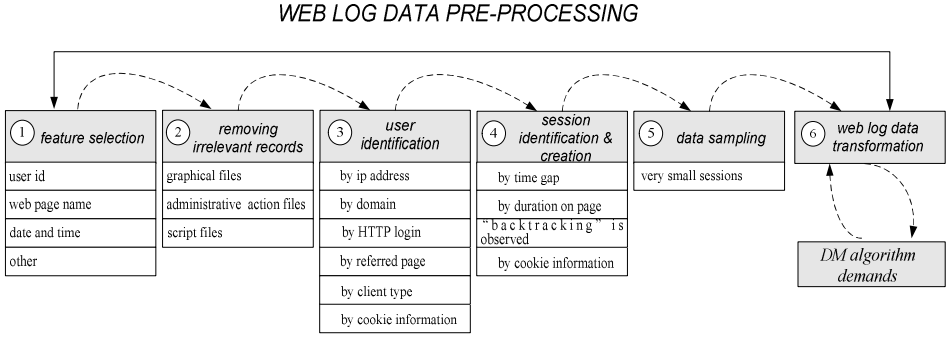

Fig 1.9. Pre-processing web log data is one of the most complex part's in KDD process. Web log data preparation is an extremely complicated process since it requires additional knowledge and understanding of the field

1.7.1. Feature Selection

Log files usually contain number of nonessential information from the analytics point of view. Thus, the first data pre-processing step is feature selection. Moreover, reducing number of features at this stage decreases hard disc space capacity as well. It is beneficial, since log files contain thousands of giga bytes of data. The final output of the pre-processing must be a set of so called sessions. The most important attributes to build episodes are computer IP address, time of the request and accessed page. Looking for web usage patterns basically these three features need to be selected. Other features are not so important unless they participate in some additional pre-processing tasks. For example, the status of the retrieved page or number of bytes downloaded accessing the page are usually not taken into account and utilised for statistical purposes, for example measuring the network loading. However, for example, HTTP_USER_AGENT and HTTP_COOKIE are used by some heuristics identifying unique users. In that case these features are used for creation sessions but not included into sessions themselves.

1.7.2. Data Cleaning

Many web log records are irrelevant and therefore require cleaning because they do not refer to pages visitors click. Most of the research in this area is confined to removing graphics (images, sound, video) files (Han et al. 1998;

Cooley et al. 1999; Faulstich et al. 1999; Cooley et al. 2000). More on this topic see Chapter 3 “Link Based Cleaning Framework”.

1.7.3. Unique User Identification

The next important step is unique user identification (Ivancsy et al. 2007). This problem arises because local proxy and cache systems do not allow easy user identification. Most web usage mining applications do not deal with this problem seriously. They automatically utilise IP addresses as visitors machine authentication and do not take into consideration the origin of the IP address. A proxy/cache server is a system that sits between an application (such as Internet Explorer, Netscape Communicator or similar) and a web server. It intercepts the request to the server to see if it can substitute it. This improves performance by reducing the number of requests that go out to the Internet. The proxy server can cache files it downloads from the Internet. Requested web pages are stored in a cache. When a previously accessed page is requested again, the proxy can retrieve it from the cache rather than from the original web server. If someone else asks for the same page, the proxy server can send back the page it is holding in its cache rather than sending the request out to the web server. Therefore, response time is increased and Internet network traffic is not overloaded.

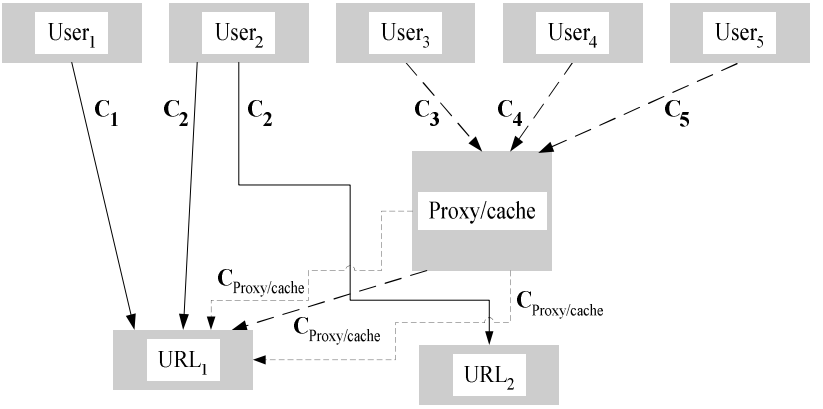

Fig 1.10. Communication between users and requested pages with and without proxy/cache server

A deficiency in exploiting proxy servers is that they do not permit all users to be identified. When a request to the web server is submitted and several users access the Internet through the same cache/proxy server, the web server logs only one user (web server will see only one the proxy IP address). In practise, more

than one user could be making these requests. See (Cooley et al. 1999) for more details on this subject.

The proxy effect on the web server is illustrated in Fig 1.10. Different users are defined as $User_1$, $User_2$, $User_3$, $User_4$ and $User_5$. C_1, C_2, C_3, C_4, C_5 indicate connections/requests and URL_1, URL_2 are web pages. If $User_1$ or $User_2$ requests pages then web server logs two users ($User_1$ and $User_2$). Different is when $User_3$, $User_4$ or $User_5$ (connections C_3, C_4, C_5) request the page URL_1. Then server identifies three connections with the same IP/host (user identification) and assumes that all three calls made by the same user. Nevertheless, different users may be requested page URL_1. In a way similar to proxy servers, cache existence does not allow all users to be identified when the request of the page is repeated. Cache keeps all previously requested and downloaded files in a special repository. So, whenever a user accesses previously requested pages, the browser takes them from the local cache repository and web server of the original page location does not log any action.

1.7.3.1. User Identification by the Domain

Sometimes domain name can help to detect if a visitor is connecting through a proxy/cache server (Pabarskaite 2003). For example, if a connection to the Internet is done through a modem, then domain definition can contain the name such as “dialup”. If a user access Internet through the proxy/cache server then domain definition may contain words as “proxy” or “cache” (e. g. spider-th031.proxy.aol.com or cache-dr05.proxy.aol.com). Removing records containing “proxy” or “cache” words in the domain allows filtering uncertain visitors. A limitation of this approach is that performing filtering of uncertain visitors increases the risk that some important patterns made by proxy/cache users will remain undiscovered.

1.7.3.2. User Identification by the http Login

The ideal way to identify users would be by using registration information entered by users themselves. Fields in the log file *rfcname* and *logname* return the name of the connection owner and visitors name respectively. Although this login process is not popular because humans usually try to avoid web sites where registration information is required unless, it is a special purpose site such as on-line banking. Then login is inevitable.

1.7.3.3. User Identification by the Client Type

Most of literature sources do not investigate proxy names but use some heuristics to identify unique visitors with the same IP address. Extended CLF file may contain additional client information – agent field HTTP_USER_AGENT which shows browser's name and operating system. Thus, one approach to identify a unique visitor is to trace agent and look at the changes in browser and operating system (Pirolli et al. 1996). If two log entries with the same IP have different agent information, an assumption is made that two users exist. An example of such situation is presented below:

```
24.10.81.100 - - [01/Aug/2000:00:12:35 +0100]"GET/ index.cfm HTTP/1.0" 200  7719 "http://
search.yahoo.com/bin/search?p=dangerous+minds""Mozilla/1.22 (Windows; I; 32bit)"

24.10.81.100 - - [01/Aug/2000:00:12:35 +0100] "GET/news.cfm HTTP/1.0" 200
7719 "/index.cfm""Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+Q312461)"
```

Though this approach justifies itself, sometimes it can produce misleading results. A theoretical chance may appear that the same visitor uses different browsers or even different machines and explores the same web site at the same time. One visitor's browsing history will be assumed to represent the activities of two persons. The opposite can also occur as well. If, for example, two visitors with the same IP address (they are connected to Internet through the proxy server) use the same kind of browser and operating system and navigate the same web site, they can be assumed as one user and hence, the analysis will be inaccurate.

1.7.3.4. User Identification by the Referral Log

The approach used here is this. Field REFERER_URL together with the access log and site topology are used to construct browsing paths for each user, see (Cooley et al. 1999). If from the set of pages a new page appears which is not accessible from the previously viewed pages, a new user is assumed. Another condition by which a new user is assumed is when in a path of previously viewed pages there appears a page already navigated. This circumstance is very limited and not accurate. It does not accept repeated pages in the same user's session what is very common in real life.

1.7.3.5. User Identification by the Cookie

The most straightforward way to identify visitors is by using cookie information. When the first time user visits the web site, cookie does not exist

yet. Therefore no information is send from the browser to the web server. Hence, the web server understands that it is a new user and passes cookie information to the browser together with the requested file. After that, web browser places cookie into a special directory where cookie files are kept. Usually cookie files contain strings which define session id, accessed web site and etc. Next time visitor access the same web site, browser sends already existed cookie to the web server (if it is long lasting cookie). Web server recognises the cookie and thus do not send it again but passes just the requested page.

The importance of cookies has received attention from the European Parliament (Roberts 2002). It voted to allow companies to use cookies to study online visitor behaviour. However, because of the availability of methods to control cookies existence on the computer, they are often disabled by the users.

1.7.4. Unique Session Identification

Sessions (in some literature sources are referred as episodes) creation should fulfil two requirements in order to use them as data vectors in various classification, prediction, grouping into clusters and other tasks: (1) all entries belonging to the same person must be grouped therefore mechanism to identify distinct users is necessary; (2) all entries done by the same user must go consecutively and distinct visits to the web site should be distinguished. In other words, a real session is the sequence of activities made by one user during one visit to the site (Berendt et al. 2002), (Huang et al. 2004). General session format contains IP address, page name and time as parameters.

Let's bring some conventions. One session S is a set of entries s made by the user while observing some particular web site. Then session model can be interpreted as:

$$S = \left\langle s.ip, \left\{ \left(s.wp_1, s.t_1 \right), ..., \left(s.wp_n, s.t_n \right) \right\} \right\rangle, \tag{1.1}$$

where $s \in S$ is one visitor's entry which contains $s.ip$ (IP address), $s.wp$ (web page) and $s.t$ (time of the entry), n is number of transactions in a session.

The analysis does not require knowledge who is the user but it is important to distinct every user from the rest. The task is not so simple due to the web sites constructions and the way information about the requests is collected. Besides, there is no single solution how users should be identified. Methodology applied to one type is not suitable for another web log data type. This is because exist different formats for handling web logs. Some web servers collect cookie information, some not. Some web sites require user authentication, others no. Therefore, different approaches for session identification exist: based using time

as a measure and based on reoccurrence of pages viewed by one visitor. Authors in (Spiliopoulou et al. 2003) declare two strategies “proactive” and “reactive” how “real sessions” are identified. “Proactive” constructs sessions using sessions id information gathered from cookies. The process is performed before or during the individual’s interaction with the web. “Proactive” strategies include user authentication, the activation of cookies that are installed on users machine. However, cookies raise number of other issues, firstly, related to the privacy among users. Secondly, still not all web site developers understand the importance of collecting and storing complete web log information and cookies become not available. Therefore the second strategy called “reactive”. Here sessions are created of post users interaction with the web information, which is not always complete. Therefore a huge number of research works is done in this field because there are problems how identify users and automatically sessions as well. Basically, “proactive” strategy outperforms “reactives” in quality of identifying user sessions. The detailed survey of these techniques is described below.

1.7.4.1. Session Identification by the Time Gap

The most popular session identification technique uses time gap between entries. If the time gap between two pages requests made by the same user exceeds some threshold, a new session is created:

$$s.t_{n+1} - s.t_n \geq time_{threshold} \,. \tag{1.2}$$

Different threshold values can be found in research literature and vary from 25 – 30 minutes (Pitkow et al. 1994c) to one or even two hours (Montgomery et al. 2001). The most popular time gap calculated empirically by Catledge and Pitkow is 25.5 minutes (Catledge et al. 1995). The authors calculated mean inactivity time within a site. The typical value was found to be 9.3 minutes, 1.5 standard deviations were added to this typical inactivity time. Hence, 25.5 minutes cut off for the duration of a visit was defined as standard inactivity time. Most commercial and free available applications use the rounded 30 minutes time gap.

Picture, see Fig 1.11, presents example of findings which have been captured from the analysis of one user’s browsing history (data taken from advertising brokering company IPA located in UK). The picture above shows the distribution of accessed pages by the particular user during a 49 day period.

This visitor probably enters the site in his office during working days and the information is probably related to his/her professional needs as the site is entered frequently – several times per day. The dotted lines represent the visitor’s

behaviour during one day. This period is depicted in the next figure. On the less detailed picture (the picture above), it was possible to identify just one activity peak. In the zoomed picture (the picture below) more than one peak can be detected. The expanded area represents visitor's activity over one day (7.6 hours). Different peaks represent separate visitor browsing periods. However, some of them belong to the same group of transactions - sessions. If the time gap between two peaks is less than a defined threshold, this activity is assumed to be in one session.

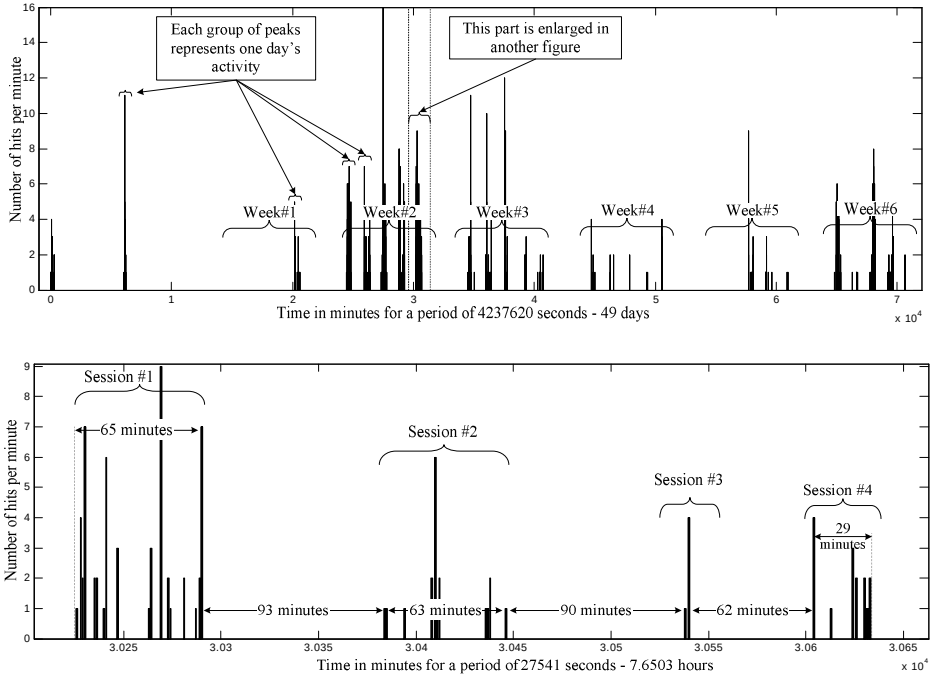

Fig 1.11. Visitors web page access distribution: picture above – over 49 days period, picture below – enlarged picture of the over 1 day period

1.7.4.2. Session Identification by the Maximum Forward References

Another way to identify sessions is using maximal forward references presented in (Chen et al. 1996). This approach converts the original sequence of log data into a path from the first to the last page before the result of button "back" action is processed. In other words, if in the sequence of not repeated pages (called forward reference) appears the page already found in a set (called

backward reference), a new session is defined starting from the next page. The last page before the backward reference page called maximum forward reference. However this method doesn't seem justifies itself since button "back" in practice is pressed very frequently and actually is a part of the common browsing activity.

1.7.4.3. Session Identification by the Duration Spend Observing the Page

Another session identification approach is proposed in (Cooley et al. 1999). This approach is based on the assumption that depending on the duration time spent observing the page, pages can be divided into two groups – navigational (or auxiliary) and information (or content) pages. Information pages are the desired visitor destinations and the duration time on these pages is much longer than navigational pages which are passed through. Respectively, the duration time of navigational pages is smaller. The next important step in this approach is to define the threshold between the duration time of navigational and information pages. Let's assume that the percentage of the navigational pages is known in the log file. Then, the maximum length (if the length is bigger, then it is information page) of navigational pages is calculated as:

$$threshold_{navigational\_pages} = \frac{-\ln(1-\gamma)}{\lambda}. \qquad (1.3)$$

This formula is derived from the exponential distribution of navigational and information pages (see (Cooley et al. 1999) for distribution graphic), γ is the percentage of navigational pages (it is assumed that is known) and λ is the observed duration time mean of all pages in the log. Based on this study 10 minutes time maximum page stay time is agreed (Faulstich et al. 1998; Cooley et al. 1999).

1.7.5. Removing Small Items

Working with the real world data it is very common that some records are not interesting for the analysis. Examples of small items can be pages which occur one or two times in the analysed dataset. Sometimes users just get into the web site and leave. One entry is logged and session with the length of one page is created. However such length of sessions cannot construct meaningful cluster of visited pages and do not bring knowledge about web site usage. Operator who performs data pre-processing tasks should, in fact, refer to the tasks requirements and filter items which are too small.

1.7.6. Data Transformation

Raw data is not the most convenient form and it would be advantageous to modify prior to analysis (Hand et al. 2001). The last data pre-processing step is data transformation, A.K.A. data formatting (Cooley et al. 1999). This process involves final preparation data for the analysis (mining). This may include:

- Final selection necessary fields. For example, time stamp is not required for association rules mining. In that case time stamp is removed from data.
- Creating additional and/or combined features. For example, it may be worth to have the feature like length of spend time to examine the correlation between the pages and staying duration on them.
- Preparation related to data mining model. For example, some models (e. g., neural networks) require transforming data into vectors with binary attributes. Then input vector is replaced with “nulls” and “ones“. Other techniques accepts semi-binary inputs like “nulls”, “ones”, “twos” and etc. indicating number of times page occurred in a session.

1.8. Analysis Steps and Knowledge Extraction

The history and the origin of data mining techniques are rather different from the disciplines where they are used today (Berry et al. 1997). Some algorithms came from biology such as neural networks and genetic algorithms. Links analysis arose from the graph theory in mathematics and etc. Basic notes describing algorithms used mining web log data are illustrated in forthcoming sections.

1.8.1. Clustering

Clustering assumes consolidating perspective customers/visitors containing similar behaviour and characteristics naturally into groups (Jain et al. 1997; Xiao et al. 2001; Kanth Ravi et al. 2002). These groups of related clients are called clusters. For example, a cluster is found of the most valuable customers. It contains attributes of clients having income more than £25000 per year and age between 30 and 45. Another example of the cluster is web pages occurrences. For example, pages A, B, C are the most visited from Mondays to Fridays.

The similarity between cluster objects (data samples) can be measured utilising Euclidean distance (Duda et al. 2000). Let say there are n objects in a data and p measurements on each object. The vector of observations for the i^{th}

object is $x(i) = (x_1(i), x_2(i),\ldots,x_k(i),\ldots, x_p(i))$, $1 \le i \le n$, $1 \le k \le p$ where the value of the k^{th} variable for the i^{th} object is $x_k(i)$. Then Euclidean distance between the i and $(i+1)$ objects is the difference between the corresponding elements and then square them.

$$d_E(i,i+1) = \sqrt{\sum_{i=1}^{p} \left(x_k(i) - x_k(i+1)\right)^2} \,. \tag{1.4}$$

Clustering is suitable for huge datasets and is the first data mining/web mining technique to examine the structure and patterns in data (Hand et al. 2001).

Dai et al. put forward techniques based on clustering users transactions to discover overlapping profiles which are later used by recommendation systems for real-time personalization (Dai et al. 2000). Personalisation means web site's adoption to individual user needs. Usually personalisation withdraws historical data about visitors experience and refines the web site according user's taste.

Cooley, Mobasher and Srivastava (Cooley et al. 2000) used a full spectrum of data mining algorithms for web personalization, based on transaction clustering, usage clustering, and association rule discovery. Their proposed approach web personalization is based on examining past users activities. This information later is used for online recommendations.

1.8.2. Association Rules

Due to the size of transactions in real world datasets and number of items in every transaction association rules are becoming more and more popular in modern transactional data analysis because they are able to discover related items occurring together in the same transaction (Kanth Ravi et al. 2002). Therefore are very useful in web log mining and have been applied in some research works (Kato et al. 2001). In practice it is almost impossible to test all possible combinations of customers purchasing, as this number is huge. Therefore, association rule algorithms significantly reduce the number of combinations to examine. One of the most popular association rule algorithm's is Apriori, introduced by authors in (Agrawal et al. 1993; Agrawal et al. 1994; Agrawal et al. 1995). The output of this algorithm produces human understandable rules of the form $X \Rightarrow Y$, where X and Y are sets of items.

Apriori uses a lot of a priori information to reduce number of combinations to be examined. The algorithm consists of two phrases. First, frequent (sometimes called large) itemsets are generated. Next, these itemsets are used to generate rules. Frequent itemsets as well as rules are output of majority association rule algorithms. They both are useful and important. However,

Apriori algorithm produces large number of rules, most of which are obvious or irrelevant (Padmanabhan et al. 1999). Thus another aim of rule induction is to interpret and understand rules. Before generating itemsets, one must determine - the size and quality of the rules. No one is interested in rules that occurred only few times. Therefore statistical measures as confidence and support for estimation usefulness of the rules are taken as selection criteria. Minimum support (size of the rule and itemset) and minimum confidence are the measures that allow controlling quality of extracted rules. These measures allow the algorithm to skip not-frequent and high-error itemsets and rules.

Thought, in order to generate large itemsets, items in each transaction are sorted. This allows reducing confusion between differently ordered itemsets. Next, all 2-sized itemsets (all combinations of single items) are examined. Ones what hold minimum supports are preserved. Later, 2-itemsets are used to generate 3-itemsets. The rule "if itemset hold, all subitemsets must also hold" are used here. This significantly increases performance of the algorithm. Further, the procedure is repeated in the similar way. 3-sized itemsets are used to generate 4-sized ones and so on. The algorithm stops when no new items are generated.

In the following phrase, obtained itemsets are used to generate rules. This is a straightforward process. From each itemset, all subitemsets are generated. This stands for IF part. The rest items in the itemset stand for THEN part. The generated if/else rule is tested against minimum confidence. If rule holds, it is preserved as a result.

As a matter of fact, some authors contradict such approach collecting rules (Piatetsky-Shapiro et al. 1994; Silberschatz et al. 1995; Padmanabhan et al. 2000). They argument that subjective measures are unexpectedness and actionability and finding interesting patterns depends on the decision maker and not solely on the statistical measure (Adomavicius 1997). The algorithm of discovering unexpected rules is presented in (Padmanabhan et al. 1999).

The most applicable area applying association rules is retail sector where sets of together purchased items are computed. However association rules are successfully used in web usage mining as well. Since one session corresponds to one transaction in retail sector.

Example: Denote S_n as a session (in retail section it would be one transaction), where *n* is a session identification number. Let say there are number of sessions made by the same or distinct user:

S_1= ("Products.html", "Software.html"),
S_2 = ("Products.html", "Software.html", "Hardware.html", "Services.html"),
S_3 = ("Software.html", "Hardware.html", "Services.html"),
S_4 = ("Products.html", "Software.html", "Hardware.html", "Contacts.html").

The first session S_1 includes two visited pages “Products.html” and “Software.html”, the second contains four pages and etc. From there following rule “IF Software.html and Hardware.html THEN Services.html” can be derived with the support s = 2/4 = 50% (number of such transactions occurred across all transactions) and confidence c = 2/3=75% (transactions on the first part of the rule supporting the rule across transactions on the first part which do not necessary support the rule).

Association rules algorithm Apriori have been widely used in web usage mining (Jain et al. 1997; Cooley et al. 1997a; Cooley et al. 1997b; Cooley et al. 1999; Cooley et al. 2000; Fong et al. 2000). However, Apriori does not always detect items having low support (frequency of certain web pages occurred together in one session) but are nevertheless interesting from the user’s point of view. Solution for this problem is to implement support constrains which specify what should be minimum support for what itemset (Han et al. 2000).

Authors in (Yang et al. 2002) propose to extend traditional association rule paradigm by imposing new temporal information and the confidence of each rule related to the prediction if certain page in a certain time moment will occur.

1.8.3. Sequential Rules

Another type of valuable rules can be exploited by discovering sequential patterns. Ones that find set of items followed consecutively in the time ordered set of transactions (Agrawal et al. 1995; Mannila et al. 1995; Pirjo 2000). In web usage mining transactions correspond to sessions and computed patterns identify characteristic sequential browsing paths.

Definition: Let say *I* be a set of items. An item is a pair (*i, t*), where $i \in I$ is type of the item and $t \in T$ is the time stamp of the event. Then, the itemset *S* is an ordered sequence of items, e. g.:

$$S = \langle (i_1, t_1), (i_2, t_2), ..., (i_k, t_k), ..., (i_m, t_m) \rangle, \tag{1.5}$$

where $i_k \in I$ for all *k = 1,…, m* and $t_k \leq t_{k+1}$ for all *k = 1,…, m-1*. The length of the sequence *S* is denoted as $|S| = m$.

The sequence consisted of itemset *S* in temporal order and defined as:

$$S = \langle (i_1, i_2, ..., i_k, ..., i_m) \rangle, \tag{1.6}$$

where $i_k \in I$ for all *k = 1,…, m* and is called an item sequence. An empty sequence of items is denoted as $S = \langle\ \rangle$.

The problem of mining sequential patterns is to find the maximal frequent sequences among all sequences that have a certain user specified minimum support (see for details (Srikant et al. 1996)).

Definition: The support is defined as the percentage of patterns among all data containing the sequence *S*. Each maximal sequence represents a sequential pattern if $supp(S) \geq minsupp$, when *minsupp* is defined by the operator and *S* is considered as a frequent sequential pattern.

Example. Requested www pages may represent sequential pattern. For example, 45% of users accessed page /Jobs_Online where followed by the page /About_company and /Careers.

Similar inconveniencies appeared utilising sequences of frequent patterns. Standard sequence miners discover only frequent sequences. This limits ability to detect rare but still key or for some point valuable web usage patterns. Spiliopoulou in (Spiliopoulou 1999) propose to extend sequence mining not just discovering frequent item sets but also by discovering "interesting" results applying PostMine algorithm. This algorithm transforms frequent sequences into a set of sequences rules and then filters this set using various statistical measures and heuristics.

Since many network traffic problems occur, web logs are also used for deploying intelligent web caching. If next user step can be predicted with the reasonable accuracy, the cache can contain predicted page and download it for user avoiding network usage (Bonchi et al. 2001).

Authors in (Mobasher et al. 2002) based on experiments on real world data made an assumption that sequential patterns are more effective for prediction tasks. And prediction information is used for perfecting web pages and thus speeding up the browsing process. However not sequential patterns are more suitable for web personalization tasks.

Framework using both, association rules and sequential patterns, for data analysis is presented in (Jain et al. 1997).

The problem with the big web sites is that they produce number of models of possible combinations of users choices. In order to reduce the number of such cases, Pitkow and Pirolli (Pitkow et al. 1999) utilized longest repeating subsequence models without losing the ability to make accurate predictions of the next user step with Markov models.

1.8.4. Other Adaptive Tools for Web Log Mining

Interesting findings were discovered after web usage survey (Pitkow et al. 1995). It was built on the special author's designed web site architecture and available to users on the web. In order to collect information about the browsing peculiarity research was based on the adaptive questions and answers. Certain

standpoints about web users were done and later used as an example about Internet users habits.

Other researchers propose a powerful Web mining tool WUM (Faulstich et al. 1999; Faulstich et al. 1999). Authors state that all items in user's path definition are equally important to discover exact user behaviour. It means that repeated pages in the same session are also essential and are analysed dissimilarly to authors in (Cooley et al. 1997b). WUM has implemented specific query language MINT (Faulstich et al. 1998) which supports predicates of the web pages and their occurrences. Navigational patterns are also supported by miner WUM. Later, in (Berendt et al. 2000), authors focused on the tool which is able to cope with the dynamic pages as well.

In (Han et al. 2000) Han et al. developed an efficient data structure web access pattern tree algorithm WAP to retrieve useful information from pieces of logs. Algorithms such like Apriori counted some difficulty when the length of the pattern grows long what is happening with Web log data. Therefore authors developed an algorithm for efficient mining of such huge sets of data with a lot of patterns. Algorithm is organised in a way that it scans database twice. First time it determines set of frequent events with some defined threshold. Second time system builds a WAP tree data structure using frequent events. Then WAP system mines the WAP tree using conditional searches.

A systematic study about development of data warehousing is presented in (Han et al. 1998) with the tool WebLogMiner (Han et al. 1997). To examine web log transaction's patterns, authors applied on-line analytic processing language (OLAP). OLAP allows characterize and examine data in the web log, view associations, predict values of the attributes, construct the models for each given class based upon the features of the web log data, produce time-series analysis. OLAP provides business analyst with a million spreadsheets at a time available in a logical and hierarchical structure. The analysis can go higher or deeper levels to look at the data from different perspectives (Peterson et al. 2000). Scientist encourage using OLAP as an applicable for analysis and visualisation tool mining web logs (Dyreson 1997).

Authors in (Balabanovic et al. 1995; Cooley et al. 2000) presented a feedback system which adapts and produces better pages for the following day. The system learns behaviour from users. Number of data mining techniques was implemented in the proposed system: clustering, frequent items and association rules. Recommendation engine computes a set of recommendation for the current session consisting of pages that user might want to visit because it is based on similar user patterns.

Model that takes both the travelling and purchasing patterns of customers into consideration was described in (Yun et al. 2000) and (Yun et al. 2000;

Pabarskaite 2003). Developed algorithms extracted meaningful web transaction records and determined large transaction patterns.

To predict future request using path that contain the ordered list of URLs accessed by the users within specified time constraint is described in (Schechter et al. 1998). The given methodology predicted pages request with the high level of accuracy what reduces time servers spend on generating the page.

Process of web log caching is a software integrated into caching/proxy server (Pitkow et al. 1994b). The algorithm implemented into this software is based on psychological research on human memory retrieval. It collects past access patterns and predicts further user actions. Authors also noticed that recent rates for document access are stronger indicators for future document requests than frequency indicator. Web log caching have been implemented by number of other researchers works (Glassman 1994; Luotonen et al. 1994).

Perkowitz and Etzioni (Perkowitz et al. 1997a; Perkowitz et al. 1997b) challenge scientists to concentrate on creation adaptive Web sites using modern artificial intelligent (AI) techniques. It means, that web sites must automatically expand their administration and management by learning from users access patterns. To achieve this, site developers should focus on the customization (modify web pages to suit users needs) and optimisation (to make navigation of the site easier). In (Perkowitz et al. 1999) and (Perkowitz et al. 1998) Perkowitz and Etzioni described algorithm PageGather which uses new cluster mining technique for indicating URLs sets for adaptive Web sites. This semi-automatic algorithm improves the organization and presentation of the web site by learning from visitors behaviour.

The system, which is based on gathering usage patterns on every day's data collection, is presented in (Balabanovic et al. 1995). Firstly, data is collected and analysed. Secondly, the system called LIRA is able to make recommendations for the following day. Recommendations consist of selection documents which system thinks users will find interesting. Accordingly selected pages produced much better results than random selected pages.

Author in (Sarukkai 2000) suggests for link prediction and path analysis to use Markov chains. Since Markov chain model consist of a matrix of state probabilities, Markov chains allow the system to model dynamically URL access patterns.

How to reduce main memory requirement analysing web logs is shown in (Xiao et al. 2001). Authors introduced the problem which appears mining traversal patterns having duplicates. An effective suffix tree algorithm is presented which compress and prunes database and reduces main memory usage.

1.9. Web Mining Results Visualisation

It is very common that various data analysis techniques produce extensive reports which are not always interpreted in a right way by data analysts and cannot be used efficiently (Chi 2002). Humans are very good at identifying patterns from visualized data (Ansari et al. 2001), (Colin 2004). Therefore different results visualization interpretations are used to present web mining results.

(1) Charts for two dimensional plots. This graphical representation is well interpreted by humans and suitable to show statistics. Example in the picture, see Fig 1.12 shows page names and frequency of visits. Limitation of two dimension plots that it shows only limited association between variables.

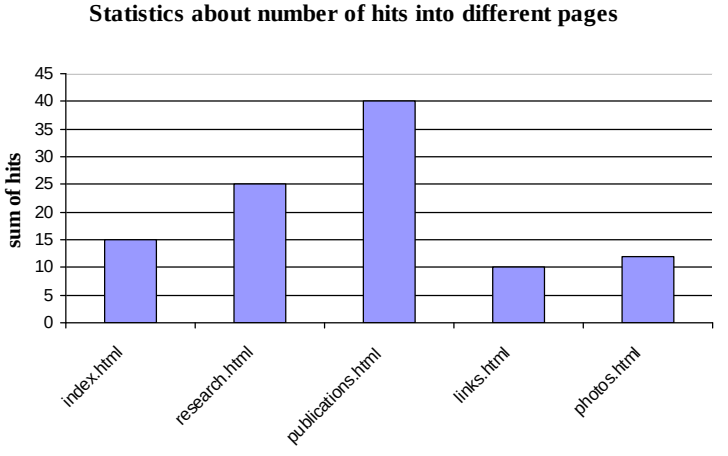

Fig 1.12. Visualisation with BAR graph

(2) Another way for representing web usage results is using graphs. Using graphs, pages can be represented as nodes and links as edges (Pitkow et al. 1994a; Dyreson 1997). Representation through nodes and edges allows exploring web site's hierarchical structure and identify typical usage paths. Such graph implementation allows observing graphical information about accessed documents and the URL paths through the site, enables selectively filter users in the access logs (restrict view by features such as domain names or DSN numbers, directory names, time), control graphs attributes (node size, colour and etc.) and events in the access log by playing back possibility, choose a layout of nodes and

links that best presents the database structure. An example of graph based web usage visualisation technique Hy+ displayed in Fig 1.13.

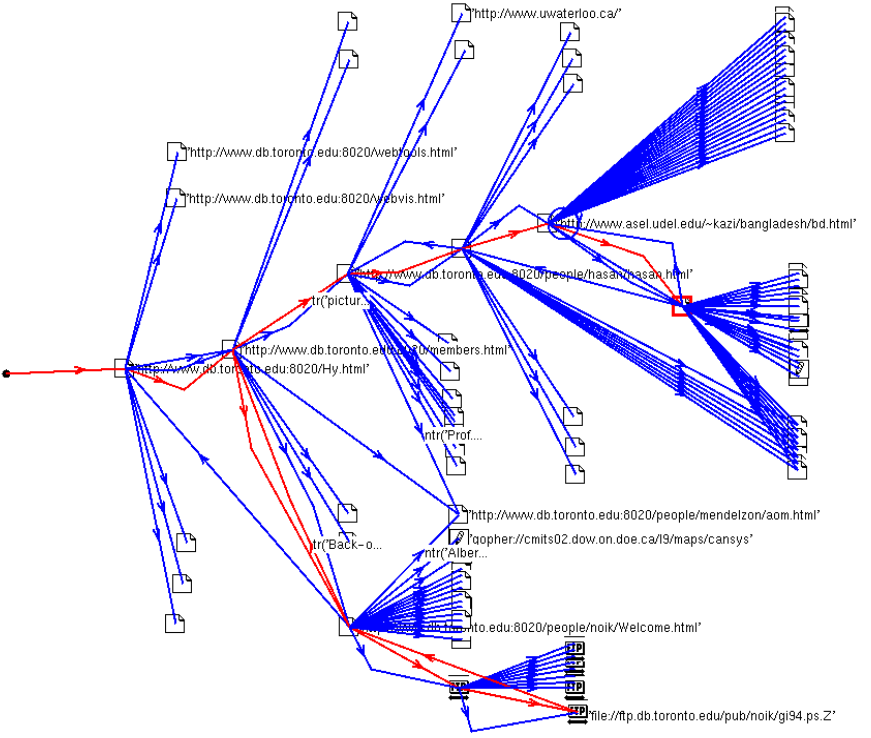

Fig 1.13. Hy+ web usage visualization tool

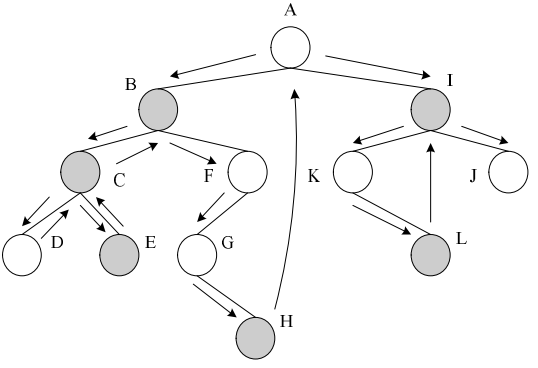

Fig 1.14. Visualisation navigational visitor's patterns using graph representation

In practice graphs can visualise electronic purchasing patterns. Fig 1.14 depicts empty circles as navigated pages without items selection and grey circles pages where items were purchased. Ascribing letters to each page (node) and arrows showing browsing directions allows easily identify what items are purchased in what sequence (Yun et al. 2000).

1.10. Summary of the First Chapter

1. Performed theoretical survey about KDD process, detailed KDD schema is demonstrated and explanation is provided for each step. Established relationship between data mining and its branch – web mining.
2. Investigated essential characteristics of web mining. Taxonomy is depicted and demonstrated that web mining consist of 3 subareas: web structure mining (analyses web site topology/links between pages), web usage mining (retrieves users navigational patterns), web content mining (retrieves text from web data).
3. Explained peculiarities of each web mining subareas and tasks that can be achieved using various data related to the web.
4. Analysis of different data collection sources is provided. Depending on the source and on the way how data is collected, it is used for different purposes and for different tasks: unstructured (text on the web documents), semi-structured (HTML documents, web log files) structured (XML documents, additional information about the visitors).
5. Knowledge discovery schema was presented which highlights steps necessary to perform web log mining process. The first step is web log data collection. Demonstrated the most popular web log file formats, explained the meaning of each field in the log file.
6. A deep and extensive analysis was performed on the data preparation (pre-processing) steps. A detailed schema is provided which points out following pre-processing stages: feature selection, irrelevant records removal, user identification, session identification, data sampling, data transformation.
7. Performed detailed analysis on each of these stages, explained various methodologies proposed by research community. Some of them never been known as a part of pre-processing process, therefore methods were classified, systemized and assigned to a relevant preparation stage.
8. On the basis of this theoretical investigation, it was established that data pre-processing takes a majority of time in knowledge discovery process as it faces various user and browsing sessions identification difficulties. The

processed data influences analysis stage by reducing number of records, analysis time and quality of the results – this is why this stage is so important.

9. An extensive selection of literature is provided to show the recent findings in web log data analysis and web log data visualization techniques.

10. It is concluded that web log data analysis is rather different from general KDD process and special attention must be paid to its unique areas: data cleaning, as many records do not represent actual user clicks, user identification, as free-form Internet structure means that most users access most web sites anonymously, session creation, aggregation of page requests made by a particular user over a period of time.

11. Material which was collected and presented in this chapter is a comprehensive guide into web mining area.

2

Web Sites' Design Survey

2.1. Introduction

This chapter introduces the Hyper Text Transfer Protocol (HTTP) and Hyper Text Mark-up Language (HTML). They form the WWW concept. WWW documents on the Internet are hosted by the web servers and special software programs display them in the browsers. This chapter explains information transmission process from the web server to the browser. Depending on the web site type, different amounts of irrelevant files are collected by web servers. Therefore in order to understand and be able to enhance web log mining process, reader should be familiar how web sites are created, what kind of structures and techniques are available and what kind of data is recorded into web servers according to the web site structure.

2.2. Web Browser

Web browsers or just browsers are programs used to access web pages and located in client's computer. Browser understands and presents text, graphics,

images, tables, forms and etc. It receives web pages, parses HTML code and displays them into the browser in a human friendly form.

2.3. Hyper Text Transfer Protocol

The Hyper Text Transfer Protocol (HTTP) is the most popular protocol on the Internet today. HTTP is the protocol for the World Wide Web (WWW). Like most of the other Internet protocols, HTTP requires both a client and a server to transfer data. The transfer is accomplished through the Transmission Control Protocol/Internet Protocol (TCP/IP) which used by web servers and also links computers connected to access Internet. HTTP server is also known as HTTP *Deamon.* It is a program that listens for HTTP requests on certain machine's port (default is 80). It also denotes physical location of the computer that stores those documents. When a client's side opens the TCP/IP connection, it transmits the request for a document and waits for a response from the server. Finally, when the request-replay sequence is completed, the socket is closed.

The request is defined by the Uniform Resource Identifiers (URIs and URLs). These short strings are addresses into information resources: documents, services, electronic mailboxes, images, downloadable files and etc. They make resources available under access method such as HTTP, file transfer protocol (FTP) and may be accessible through a single mouse click. Uniform Resource Identifier (URI) is a general name of the string which refers to the resource. Uniform Resource Locator or just URL is a part of URI and is associated with such popular URI schemes as HTTP, FTP and mailto. The structure of the URL is hierarchical. The first part specifies protocol used to transmit the resource. The other part indicates path, files name and other specific to the requested file symbols, e. g.: 1. protocol name, e. g. HTTP, 2. domain name, e. g. www.mii.lt, 3. port address, e. g. :80 (HTTP default), 4. directory where requested file is located, 5. name of the requested file, 6. internal links or anchor #. Typical example of the URL: *http://www.mii.lt/index.php?siteaction=personnel.browse&* has following meaning. Protocol used is http, accessed via www in the domain mii.lt, which is in Lithuania, the file which is downloaded is "index.php?siteaction=personnel.browse&".

Another example is *ftp://ftp.leo.org/pub/program.exe*. This URL is interpreted as follows. Protocol is ftp, the resource is on the ftp machine which is part of ".org" domain. The resource located in the "pub" directory and the file is "program.exe".

HTTP connection requires both web server and client participation. Next sections explain how client and server side HTTP request works.

2.3.1. Client Side HTTP Request

Browser starts the action - sends a request asking for a file in the defined location by URL. Since URL is an addresser, it allows web browser to know where and how to go to the desired location. Following actions are performed accessing the web site (Jeffry Dwight et al. 1996):

1) Browser decodes host of the URL and contacts web server.

2) Browser gives the rest part (directory, files name, internal links) of the URL to the server.

3) Server translates the URL into a path and file name.

4) Server sends the file/page to the client's browser.

5) Server closes the connection.

6) Browser displays the document/page.

Example of the client's part HTTP request. The requested page is http://www.mii.lt/index.php?siteaction=personnel.view&id=217:

```
    GET      //index.php?siteaction=personnel.view&id=217
HTTP/4.01
    User-Agent:
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+98;+Win+9x+4.
90;+sao)
    Host: www.mii.lt
    Accept: image/gif, image/jpeg, */*
```

The request contains method, requested document, HTTP protocol version that the client uses (HTTP/4.01), software/browser name and version, the server host and type of the objects or applications on the server which client can accept.

2.3.2. Server Side HTTP Response

Server side response of the above described request will look like:

```
HTTP/4.01 200 OK
  Date: Thus, 05 Sep 2007 16:17:52 GMT
  Server: Apache/1.1.1
  Content-type: text/html
  Content-length: 1538
  Last-modified: Mon, 05 Oct 2006 01:23:50 GMT
```

The first line contains HTTP protocol version and response code (200 OK means that page was retrieved successfully), data and time of the retrieval, web server type, name and version, downloaded page type, the size of the downloaded

object, last page's modification date and time. After the head of the response, the content of that page is presented. However just the textual part will be received. All images, sound and other files inside the HTML are retrieved from the web server through the supplementary HTTP protocol connection. Link references are retrieved from the original location also during additional connections when the user clicks them with the mouse.

2.3.3. HTTP Proxy Request

Sometimes proxy/cache servers exist on the client's side. These servers are the system that sits between an application (such as Internet Explorer, Netscape Communicator or similar) and a web server. It intercepts the requests to the web server to see if it can substitute the request reducing the number of requests that go out to the Internet. This is the outcome that proxy server can cache files it downloads from the Internet. Requested web pages are stored in a cache. When a previously accessed page is requested again, the proxy can retrieve it from the cache rather than from the original web server. Therefore, time for retrieving web page is saved and Internet network traffic is reduced.

2.4. Hyper Text Mark-up Language

Hyper Text Mark-up Language is abbreviated as HTML and sometimes is called a Hypertext. Hypertext is a type of semi-structured text/document, which forms HTML code and contains information as text visible on the screen while browsing, references to other pages and embedded objects like images, sounds, scripts and frames (Perato et al. 2001). This code describes how documents should look like and where different objects on the document have to be situated. Hypertext allows to link information in a way that is not linear like the pages of a book, but associative, so that people can choose their own path through link references. Hypertext links documents from distinct resources owned by different authors.

HTML document is transmitted via HTTP protocol from the server to the client. The server part contains the code and HTTP protocol passes it to the web browser. Then web browser interprets the code and users see the content of that code in a way human can interpret.

The HTML code consists of special tags which describe how the browser should format the document. Any tag starts with <tag> and ends with </tag>. HTML code is situated between <html> tags. The header is enclosed into <head>…</head>, title of the page – <title>…</title>, the text body –

<body>…</body>, table – <table>…</table>, lines/rows - <tr>…</tr>, columns - <td>…</td>.

HTML anchor tags sometimes called just anchors defined as <A>…</A> *anchor* type tag marks links, e. g.,

<img alt="Svetainės struktūra" src="..." ... >

These tags can be placed under the text or image and plays a crucial role in ability to browse from one page or site to another with the help of links. The detail description of the HTML code can be found in literature (Yi et al. 2000; Musciano et al. 2004).

2.5. Web Site's Design Structures

At the beginning web pages were developed for embedding text, images and tables into the HTML code. Due to its static feature, it was very inconvenient update them frequently. Therefore, with the appearance of scripting techniques, it became available to generate web pages whose contents change dynamically. Exist two types – client and server side scripting.

Client side scripting technologies are those that are part of the web client (the browser), embedded into HTML code (see Fig 2.1) and executed on the client's machine. The scripts created using Java Applets, Java or Visual Basic (VB) scripting languages, ActiveX components and etc.

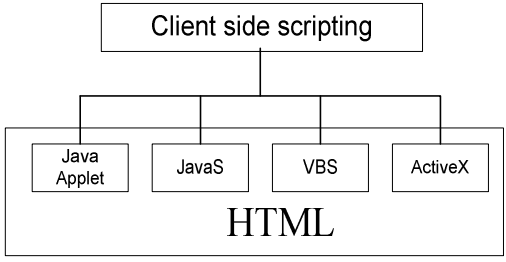

Fig 2.1. Client side scripts are embedded into HTML and in most cases executed locally

Technologies implemented on the server called server side scripts. The most popular types are: Active Server Pages (ASP), Hypertext Processor (PHP), Cold Fusion (CFM/CFA), Java Server Pages (JSP), Common Gateway Interface (CGI) and Application Programming Interfaces (API) for web programming. All those

technologies have their own syntax and their implementation is also different (see Fig 2.2).

Server side scripts are classified into two groups:

1) *Implemented into the HTML code* as small programs and recognised by the specific syntaxes. The most popular ones are ASP, PHP, CFM/CFA and JSP.

2) *HTML code is embedded into scripts*. Scripts are recognised by the specific syntaxes as well. Here two types can be shown CGI and API.

First group are called CGI are run as separate programs while request to the web server is submitted. These files also can be divided into two groups: executable *.exe, *.dll and interpretable. With extension *.exe behave as executable programs and with *.dll as dynamic libraries. The difference is that this type of scripts cannot be executed without an interpreter. For example, script with extension *.pl understood by the PERL compiler which reads and interprets files with the extension *.pl. The other group is called API and depending on the web server type called differently: ISAPI, NSAPI, WSAPI. These scripts share the same process space as web servers.

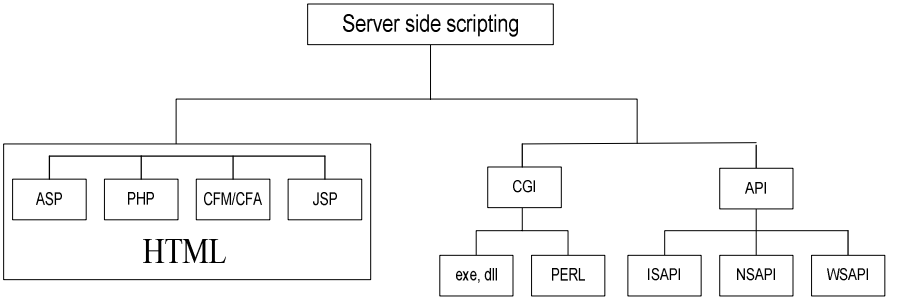

Fig 2.2. Server side scripting can be divided as "inside HTML code" or "HTML code is inside the program"

Additionally to dynamic pages other types of web page enhancements exist. Frames are used to have more attractive and complex web page view on the browser screen. Using frames browser's window is divided into parts to display data in each of them. Each window can contain information from different data sources.

Detailed description of the above mentioned techniques is given in the forthcoming sections.

However, major directions designing web pages is depicted in Fig 2.3. So, firstly static or dynamic web page design strategy must be selected. To improve their capabilities and attractiveness, frames and scripting on the client side can be implemented. Dynamic pages development process has different concept to

static. In the case of static pages, request is send to the browser not invoking additional activity. Requesting dynamic pages programs-scripts are execute on the server in order to get page contents.

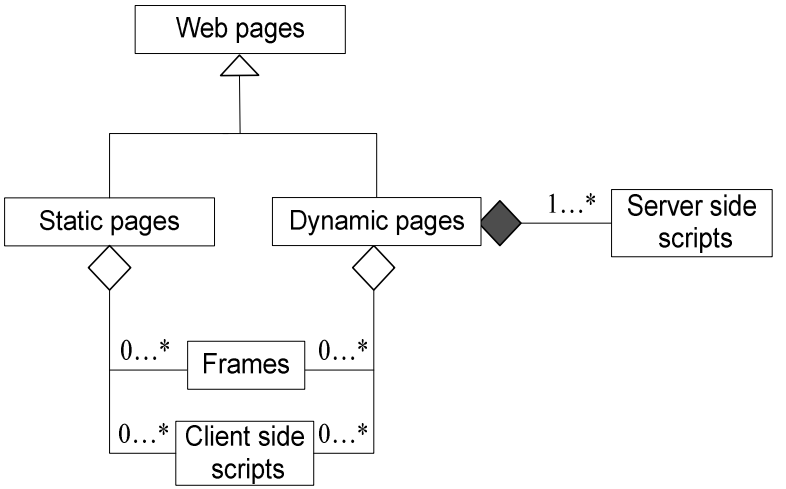

Fig 2.3. The prototype of possible web page design structures. Web pages are two types: static and dynamic. Static and dynamic web pages are made up of frames and client side scripts. 0…* indicates that relationship is of type null (not exist) to many. Dynamic pages are accompanied with server side scripts. Filled rectangle means that server side scripts are compulsory part and relationship 1…* means association is one to many

2.5.1. Static Web Pages

Static web pages are made up of objects such as text, images, link references and tables. These objects are included into HTML code manually. Making changes in a static page involves using HTML editor. This can require additional knowledge on how to maintain and design web pages. However, not everyone can make these changes and specialist – a designer/maintainer should be involved. Due to the rarely and late updates static web sites present not very new and valuable information that is why they are not popular for commercial purposes.

Fig 2.4 shows how web browser retrieves static pages from the web server. When user requests a page, web server takes the document from the disc and sends it back via the HTTP protocol to the browser. Then additional HTTP connections are generated in order to get different embedded files located inside

the requested document. Alternative data sources do not participate during static page retrieval process. Static web pages can also contain scripts executed on the client machine (see section "Client side scripting").

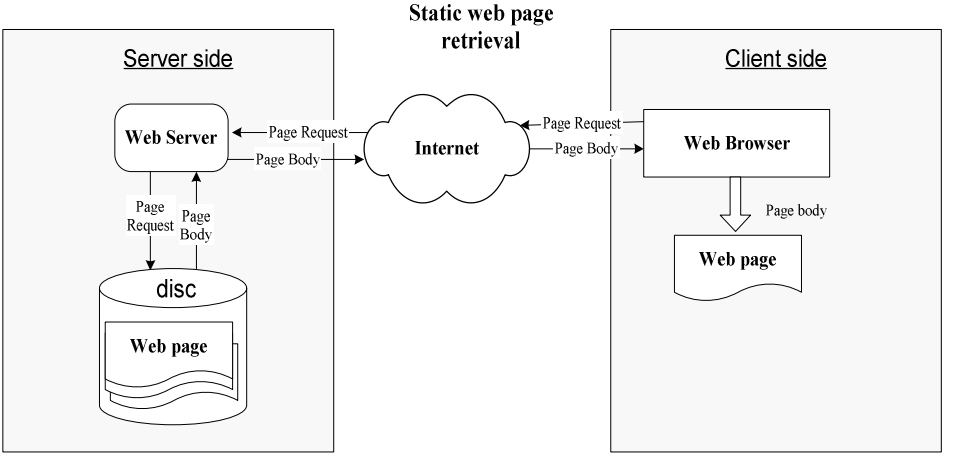

Fig 2.4. Static web pages are generated by web servers using information placed straight on html documents

2.5.2. Dynamic Web Pages

Dynamic web pages like static contain such object as text, tables and images. Which are embedded into HTML code. Additionally to this Fig 2.5 illustrates two types of available scripts for dynamic pages. Ones are optional and are run on the client side. They usually do not involve web server into the process and act as in static pages case – locally on the machine where the page is watched. Others are always generated invoking server side scripts to present the content of the page. Therefore server side scripts are always present. From the design point of view, dynamic web pages contain very few texts however server side scripts pull information from other applications like databases, files and other data repositories which is then presented to the client's browser.

Dynamic web pages change their contents depending on the environment. It can be user actions or the contents of file/database (see Fig 2.5). Let's examine user's action first. Web browser sends user's request in a form of a string to a web server. Web server passes it to a scripting engine. The engine receives the string, reads at the top of the script file and works through the list of instructions according to the string properties. These instructions tell the server to do things such as create an HTML page, add a particular database record, modify or search

the database and many more things. Then generated page is returned to the browser as a plaint HTML.

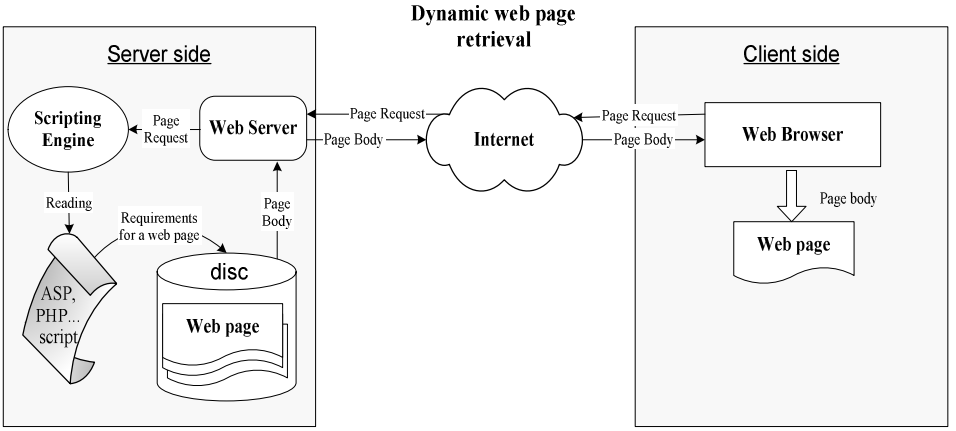

Fig 2.5. The content of web pages changes depending on the different parameters selected and sent by the user and on the contents of data repository itself

2.5.3. Frames

Frames are ability to publish web pages in a more advanced way. They divide browser's window into different panes so that information in each of them can be viewed independently. Adding frames special anchors FRAMESET, FRAME and NONFRAMES are included. The FRAMESET is the "body" of the frame. Usually it contains size of different panes. FRAME attribute identifies a single frame in the frameset. It contains information about the frame name URL name associated to that frame and other kind of information. NONFRAMES anchor creates possibility to observe frame based web pages with the browser which does not support frames.

Example of the frame based web site. The main window is divided into the desired number and shape of panes. Fig 2.6 demonstrates four panes: left up, right up, left low and right low. The prototype of the HTML code with frames is presented on Fig 2.6. Frame tags <FRAMESET>… </FRAMESET> define frames existence. <Frame Scr = …> refers to additional HTML file, in this case *frame_index.html*, which defines files presented on a different pane. FRAME_A.html is assigned to the right up part of the browser and thus, the contents of FRAME_A.html file will be viewed. The same with other files FRAME_B.html, FRAME_C.html and FRAME_D.html. The last one FRAME_D.html is created for clients which browsers do not support frames.

```
<HEAD>
   <TITLE>FRAME Example</TITLE>
</HEAD>
<FRAMESET Rows="104*,154*" cols="*">
  <FRAMESET rows="*" Cols="294*,397*">
    <FRAME Src="frame_index.html"  Name="ul-frame">
    <FRAME Src="frame_index.html"  Name="ur-frame">
  </FRAMESET>
  <FRAMESET rows="*" Cols="294*,397*">
    <FRAME Src="frame_index.html" Name="ll-frame">
    <FRAME Src="frame_index.html" Name="lr-frame">
  </FRAMESET>
</FRAMESET>
<NOFRAMES>
   <BODY>
```

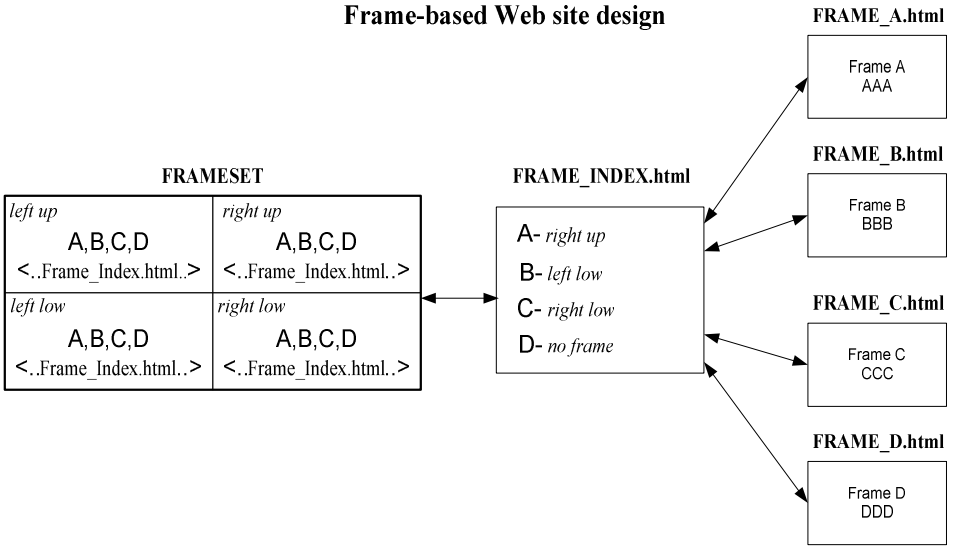

Fig 2.6. Frame based pages design

Frame based web site works as follows. When such a page is requested, browser shows the "STARTING VIEW" (see Fig 2.7) of the page FRAMESET together with the link references "A, B, C, D". These links were embedded into the main FRAMESET during the design process. Therefore, for example, by clicking the link "A", the FRAME_A.html page is displayed on the right up pane, since link

A is associated with that position on the window. The same is with the other links, except D, which is suited for non-frame pages.

When D link is requested, information from FRAME_D.html file is displayed on the newly opened browser window without frames (see Fig 2.7).

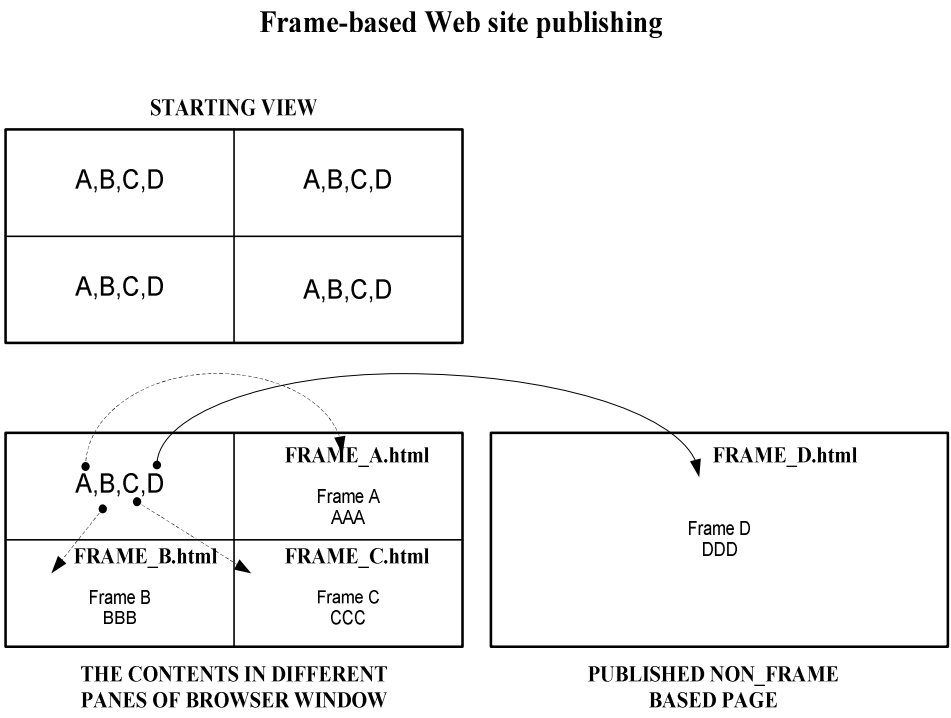

Fig 2.7. Published web site with frames and the content of each pane window is presented

2.6. Client Side Scripting

2.6.1. Java Applets

Because Java is a programming language, it allows not just creating standalone programs and programs which are embedded into the web pages. Parts of the Java language capabilities are special programs – *applets*. *Applets* can only be run on a web browser (client's side) or in a special program meant to run applets like the *appletviewer*. Java applets are created to run in web pages

and thus can significantly improve browsers view using a range of functionalities not available to standalone applications. Java brings more interactivity to WWW because, web applications become real instead being plain documents containing static information. Java *applets* allows having sound, animated graphics, even own user interfaces. An example of such interface can be a small window that displays current stock data and gives a price when users enter sample amounts of the stock to buy.

Having possibility inserting programs into the web page using special tags, Java opened new possibilities presenting information on the web. Java *applets* which are embedded into HTML pages use <APPLET> … </APPLET> tag. This tag contains specific information regarding to the Java *applet* which will be presented on the page. Tag <APPLET> customizes applet attributes such as code, area size where *applet* appears on the page, e. g.:

```
<html>
<applet code=filename.class width=n height=n>
</applet>
</html>
```

Java *applets* are not restricted by the network or server overloading. Once they are transferred to the local computer, they are executed according to the technical characteristics of that computer. It is up to the browser to interpret Java *applets* and execute them. When the user loads a page containing Java code, in the form of Java *applet*, the browser tells the applet to start. When the user leaves that page, the applet is told to stop. Java applets are much more independent than, for example, JavaScripts which are executed on the "event handler". Java *applets* are independent programs. They may respond to a mouse click within its active area, but it won't be listening for the *submit* button being pressed. An applet is a little application. JavaScript code is more like a *Dynamically Loaded Library* (DLL), which is activated in response to something.

2.6.2. JavaScript and VB Scripts

Java or Visual Basic scripting languages are small pieces of programs embedded into HTML code with a purpose to enhance its capabilities, increase functionality and interaction with the end user. JavaScript is based on the Java and C++ languages; VB Script – on Visual Basic and Visual Basic for Applications. The syntaxes of scripting languages differ from the original languages but some similarities in syntax and structure have left. The difference from the original Java or VB programs is that functions written using scripting

languages cannot be executed alone not being implemented into HTML code. Below are examples of the VB and JavaScript code implementation into HTML:

```
<SCRIPT
LANGUAGE="VBScript">
VBScript
commands.</SCRIPT>
```

```
<SCRIPT
LANGUAGE="JavaScript">
JavaScript
commands...</SCRIPT>
```

Web pages become more attractive and alive using script languages. Java or VB scripts generate events because they have special purpose items such as "event handles". They allow the developer to write functions that contain code triggered by specified events which may occur on the specific user actions, e. g., clicking *Submit* button of the form. Examples of possible functions generated using Java and VB scripts are (1) rotations image or banner on mouse over them; (2) form validations, e. g. an application which collects and then validates input data of the correctness before sending it to the server; (3) directions/redirections; (4) actions of a mouse on the picture or over an anchor generates certain event.

Web pages containing client side scripts are retrieved from the web server in a usual way like static pages. During the first connection to the web server text of the page is retrieved. During other connections web server passes embedded parts to the web browser. The existence of scripts becomes noticeable when user performs certain set of actions which invokes "events handles" to be active. If say script programs are asked to present additional files or images, request is made to the web server again (see Fig 2.8).

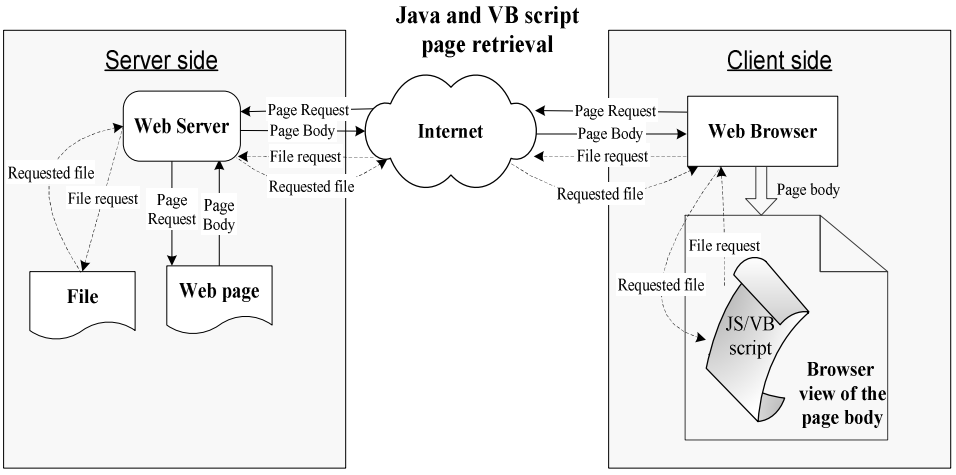

Fig 2.8. Page retrieval of the documents containing Java or VB scripts

2.6.3. ActiveX Components

ActiveX is a centrepiece of Microsoft's overall Internet strategy (Afergan et al. 1996). They are similar to Java and VB scripts, since are embedded into HTML to enhance web capabilities. The difference is that ActiveX components are fully featured executable files able to do anything any other executable computer program can do. It can be used to run movie, video clip or animated images. Even more, they can contain links, act on user actions and do many more. ActiveX components have a big concern about the security since they are small programs running on the client side and have full access to the user's file system and therefore cause serious damage. Due to this Microsoft provides an authentication who is wrote the control.

ActiveX component is an object having three parameters. The first parameter of the component describes the type. The second checks if there is a control on the computer. The third parameter describes multimedia itself (name, size and etc). For example, Macromedia multimedia type flash plug-in is an ActiveX component. When browser loads the page with the ActiveX component it detects the flash type and checks if there is a player installed on that computer to run it. If no, then browser retrieves it from an appropriate address defined in the parameter. Finally the security statement is presented about the control creation source. And if the user accepts it, control is downloaded, installed and run automatically.

2.7. Server Side Scripting

Server side scripting can be divided into two categories according how they are implemented: inside the HTML or containing HTML code inside.

2.7.1. Script in HTML

This category of scripts is popular because it allows creating sophisticated pages, controlling and manipulating data sources. Text fields and text areas can change dynamically in response to user responses. This category of scripts are implemented in HTML and executed on the server side.

PHP scripting language can be used as an example how dynamically control data source. PHP code is embedded while creating the page by web site developer. Say script has instructions to present articles of different formats such like word (*.doc) and adobe acrobat (*.pdf) to the user upon his request. For example, user asks for a word document of the article. The string containing requested type of the document and name is send. Then script looks for file name

having extension (*.doc). If it is found, then script produces the output of word document and passes back it to the user.

The main difference between scripting techniques is that they are created using different tools, platforms and based on different syntaxes. Detailed description of these scripting techniques can be found in (Active_Server_Pages), (PHP_Hypertext_Preprocessor), (JavaServer_Pages(TM)_Technology).

2.7.2. HTML in Script

CGI is the representative of the category where HTML is embedded into the script. CGI acts as a link between an application (e. g. database) and a web server while the server receives and sends data back to a browser. Lets analyse how CGI woks. When web page is requested, browser sends information in the form of a string to the server. Server executes CGI script. The scripts access the database and looks for information the browser requested. After information is found, it is formed into HTML and sent to the server which passes it to the browser. Different languages like C, C++, Java, Perl, SH and etc can be used for CGI scripting. Developer can choose according availability and requirements.

There are many areas where CGI scripts can be applied. For example, it is often used for processing information entered into forms. The script is activated when the button submit/send is pressed. The script processes information and then sends back it to the browser through the web server. This action allows displaying information entered by the client on the screen. Additionally CGI scripts are used to animate images, generate customized HTML pages, show current time and date, name and version of the browser, include internal counters to calculate number of visits to the site and etc.

2.7.3. HTML in Script – Embedded in HTML Server

CGI applications, by their nature, must be loaded and unloaded each time they are used. There is no way to load a CGI script and keep it ready for future use. The Application Programming Interface (API) provides an alternative to CGI programs because has higher performance type capabilities for web servers. API allows web developers to build applications that are executed much faster, because their components use the same process space as web server. This means, that API programs become parts of the web server. API applications are Dynamic Link Libraries (DLL) that loaded into memory and stay there at all times. Limitations having them are that due to the type of executions (working in the same process space) an access violation by the API application may crash the HTTP server. Additionally to API applications, API filters are used for the following enhanced tasks: secure visitors authentication to the web site handling

visitor's user name and password, message encryption (e. g. credit cards details are send securely using API filters).

Several types of API according to a distributor exist: ISAPI supported by Microsoft Internet Information Server (IIS), NSAPI supported by Netscape Commerce and Enterprise Server and WSAPI supported by O'Reilly Web Site and Web Site Pro.

2.8. Analysis of the Files Recorded into Log Files Depending on the Web Site Structure

Web servers record all files downloaded to the user. These files can be of the types as text (T), images (I), client side scripts (CSS) assumed that they retrieve additional files, server side scripts (SSS), frames (F). The following table (see Table 2.1) can be noted.

Table 2.1 Web site structure and file types recorded into web log files

	Text	Images	Frames	CSS	SSS
Static pages	●	●	●	●	
Dynamic pages	●	●	●	●	●

Running static web pages log file contains all text documents. All embedded files such as images are also logged no matter that they were not requested by a purpose. The presence of client side scripts in a log file is not noticeable until additional files are not loaded. Running dynamic web pages, beside mentioned files additional records are logged into the log file. For example, below is an example of dynamic web page log file as a result of PHP script:

```
   "GET          /Charmaine/thumbnail.php?p=DSCN7784.JPG
HTTP/1.1" 200 5669
   "GET          /Charmaine/thumbnail.php?p=DSCN7878.JPG
HTTP/1.1" 200 4228
   "GET          /Charmaine/thumbnail.php?p=DSCN7880.JPG
HTTP/1.1" 200 4228
```

As can be seen, every transaction contains the same text up to the tag "php?p=", after, arguments appear which identify different files.

A number of transactions recorded into the log file having frame based web pages is huge. The web server logs a number of requests instead simply logging

just one transaction. This is because of the design of frame pages described earlier. The picture (see Fig 2.9) presents an example of the typical use of frames. The picture shows HTML code, where *index.cfa* presents a complex structure of the web page. It consists of a new *frame* <*frameset…*>. Every frame has its own name <*frame name…*>. The first frame contains a page with the name *topbar.cfa*, the second frame splits into two different web pages: leftmenu.cfa and notrecognised.cfa. When the page *index.cfa* is requested, all three other pages are downloaded as compounded parts of this frame. At the same time web server logs request of *index.cfa*, */topbar.cfa*, *leftmenu.cfa* and *notrecognised.cfa* into the log file. It is very likely that just one of these files presents the desired information.

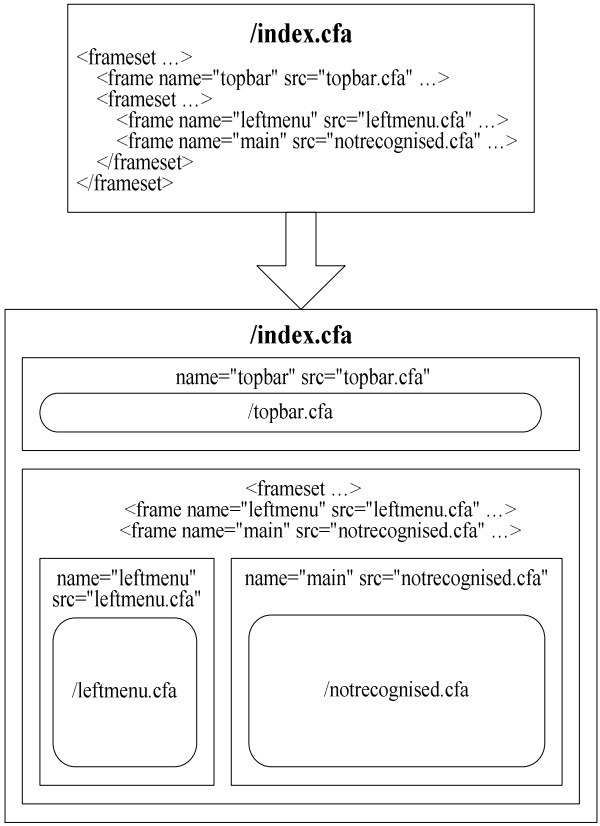

Fig 2.9. An example of web page containing frames

2.9. Summary of the Second Chapter

1. In order to understand what kind of files are recorded into web log files, a deep overview was provided, which explains how HTML protocol and page transmission process from the server to the client machine works.

2. Introduced main concepts of HTML language. Comprehensive analysis was performed on existing web page design concepts and technologies. Detailed schemas provided with scripting techniques. Distinguished and explained differences between two types of scripting: client side (scripts run on the client machine), server side (scripts run on the web server). Demonstrated web page transmission process using various types of scripting languages and special features – frames.

3. This survey demonstrates that exist many design structures. Because exist different web site creation methods, files having different extensions are recorded into web server. It is difficult or sometimes even impossible to access if this file represents the actual user click without manual check. If the web site is huge, this manual checking becomes even more painful. Ten or fifteen years ago, when web log mining area has just started and web sites were quite simple, standard cleaning did most of the job. With nowadays technology, it is very inefficient method. This chapter highlight the need of new web log data cleaning concept which is introduced in Chapter 3.

3

Link Based Cleaning Framework

The main results discussed in this chapter are published in (Pabarskaite 2002).

3.1. Introduction

Nevertheless, though data cleaning is an important step in data mining, data warehousing and other database areas, it has received relatively little attention from the research community (Lup Low et al. 2001). Data cleaning involves various tasks to produce data which can be successfully used for analysis (Fayyad 1996). It is also a time consuming process as in many cases it is still semi-automatic and therefore not efficient (Famili et al. 1997). Data cleaning have to be automated to make data sharing affordable to monitor and evaluate (Infoshare). Currently, research about data cleaning in large repositories covers the following problems:

- Data duplication. Data repositories often have large amounts of duplicate entries. This duplicated entries can be difficult to remove without the assistance of intelligent systems because usually analysing these items is complex and domain dependent (Hernandez et al. 1995).

- Missing values. They are not appended during the entry process by error or on purpose.
- Data entry mistakes. This is when data is placed in the wrong fields.
- Irrelevant data. This is the case when data have no appropriate value to the intended process.
- Spelling variations, e. g. between US and UK English language: visualization and visualisation (eng.), disk (amer.) and disc (eng.).
- Non standard representation. For example, there are about 150 ways of representing the time and date.
- Unit differences. The same thing, for example, some departments describe labour in terms of hours, others in weeks, ones measure everything using imperial metric system others decimal and etc (Infoshare).
- Incomplete values. It means that records are too long for certain fields. For example, field length is defined having 256 symbols, but actual record is longer. In that case original record is truncated and placed as incomplete.

Cleaning is an important step in web log mining. The major part is removing irrelevant records. Other problems such as incomplete data fields, spelling variations and etc. are not so important since data collected into web log files is not gathered by humans and no other kind mistakes can be practically found. As far is known, there was no deep and comprehensive analysis in the area of web log data cleaning. No comparison and systematisation was provided. Therefore this chapter of the thesis is dedicated to highlight web log data cleaning issues by providing new method how to solve existing problems.

3.2. Irrelevant Records

Many web log records are irrelevant and require cleaning because they do not refer to the pages visitors click on (Markov et al. 2007). The following types of records in most cases are treated as irrelevant and removed from the web log:

- graphic (images, sound, video) files in web pages and are downloaded together with the document during the request process and in most cases they are not visitors' subject for observation therefore they are likely to be redundant (Han et al. 1998; Cooley et al. 1999; Faulstich et al. 1999; Cooley et al. 2000);
- files generated from automated processes (search robots);
- records, where time interval between requests is too short to catch the contents of the page;
- repeated requests for the same URL are made from the same host during the same browsing session;

- a set of requests from one host all of whose referrer page is empty. It means that if referrer page field is empty, then this URL was typed by the user or this page was bookmarked or requested by some program-script (Berendt et al. 2000);
- users connecting through cache or proxy servers (Pabarskaite 2003).

From the listed records cleaning methods it was decided to make cleaning systematization (see Fig 3.1) which covers known to the author web log data cleaning issues.

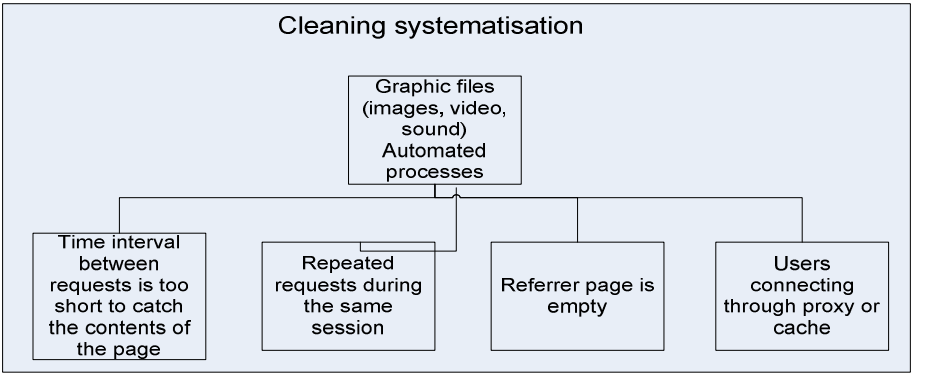

Fig 3.1. Cleaning systematisation

The most significant part is removing irrelevant graphic and various files related to the automated processes and files which are necessary for the web site creation. Then another cleaning stage is applying various methods to provide further data refinement. However, these methods have many limitations and not always possible to apply them. The following further data cleaning methods can be used as a part of web data cleaning process:

Time interval – time to observe the page cannot be defined in advance; it depends on the page contents. It is logical to make an assumption that 1 or 2 seconds is too short to catch the contents, however the final decision has to be made by web log data analysis.

Repeated request during the same session – it depends on the data analyst's goals. Sometimes it is worth having the all visited pages even if they are repeated (e. g. A, B, D, E, B), but in other tasks, only unique pages are interested (e. g. A, B, D, E). So this step depends on the final tasks requirements.

Referrer page is empty – this feature depends on the web server's configuration. It might be "turned on" or "off". So if it was turned "off", the referral field is empty and it is not possible to trance where from the user comes

from. It also depends on the task, if we are not interested where from users come from to our website, this type of cleaning is irrelevant.

Table 3.1 Extensions of redundant file types which appear in web log data

Extension	Description of the file	Example of file
GIF	Image file, extended name is “Graphics Interchange Format”	reports.gif
JPEG	Image file, extended name is “Join Photographic Experts Group”	percent.jpg
BMP	Image file extended name is “Windows Bit-Map”	street.bmp
PNG	Portable Networks Graphics, for well compressed images	rect.png
TIFF	Image file, created by scanners	pict.tiff
WPG	Image file, “WordPerfect Graphics”	image.wpg
MPEG	Video/Audio file presents video or audio data, extended name is “Motion Picture Experts Group”	movie1.mpeg
MOV	Video/Audio file by QuickTime presents video or audio data	movie2.mov
ICO	An icon so that when visitors to your site bookmark or add it to their "favourites" a nifty little icon shows up instead of that grab default explorer icon	favicon.ico
SWF, FLV	Flash- Shockwave animation	game.swf
JS	Java script (additional extensions)	/js/scroll2.js
TXT	Text file used for search engines. It contains different information about the site (e. g. keywords) to make it found faster	/robots.txt
CSS	HTML cascaded style sheet, template files which allows lots of different pages can have the same heading description, fonts and etc	news.css
PFR	File defining picture frames	miller.pfr
admin	File related to administration actions	/admin/question.cfm
map	File contains the information that relates the coordinates of a mouse click to a URL that you want to be returned when that area is clicked. After a region is outlined, the user is prompted to enter a URL link	/map/glob.map
Zip, rar, 7z	Archived files	a.rar
Doc, ppt, xls	Microsoft office files	Document.doc

User connecting through proxy or cache – most users in our days use high speed internet connection, therefore this method when users connect through proxy/cache is not effective in current technologies.

Most irrelevant records are removed through the first graphical and administrative file removing stage. Table 3.1 lists the most common file extensions and describes types of the files. Sometimes the types of the files need to be removed depend on the data source. For example, if graphical repository is analysed and the task requires identifying associations between accessed images, then files referring to images should remain. Another problem is opposite. Additional files are not removed since it is difficult to recognise them as different scrip languages generate various files which are stored into web log files (some files are specific to the web site, created using special scripts languages). The conclusion drawn is that data cleaning from irrelevant records is strongly related to the domain and web site design structure. This issue is pointed by other scientist as well, the effect and necessity for data cleaning seem to be very dependent on the technical implementation of the web site (Herder O. et al. 2006). The list of files presented in Table 3.1 can change and can be extended by different script files activities.

Clicks stream analysis software Webtrends was used to demonstrate how cleaning is performed using standard web log analysis tool where data cleaning is based on file extension (Webtrends). It means that data analyst can select file types to exclude from the raw data. The data for demonstration purposes was taken from web server which monitors users activity to the IPA web site http://www.ipa.co.uk/. The test was the following; data was cleaned excluding pictures (gif, jpg, txt) and then top most visited pages selected. Results of this study presented in Table 3.2. The table shows top most frequently visited pages viewed by visitors. The first column represents names of the web pages, second shows number of times it was accessed. The last column was added artificially fro indication purposes. Values for this column came after checking manually whether actual page can be access by the mouse click. If page was accessible, indication of page relevancy “Link page” was added to the last column and if page was not accessible, indicator “Irrelevant” was added. As can be seen, Webtrends generate poor outcome, using standard technique removing records by file extension.

Table 3.2 Top(most frequently visited) pages according to Webtrends

URLs	Total number	Indicator
/ipacareers/	19 760	Irrelevant

/	17 988	Irrelevant
/content.cfm	**13 453**	**Link page**
/ipacareers/factfile/listcategories.cfm	**3 492**	**Link page**
/cpd/form.cfm	3 347	Irrelevant
/search/newsearchframeset.cfm	2 607	Irrelevant
/ipacareers/splash.cfm	2 496	Irrelevant
/aboutmembers/	2 312	Irrelevant
/careers/	2 296	Irrelevant
/careers/menu.cfm	2 181	Irrelevant
/ipacareers/home_mainlayer.cfm	**2 091**	**Link page**
/cpd/	1 886	Irrelevant
/ipacareers/welcome.cfm	**1 868**	**Link page**
/thankyou.cfm	1 742	Irrelevant
/alertreg_action.cfm	1 725	Irrelevant
/services/main.cfm	**1 440**	**Link page**
/cpd/login!.cfm	1 434	Irrelevant
/services/dynamicmenu.cfm	1 413	Irrelevant
/cpd/add!.cfm	1 406	Irrelevant
/members/map/searchoptions.cfm	**1 370**	**Link page**
/news/	1 353	Irrelevant
/services/	1 350	Irrelevant
/news/menu.cfm	1 343	Irrelevant
/services/training/cpd/loghome.cfm	**1 313**	**Link page**

3.3. Cleaning Module

As can be seen from the example in Table 3.2, standard cleaning method leaves many records which do not correspond to the actual users clicks. In order to remove them data analyst has to test manually whether this link is accessible. It is done by pasting the link into the browser bar. If response is received and the page is displayed to the user, then such a record is a link, otherwise, if an error is received and the page is not displayed, it means that record is irrelevant. This is a long process as it has to be done manually. Therefore, in the research work (Pabarskaite 2002) an automated method was introduced which we call "Link based cleaning". According to this method records which refer to the actual users clicks are retained and others removed. This method makes cleaning process much faster, as it is done automatically and results become more realistic. They correspond to the actual users' activity.

Method consisted from two stages. The first one – retrieving HTML code from the web server. The second one is filtering. Both these stages will be discussed in detail in the following sections.

The method was implemented in C++ programming language using Borland C++ compiler. The database used was Microsoft Access 2000. ODBS connectivity was used to connect programs in C++ to Access database. The prototype of the advanced cleaning engine was tested on Windows 2000 operating system, with 2GB of RAM.

3.3.1. Retrieving HTML Code from the Web Server

In the first "Link based cleaning" stage information from the web server is retrieved (see Fig 3.2). Web log file serves as a source of web pages to be downloaded. Then a special developed script takes each page from the web log data and passes it to web server requesting the contents. In other words this engine simulates mouse click operation. Web server returns the content (body) to the page analyser engine, which then parses HTML code, detects links by an anchor <a>… and locates them under the second "Page" column (see Fig 3.2) under the "Links Table". The procedure is repeated for all pages from the log file. The first column "Source" indicates files taken from web logs, the second column "Page" lists all links accessible from the page in column "Page". The third column is text displayed on links. However, there are some text tag problems. Sometimes, links can be images or JavaScript files and text is not specified. Thus, no text tag can be retrieved using these links and column "Text" stays empty on such records. In some other cases, text can be completely meaningless, for instance "click here", etc.

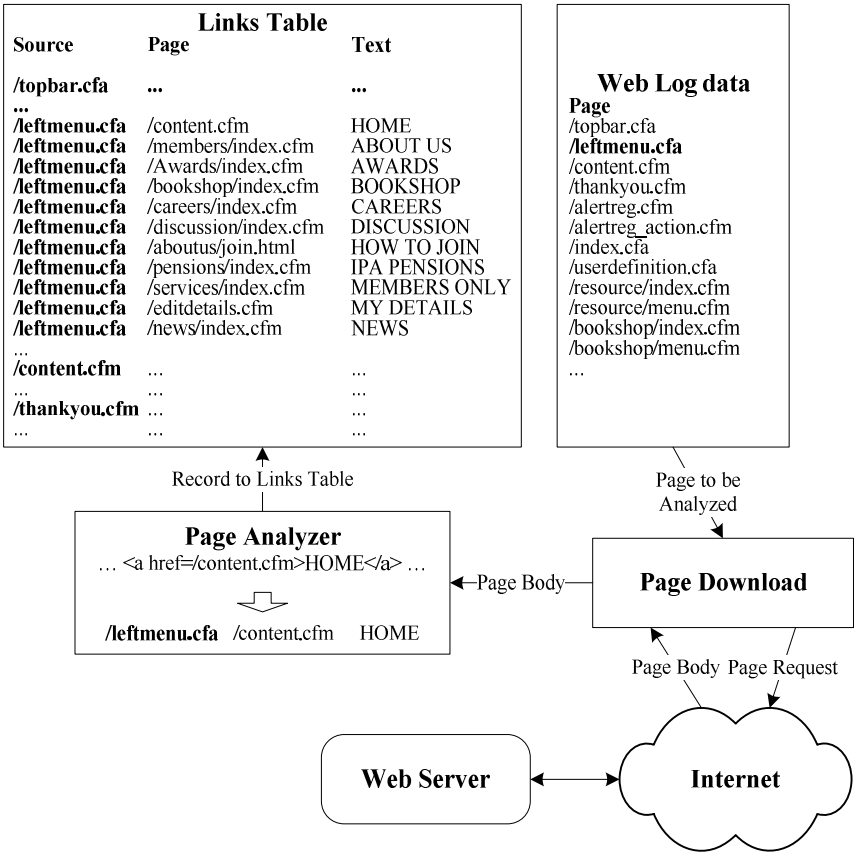

Fig 3.2. Page retrieval engine: it takes a file from web log, downloads it contents, analyses and allocates certain parts from HTML into "Links Table"

An example can be analysed (see Fig 3.2). **/leftmenu.cfm** from the log data is taken. The contents of **/leftmenu.cfm** is requested, downloaded and passed to the cleaning engine. Engine parses the *body* of the page which is HTML code, finds all references to links, e. g. **/contents.cfm, /members/index/cfm**, etc. (see column "Page") and retrieves references text tags (see column "Text").

After "Links table" is generated, the second part of the method is applied.

3.3.2. Filtering

The next step is filtering. During this stage irrelevant records are filtered from "Links Table". Irrelevant page can be defined as page which is found

among web log data (web server collects this data) but not accessible by a simple "mouse click". At the same time, this page cannot be retrieved by the developed engine. However this file appears in a log file because it consists as compounded part (e. g. frame, picture, video, script) of the page user actually was interested in.

The filtering engine works in the following way: it takes the file from web log data, checks if the file exists under the link references column "Page" in the table "Links Table". If the page is not found, it means it cannot be accessed and therefore should be removed. If the page is found, then it is stored into a cleaned data file (see Fig 3.3).

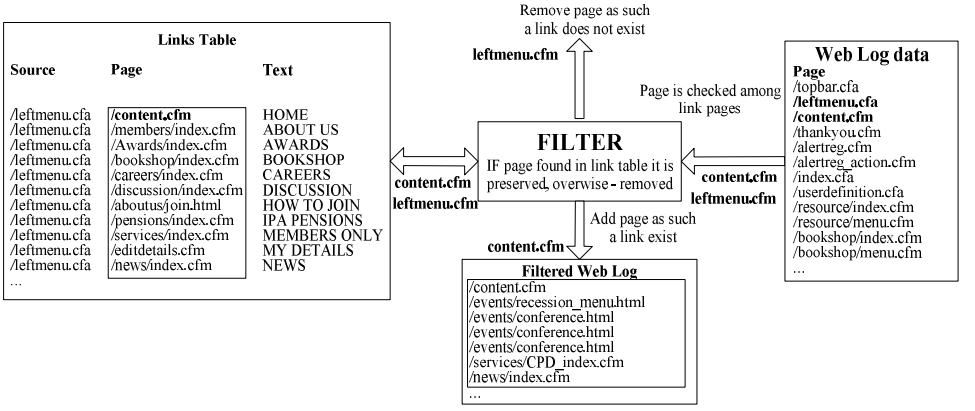

Fig 3.3. Filtering stage removes irrelevant files and constructs clean dataset

For example, see Fig 3.3 **content.cfm** is taken from the unclean web log file and passed to the filter. Filter finds content.cfm in "Links Table" under "Page" column, therefore **content.cfm** is recorded into "Filtered Web Log" table. Another example using /leftmenu.cfa. Filter finds that page does not exist in a "Links Table" under "Page" column, therefore **/leftmenu.cfa** itself is not a link and is removed.

Filtering is very efficient as it is difficult to measure which URL is relevant or not. For example, image files have extension *.gif, *.tiff and etc. In most cases files having these extensions are irrelevant and it is easy to remove files by indicating these specific file extensions. But if for example the web site is designed using frames, practically there is no technique except manual checking for file's relevancy. In this case the proposed technique performs automatic file checking and, as will be shown in the experimental section, with a high percent of confidence.

3.3.3. Link Based Web Log Data Cleaning Algorithm

Below is a pseudo code for the cleaning technique described above which consists of two parts: generating links table and filtering irrelevant records.

Definition: WL – web log, LT – links table, p – page (URL) name, PB – context/page body.

Input: WL
Output: LT
Algorithm #1: generating links table

```
1    procedure generate LT
2
3
4    LT = ∅
5
6
7
8
9    if WL ≠ ∅
10
11
12      for ∀ p ∈ WL
           PB = download page p body
           AH = find all hyperlinks in page body PB
           for ∀ h ∈ AH
              insert distinct pair (p,h) into LT (source and page columns)
           endfor
        endfor
     endif
```

Input: WL, LT.
Output: clean WL.
Algorithm #2: filtering irrelevant pages

```
1        procedure clean WL
2
3
4
5
```

```
6
7
8          if WL ≠ ∅ and LT ≠ ∅


             for ∀ p ∈ WL


               if p ∄ LT


                 remove p from WL
               endif
             endfor
          endif
```

3.4. Model Evaluation

It is rather difficult to evaluate how good web log cleaning is. There is no simple measure that can evaluate the overall quality of cleaned data. This situation is different from classification and/or prediction tasks in data mining, where quality of the model is determined by misclassification error or mean square error (MSE). The simplest way to evaluate the quality of the web log cleaning is an analytical one. Files that physically depend on a particular page and are always downloaded together must be eliminated. The purpose is to analyse user actions but not downloaded pages. Therefore, web log cleaning is actually a mapping process between downloaded page list and user actions. *The analytical evaluation is to determine cases where cleaned web log reflects user actions. The number of missed cases can partially reflect quality of cleaning.* For example, displaying web page involves downloading the page itself as well as numerous images and other files. Web log cleaning must filter all these helper files and leave only one page representing specific user action.

To evaluate “Link based” model, results were compared to the standard web log cleaning method which is based on removal files having extensions provided in Table 3.1 and used not only in research studies referred in (Dyreson 1997;

Cooley et al. 1997a; Cooley et al. 1997b; Markov et al. 2007) but also in a number of other works and applications. For evaluation purposes data from "www.civitas.lt" website was used. The data contained records from the 1st of September 2008 to the 4th of September 2008, the file size was 1Mb, see Table 3.3 for statistics. The raw web log file was in a standard CLF format and example of line record is displayed below:

207.200.81.166 - - [01/Sep/2008:08:29:22 +0300] "GET /files/Tyrimas_Lietuviu_emigracija_Studija.pdf HTTP/1.0" 200 32768 "http://directory.mozilla.org" "Robozilla/1.0"

Table 3.3 Data statistics

Web site	Period	No of records in web server
www.civitas.lt	01.09.2008-04.09.2009 (inclusive)	5 973

To evaluate how good proposed method works, we have to compare 3 data sets:

1. file with pages generated by the web server and called *Log_link*;
2. file with pages generated by the special script which automatically retrieves all links displayed on the web page (proposed by author "Link based" method) and called *Auto_link*;
3. file with web pages which was created manually and called *Manual_link*. All available pages and links were tested by manual mouse click operation and then placed into the file. This manually created file was used as a benchmark for evaluation study. Ideally this file should contain only accessible by a mouse click files;
4. only unique (distinct) pages were picked from those three data sets. No of records in all those files presented in Table 3.4.

Table 3.4 No of unique(distinct) URLs in Log_link, Auto_link and Manual_link tables

Log_link	Auto_link	Manual_link
465	499	181

To evaluate the model, the following procedure was followed:

Standard cleaning method was applied to the Log_link (this file was generated by the web server). Standard means that files having extensions gif,

jpg, css, etc are removed. After cleaning Log_link file by the standard technique, 399 records were left. See Table 3.5, first column.

In order to test how many records left after applying “Link_based” method, we have to compare Log_link with Auto_link. Records which appear in both files were selected. See Table 3.5, second column.

Finally web log file Log_link was compared to manually selected file Manual_link, as the later one served as a benchmark. See Table 3.5, third column.

Table 3.5 No of records after cleaning

Standard method	“Link_based” method	Manually selected
399	295	157

Statistical measures – type I errors and type II errors where tested to compare both methods. These errors often referred to as false positives and false negatives respectively (Allchin 2001; Lup Low et al. 2001).

Type I errors (the “false positive”): the error of rejecting the null hypothesis given that it is actually true; e. g., Doctor determines that a person is ill when actually he is not.

Type II errors (the "false negative"): the error of failing to reject the null hypothesis given that the alternative hypothesis is actually true; e. g., Doctor determines that a person is healthy when he actually is sick.

In other words, these errors are called false positive and false negative and can be described in the way which is described below.

The false positive rate is the proportion of negative instances that were erroneously reported as being positive and estimated as:

$$\text{false positive rate} = \frac{\text{number of false positives}}{\text{total number of negative instances}}. \qquad (3.1)$$

The false negative rate is the proportion of positive instances that were erroneously reported as negative and estimated as:

$$\text{false negative rate} = \frac{\text{number of false negative}}{\text{total number positive instances}}. \qquad (3.2)$$

The effectiveness of the cleaning strategy can be measured by the degree by which data quality is improved through the cleaning process. Therefore we measured the total error how many incorrect instances (False) were recognised as good (Positive) and good instances (True) recognised as bad (False), see Table 3.6.

Table 3.6 Cleaning Evaluation

	"Link_based" method		Standard method	
	Positive	Negative	Positive	Negative
True	151	6	157	0
False	144	Not calculated	242	Not calculated

Because we are interested in both type errors, False Positive and True Negative were accumulated and divided by the total number of records:

$$Error_{\text{total error}} = \frac{\text{False positive + True negative}}{\text{total no of records}} \times 100\%, \qquad (3.3)$$

Total error with automatic "Link_based" method:

$$Error_{auto} = \frac{144+6}{465} \times 100\% = 0.32 \times 100\% = 32\%, \qquad (3.4)$$

Total error with standard method:

$$Error_{std} = \frac{242+0}{465} \times 100\% = 0.52 \times 100\% = 52\%. \qquad (3.5)$$

From the error rate calculated above, "link" (auto) based method produced an error rate of 32% compare to the standard (gif, jpg, css, etc.) which showed 52% of mistakes.

3.5. Limitations

The proposed web log data cleaning methodology has several limitations:

MULTINATIONAL DISTRIBUTED WEB SITES

With the growth of internet, many web sites become multi-national. It means that the same web site is hosted in different regions or countries. For example, information about the products hosted in one region/country, information about company's internal structure, contacts hosted in another region/country. It becomes very difficult to trace single user and his browsing patterns.

To tackle this problem, web site maintainers develop special purpose scripts which are part of every web page. These scripts are executed on page load and gather very accurate information about the user. They place usage information straight into the database. If these scripts are used, the proposed cleaning methodology becomes absolute. However, the benefit of proposed method is that nothing has to be done to the web site prior to the analysis.

DATA NEWNESS

Data cleaning using this methodology works on the conditionally new web logs. It means that web pages in web log must exist on the web site. If the web site is updated, web logs from newly updated web do not match files logged in web log. In these situations the correct cleaning is not guaranteed.

PARSING WEB PAGES

It is pretty easy to parse HTML file and retrieve contained links. However, as the web page design technologies are evolving, more and more enhancements are introduced. In some cases these are various scripts as JavaScript and VB script in other cases there are ActiveX plug-ins such as macromedia flash, etc. It becomes more and more difficult to retrieve links from a web page as sometimes HTML in the log contains the minor part.

GOOD INTERNET CONNECTION

Proposed technique requires Internet connection, so the web pages can be downloaded. This initiates following problems.

(1) Internet traffic and storage problem. The amount of Internet traffic generated by the retrieving engine can be relatively big. Additionally, if downloaded pages are stored locally, the amount of space used can be huge as well.

(2) Login and password problem. Web sites, which require user to login, are problematic even if login and password is known. The analysis engine should be able to log in automatically. This problem was noted in one of the real world datasets analysed. Luckily, it was solved in a pretty easy way. In majority of web sites authentication is performed using cookies. These cookies are stored in a specific browsers directory and each time page is requested it is sent to the user. The problem can be solved in the following way. First, user using web browser logs into web site. Browser stores the cookie in specific directory. Next, user launches analysis engine. Analysis engine picks up cookie file from the browser, parses it and uses cookies in all its requests.

(3) Special purpose web pages. There can be some special web pages that are designed for internal/administrator usage. They are not for public use. Like other pages, requests of these pages are logged into the log file. While retrieving such a page some side effects can happen, since data analyst and Webmaster usually are different people or even different organizations and not necessary communicate very well in the real world. For example, loading some purpose page while running the cleaning engine can initiate database cleanup /admin/cleandatabase.php, send e-mails to all registered users or other side effects.

3.6. Summary of the Third Chapter

1. Developed a new original and dynamic web log cleaning method to remove irrelevant pages/records from the data automatically. Meanwhile, standard and widely used methods perform static cleaning which needs to be modified every time the web site structure changes and is not very efficient.
2. The developed special data cleaning engine performs the following tasks: downloads web pages, analyses them and finds all links user can click on. Later these links (user can "click on") are used as the only "good" links and all the other ones are removed.
3. Proposed method works better than standard techniques. This claim is supported by a comparative study evaluating new method "to ideal" (hand cleaned) and standard method used by majority of the research community.
4. Performed analysis and evaluation of the link based methodology concludes the following:
    a. new method reduces number of records,
    b. new method increases the quality of the results in the analysis stage.

4

Decision Trees for Web Log Mining

4.1. Introduction

In this study, decision trees were proposed for web log mining. They identify usage path easily and produce human understandable outputs (Chi 2002). Number of decision tree approaches exists. In this case study the C4.5 (Quinlan 1993) was selected as it is fast, powerful, free, easy to use and used by many research and commercial institutions.

Prior to the attempt using decision trees, number of other classifies were tested and the work can be found in the study (Raudys et al. 2004). In the standard way every expert is trained on the whole data set. Then experts' decisions are given to the fusion rule (e. g. majority voting) which selects the best output. Our experience showed that "in many problems" the outputs were linearly separable. In a part of the applied problems a certain "nonlinear intersection" of the pattern classes was traced.

In order to research possibilities to diminish classification error rate in the study (Raudys et al. 2004) we proposed to use different classification structure in order to obtain non-linearly separable outputs. Original data space was transformed into 2 dimensional feature space (number of methods can be utilized, e. g. principal components, auto associative neural networks tec.) see (Fukunaga. K 1990) for description. Then this space was divided by the operator manually

(but simple cluster analysis algorithm can also be used) into non-intersection regions and assigned a decision making process to an expert-elementary, the first stage classification rule (fusion of expert decisions). Each expert was trained on each separate region data set. Training of the fusion rule was performed using various classifier combining strategies including the voting, multi layer perceptron and linear trainable fusion realized by the single layer perceptron.

In the recognition stage, decision is done according to the best output produced by all experts. As this strategy depends on the data structure, on how the data is transformed into 2 dimensional space and on the way how regions are assigned, the output performed better on some data types. In several real world pattern classification task this approach appeared promising. Using web usage data methodology produced results comparable with decision tree classifier. Another point was that the output wasn't as easy interpreted by users as decision trees. See (Raudys et al. 2004) for detailed description of the new rule strategy.

4.2. Decision Trees

Decision trees are powerful and popular tools for classification and prediction. The attractiveness of decision trees is due to the fact that, in contrast to neural networks, decision trees represent rules. Rules can readily be expressed so that humans can understand them or even directly used in a database access language like SQL so that records falling into a particular category may be retrieved.

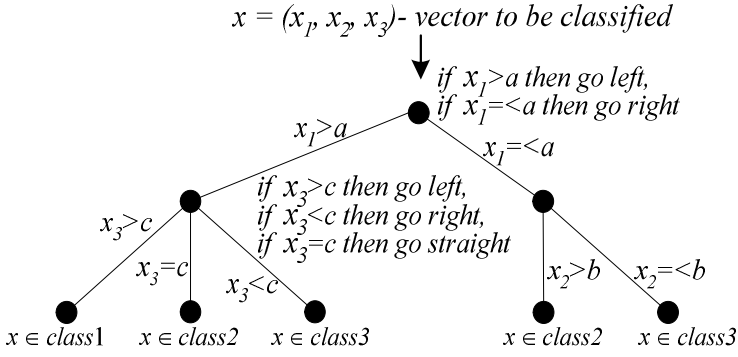

Fig 4.1. Example of the decision tree model

In some applications, the accuracy of a classification or prediction is the only thing that matters. In such situations we do not necessarily care how or why the model works. In other situations, the ability to explain the reason for a decision is

crucial. There are a variety of algorithms for building decision trees that share the desirable quality of interpretability. A well known and frequently used over the years is C4.5 or improved, but commercial version See5/C5.0 (Data_Mining_Tools).

Decision trees find sets of rules that classify data item into one of a set of the predefined classes. Traditionally, trees are drawn with the root at the top and the leaves at the bottom (see Fig 4.1). The path from the root to the leave represents possible ways of assigning records and can be expressed in a way of a rule. Each leaf in the tree defines a class.

4.3. C4.5 Algorithm

C4.5 builds decision trees from the set of training data in the same way as its predecessor ID3 (Mitchell 1997). It uses the concept of information entropy. Each attribute of the data can be used to make a decision that splits the data into smaller subsets. C4.5 examines the normalized information gain (entropy difference) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision. The algorithm then recurs on the smaller sub-lists.

This algorithm has a few base cases, the most common base case is when all the samples in your list belong to the same class. Once this happens, you simply create a leaf node for your decision tree telling you to choose that class. It might also happen that none of the features give you any information gain, in this case C4.5 creates a decision node higher up the tree using the expected value of the class. It also might happen that you've never seen any instances of a class; again, C4.5 creates a decision node higher up the tree using expected value. For more detailed description please refer to (Quinlan 1993) and (Quinlan 1996).

4.4. Data Description

For experimental study, log files from online journal, running in Germany were taken. Web site presents different social/cultural/life activities in a Munich city where visitors can browse for specific information on tourism, education, religion, past, current and forthcoming events (e. g. concerts, exhibitions), get their descriptions and location. Each log file represented a different summer month: June 31MB 2000, July 45MB 2000, August 42MB 2000.

Table 4.1 Data description

Period	Size	Web site name
01.06.2000 – 31.08.2000	118 Mb (June 330897, July 457732, August 426627)	http://www.munichfound.de/ online journal about different events in Munich, Germany

4.5. Data Construction

As known, user browsing sessions can have different length (number of pages viewed by the user). However, C4.5 can process only on fixed length records. In order to decide how many page history to use, experimental study using different length history (2,3,4,5,6,7,8 pages) was performed and it was discovered that in most of the cases the algorithm constructs rules using up to four pages only. There were just several cases containing longer web pages paths. Therefore, sessions with less than four transactions were removed. From remaining sessions data using 4 page windows was constructed (see Fig 4.2).

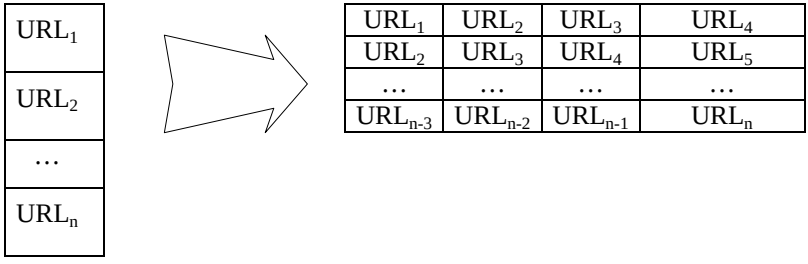

Fig 4.2. URLs in one session before and after mapping

As one of the tasks, the effort to predict 4th page based on previous 3 was made. Decision trees are designed to work with reasonably small amount of classes. In this problem, the number of classes must be equal to number of pages. Number of pages is quite large so it is not possible to use each page as a separate class. Therefore, some pages were grouped. The structure of the web site is quite simple and pages "naturally" group. There are template pages, and depending on arguments different content is displayed. For example *location* template depending on arguments generates description of different locations. The same is with the other pages. They are formed under templates like *restaurants*, *events*, *leisure*, *museums* and etc.

Finally, to avoid fitting/overtraining effect, data was divided into training and testing datasets. 50% of records were used for training and 50% of records were used for testing.

Table 4.2 Final data description

label	name	test	train	Total
a	/onelocation	10691	10462	21153
b	/events	3681	3673	7354
c	/classifieds	5219	5189	10408
d	/leisure	5314	5266	10580
e	/restaurants	3753	3749	7502
f	/cinema	3608	3582	7190
g	/jobmarket	2044	2181	4225
h	/dininganddancing	3592	3804	7396
i	/sports	1948	1998	3946
j	/museum	2901	2846	5747
	TOAL:	42751	42750	85501

4.6. Problems

4.6.1. Problem#1

As it was mentioned in previous section, first task of the exercise is to predict 4th page based on the previous 3.

Data preparation. Experiments were performed to deliver complex, four URL based rules having following syntax:

if S(i)=URL1 **and** S(i+1)=URL2 **and** S(i+2)=URL3
    then S(i+3)=URL4
endif

where `S(i)` stands for *i* transaction in the session, `S(i+1)` stands for the next (*i+1*) transaction and etc. Users which have more than three pages in the session have been mapped using 4 page window described in Table 4.2. Data description summary is presented in table above.

Results. Table 4.4 presents error matrix. First 10 rows are for training results and last 10 rows for testing. The average error was of 6.7% for training and of 6.9% for test data. In other words, knowing three user steps the forth step can be predicted with 93.1% accuracy.

Table 4.3 Examples of rules generated by C4.5

Rule No.	Confidence (%)	Support (%)	Sequential rules
1	95	0.05	/dininganddancing.cfm /nightlife.cfm /location.cfm ⇒ /onelocation.cfm
2	87	0.15%	/events.cfm /jobmarket.cfm /classifieds.cfm ⇒ /oneclass.cfm

Table 4.4 Output matrix was generated by training on 42751 cases and testing on 42750 records. Letters a-j stands for different groups of pages

Training										
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	classified
9943	5		285	204	1	41	498	196	15	(a): class
5	3554		5	5	3	35	7	2	13	(b): class
7		5156	86		3	5		1	1	(c): class
55	3	16	4790	1	46	4	3	2	3	(d): class
73	1			3432		12	1	1	3	(e): class
5	3	41	76		3553	4	2	1	2	(f): class
56	92	3	39	32		1866	43	20	107	(g): class
447	11		9	29		36	3026	1	14	(h): class
94	2	2	10	48	1	15	7	1721	6	(i): class
6	10	1	14	2	1	26	5	3	2737	(j): class
Test										
9685	6		284	190	1	46	550	220	5	(a): class
3	3531		9	3	4	36	3	4	10	(b): class
3		5118	85		2	1	1	1	2	(c): class
62	1	10	4752	1	36	5	1	6	2	(d): class
88	7		2	3460		26	2		3	(e): class
3	3	56	74		3535	3		2	1	(f): class
54	105	3	36	25	2	1977	46	10	122	(g): class
467	12		6	28		50	3194	1	6	(h): class
91	3	2	4	42	1	12	3	1749	3	(i): class
6	5		14		1	25	4	5	2692	(j): class

The C4.5 has an advantage over the other decision tree approaches, that it can generate rules and show dependencies between different items. Table 4.3 presents some examples of the rules discovered. For example, the first rule shows that 95% of visitors interested in /dininganddancing.cfm, /nightlife.cfm and /location.cfm were also interested in /onelocation.cfm. In addition to this, these visitors make 0.05% of all transactions. The second rule shows that 87% of visitors interested in /events.cfm, /jobmarket.cfm, /classifieds.cfm were also interested in /oneclass.cfm and this combination covers 0.15% of all transactions.

4.6.2. Problem#2

Analysis of the web log showed that there are number of users, terminating web site browsing after four steps. Understanding why it is happening can be quite useful. It may be worth putting the most valuable information into those pages to encourage user to keep browsing. Or it may be worth to analyse the contents of these pages and decide whether browsing termination occurs "naturally" or because of some reason. Therefore, it was decided to perform experiments and try to discover some browsing termination roles. Two classes were formed – users who terminated browsing after four pages and those who were continuing browsing.

Data preparation. Based on the length of the session, data was divided into two classes (see Fig 4.3). Class1 was made up of users with just four hits per session. Class2 contained users with more than four hits per session. Division formed 880x4 and 39991x4 data matrices respectively. Prepared data was divided into training and testing as described earlier.

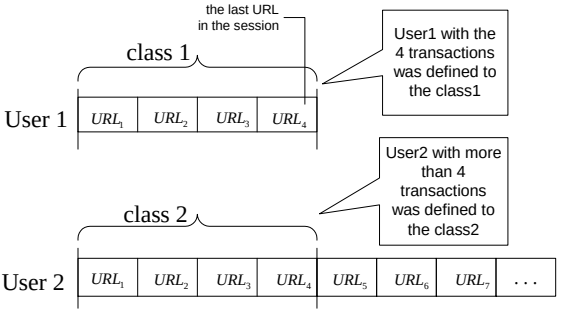

Fig 4.3. Data division into classes. Class1 – users with four hits per session. Class 2 – users with more than four hits per session

Results. Experiments showed that using decision tree approach, pages leading to session termination can be predicted with a reasonable accuracy. The

training set showed 10.5% and test set showed 12% average misclassification error.

An example of the generated rule is:

if S1 = /classifieds.cfm **and** S2 = /oneclass.cfm **and**
S3 =/events.cfm **and** S4=/todayall.cfm
then *finish*
endif.

This means that in 2.27% of cases users will finish browsing if given conditions are satisfied. This rule covers rather large number of cases and web site developers should pay attention to this information. It can be reasonable to put important information (main news, services, events, advertisements) to these pages, to encourage users not to stop terminating browsing.

4.6.3. Problem#3

Problem#3 is similar and at the same time different to Problem#2. Similar, because the aim of the experiment was to predict any set of four pages leading to browsing termination. Different, because last 4 pages of any session is predicted. Data construction process is presented in Fig 4.4. Determining harmful pages (after which users are likely to terminate their browsing) is the same important as to know valuable web pages (frequently viewed). This can help to answer the question "why do visitors finish browsing?" If there are large numbers of such users, it can reasonably be inferred that interesting and "sticky" information should be added into these pages. In addition, it would be useful to determine if users finish browsing "naturally" by obtaining information they need or do they simply lose the interest.

Data preparation. Experiments were performed to find combinations of pages after which users are likely (or not likely) to finish browsing. Two classes were formed. One class was made up of four transactions occurring at the end of the session (see Fig 4.4). The second class was created out of all the other transactions. This respectively formed two classes with 7258 (who finish browsing after four pages) and 124059 (who continue browsing after four pages) vectors. The prepared data was divided into training and testing according earlier described conditions (50% for training and 50% for testing).

Results. Experiments produced 6% errors in the training data and 6% errors in the test dataset. An example of the generated rule is:

if S(i)=/events.cfm **and** S(i+1)=/onelocation.cfm **and**
S(i+2)=/leisure.cfm **and** S(i+3)=/kids.cfm
then finish
endif.

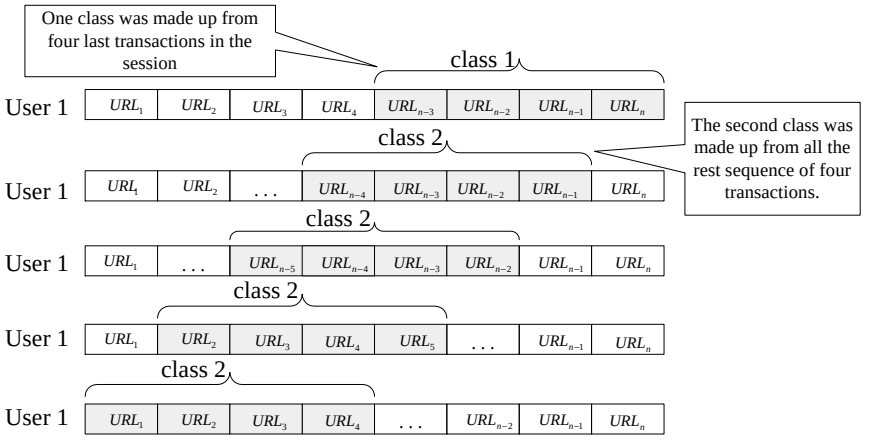

Fig 4.4. Data is divided into two classes. One class is made out of four last transactions in a session. Another class is made out of all the other transactions

This rule covers 0.26% of all transactions and shows sequence of web pages after which visitors "naturally" or by some reason are likely to terminate browsing process. Conversely, pages not having such sequence of pages are not likely to finish browsing. Such interesting combination of pages can be analysed by site developers and reasons of such user's behaviour can be investigated.

4.7. Summary of the Fourth Chapter

Research study was performed to investigate how decision trees are suitable for web log data analysis. At the time, there was no comprehensive study on the subject. There were no guidelines how to prepare data and what kind of knowledge it is possible to obtain using decision trees. It is the first empirical evaluation on how decision trees perform data mining tasks using web log data. Summary of this research can be presented in following points:

1. It was determined that a specific data preparation is necessary in order to perform web log mining using decision trees. User sessions can have variable length –but decision tree C4.5 supports only fixed length vectors. It was noted that special data preparation is required for every different task examined.

2. Trying to predict each web page as a class using C4.5 is not possible. The number of different pages was huge, it was concluded that one needs to group together some pages according to the topic. In this particular example 10 class were constructed.

3. Investigated 3 hypotheses what could help to improve web site user's retention:
    a. On the basis of three viewed pages predict the fourth page to be viewed;
    b. Initial four pages resulting in user terminating browsing;
    c. Last four pages of the session leading to browsing termination.
4. Experiments showed that in all three hypotheses C4.5 algorithm can be successfully used. Each hypothesis represents major question which are interesting to analyst or web site developer analysing web site usage. The confidence of each produced rule by algorithm is measured by the error rate how many are classified incorrectly.
5. Few examples where presented of the discovered rules and one can state that decision rules provide understandable interpretation with reasonable accuracy.

5

Text in Web Log Mining

5.1. Introduction

Up to now, most of the research in web usage mining focused on knowledge discovery to uncover user navigational patterns. These are important issues for successful e-business. To improve the quality of web site browsing, attention is paid to web usage features such as location of pages, discovering sequential patterns in web site access and the frequency of web page access. However, features such as these together with text information retrieved from links have not been analysed. In this chapter we propose text usage together with web log data for mining users' navigational patterns. The proposed method combines web pages together with hyperlink text. The hypothesis behind this is that text influences visitors' decision to browse some other page or terminate web site browsing. Number of experiments showed an increase in accuracy while performing different web mining tasks. For example, predicting next user step knowing past browsing patterns (pages) and text on these pages. The developed engine extracted the text from links in the web page. This text is parsed and used as additional features. The results were published in (Pabarskaite et al. 2002).

Information visualization is an important part of knowledge discovery process introduced by (Fayyad 1996). However, it is important to present the

Information visualization is an important part of knowledge discovery process introduced by (Fayyad 1996). However, it is important to present the outcome in the understandable format. This process is the stage when results of the mining process are presented in a form most readily understood by the end user/business analyst (Fayyad et al. 1996).

Technical experts are comfortable with knowledge discovered from web pages in technical terms. For example, a list of five most common pages through a web site being discovered might be presented in the form of URLs. The end user might find this technical form of representing this useful information baffling. A non-technical end-user would likely find more useful the most common path represented in a form of page names such as Homepage instead of index.html and TOPIC search page instead of /xxx.html.

This is a problem addressed in the second part of this chapter. The proposed solution will be developed and discussed.

5.2. Using Text together with Web Log Data

To solve certain tasks scientist used text together with web pages as additional features (Mobasher et al. 2000; Dai et al. 2003). Authors showed the quality of web mining results increases using text together with the web pages. This is important for web usage tasks when small amount of data is available or when the web site changes regularly. Authors performed experiments constructing different data sets. In the first data set, features were web pages (urls). In the second data set features contained just text from HTML. Third type of data for experiments contained web usage and text. The last type of data set separated users with similar navigational patterns best. It appears that using context data, it is possible to separate different user groups even thought their navigational patterns match. This combined approach is very efficient for recommendation systems because additional historical information is essential for this type of task.

5.2.1. Combining text with web usage information

Here we propose using text as additional features for mining navigational patterns as meaning of the text might influence web site visitors' choices. The text used is the same one which is displayed on the real web site. So detailed steps will be provided in order to show how proposed method works.

The first step when using web pages and text is to associate each URL with the text. For this purpose the following steps have to be done:

- The contents of each page is downloaded;
- The downloaded context is analysed to extract text tags.

Pages without text tags have to be eliminated as they do not contain textual information which is necessary to perform this task. When textual information is retrieved, text is mapped into the individual words list.

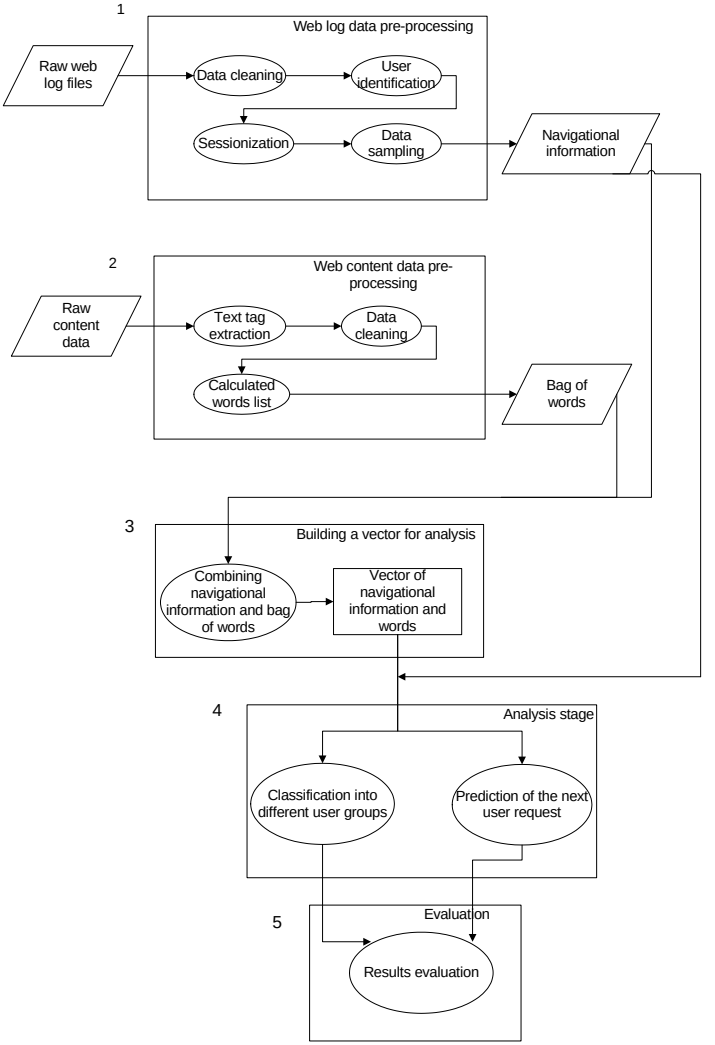

Fig 5.1. Combining link text and web log usage analyzing users behaviour. 1–5 indicates sequence steps.

- Articles (a, the), prepositions (for, in, on and etc), question words (how, why and etc) or other words or expressions like “click here” are removed.
- Cleaned words are counted, i. e., how many times each word appeared in the text tags. Words having the higher frequency are taken into account.

Web log data should be prepared according to the steps discussed in Chapter 1. These steps are: data cleaning, user identification, sessions’ creation, data sampling. The final data preparation stage involves mapping session’s URLs and words into the fixed length vectors Fig 5.1, step 3. One vector represents one session. Multidimensional vectors consisted of URLs and words. Web pages and words are mapped into semi-binary features. For example, if the page or word does not appear in the session, zero is assigned. If the page has been viewed during that session, the existed number is incremented by one. The detailed schema of the whole process is depicted in the Fig 5.1.

5.2.2. Experiments

A dynamic fast growing advertising brokering company located in UK provided web logs for the experiments. The web site contains information about services available through the company. It covers general information about the company, clients, partners, different media planning projects and careers opportunities etc.

The web log file contained two month records collected from 14th of September 2001 to 16th of November 2001 and was partially cleaned. Image request records had already been removed from the dataset. Each row in the dataset represented a web page access transaction viewed by a visitor. The data format was not in a standard format. The dataset was extracted from the database generated by Webtrends (web traffic analysis software package for web administrators and marketing managers). Traditional log fields such as IP, GMT zone, request (GET, POST), http version, server request code and requested file size had already been removed. Data fields such as page name and time of hit were present in the original form. To compensate some important fields, three new ones were added. They were user identification, user type and session id. These new types were created using internal companies cookie records about registered and guests visitors. The data contained two types of web site visitors. Every visitor can become a so called *guest user*. They cannot access some restricted pages where authorised users can get in. The second type is *registered users* who pay fees and are members of that particular web site. They are the most valuable web site visitors.

One aim is to turn more guest users into registered ones.

How this can be done? The answer can be to uncover hidden behaviour of different user groups. This hidden behaviour can be employed persuading visitors guest to become registered visitors.

The *first hypothesis* in pursuing this goal is that different users groups, visitors and registered, can be classified with the significant accuracy. This knowledge of typical guest or registered browsing behaviour makes it possible to identify an unclassified log record and assign it to one of predefined used groups.

The *second hypothesis* is predicting the browsing behaviour of different visitors/guests groups using historical browsing data. For example, does the visitor come back into the web site? This particular question should be interesting for companies offering different services to people. If a particular visitor is likely to enter the web site for the second time, it is probably worth to send him/her a commercial offer?

To prepare the data for experiments, each URL was associated with a particular piece of text. All URLs, existing in the web log, were downloaded. Downloaded pages contents in HTML format were analysed and text link tags extracted.

For example, the following piece of HTML code was analysed:

Excellence in CPD Awards 2002. Both the page (/awards/cpd_awards/guidetoentrants.html) and a text tag (Excellence in CPD Awards 2002) were recorded into the database. Table 5.1 shows examples of relations between URLs and text representations.

Table 5.1 URL links related to text tags

Web pages	Text
/content.cfm	Home
/careers/scrollnews2.cfm	CAREERS
/awards/cpd_awards/guidetoentrants.html	Excellence in CPD Awards 2002
/ipacareers_html/intro.htm	Agency Factfile
/services/training/CVposting/searchCVhelp.htm	Help on how to search for CVs

Afterwards, text was mapped into the individual word list. Words like articles (a, the), prepositions (for, in, on and etc), question words (how, why and etc) or other words or expressions like “click here” were filtered out (see Table 5.2).

In the next step, mapped words were counted i.e. how many times each word appeared in the text tags. Words having the higher frequency of appearance were taken into account. For experimental purposes, 121 words appearing more than 4 times were selected.

Table 5.2 Words are mapped into the individual words list

Text	Generated Word List
Home	home, careers, agency, factfile, help, search, cvs
CAREERS	
Agency Factfile	
Help on how to search for CVs	

The final data preparation stage involved mapping session's URLs and words into fixed length vectors. Multidimensional vectors were constructed of the following features:

- the number of pages viewed per session;
- session duration information;
- URLs;
- words.

The most viewed web pages (119) and the most often occurred words (126) were mapped into semi-binary features. For example, if the page or word does not appear in the session, zero is assigned. If the page has been viewed during that session, the existed number is incremented by one. (...0,0,1,0,0,3,0...) indicates that word/URL x appeared once and word/URL y appeared 3 times in one user browsing session. So final vectors looked like this:

Only URL (standard data):
[(no viewed pages)(session duration)(URLs)]
[(14),(12min),(...0,0,0,1,2,0,1,0...)]
1+1+119=121 features

URLs + words (enhanced data):
[(no viewed pages)(session duration)(URLs)(words)]
[(14),(12min),(...0,0,0,1,2,0,1,0...)(...0,2,0,1,0,0,1,...)]
1+1+119+126=247 features

For **hypothesis #1,** final dataset consisted of 8160 guest and 8408 registered users with 247 features/dimensions in each vector.

As it was mentioned earlier, the aim of the **hypothesis #2** is to predict if the user will return to the site. The pre-processed web log data consisted of 12312 sessions made by users who enter the site more than once and 4255 sessions made by visitors having only one session in the dataset.

For experimental study 3 classification algorithms where selected. Please note, that algorithm selection is not very important in this study. The aim of the

experiments is to demonstrate that adding word usage information to data allows to improve classification accuracy. List of algorithms:

- SLP – single layer perceptron classifier. It is linear classifier separates classes by the single line. It usually trained by gradient descent method.
- MLP – multi layer perceptron classifier. It is nonlinear classification algorithm capable to separate pattern classes with more complex than line decision boundaries (Raudys 2001).
- *k*-NN – *k*-nearest neighbour classifier. Simple nonlinear classifier however very slow. It is quite often used as a benchmark classifier (Berry et al. 1997).

Data and algorithms were prepared in the following way. First, in order to avoid overtraining, the data was divided to training and test sets (Raudys 2001). For SLP and MLP vectors were randomly divided into 3 equal parts: 33.3% (⅓) for training, 33.3% for validation and 33.3% for testing. For k-NN algorithm data was divided into 66.6% (⅔) for training and 33.3% for testing. *Second*, it is known, that SLP, MLP and *k*-NN produces more reliable results if data is normalised. Scaling gives equal opportunities for each feature to influence the final classification result. For this reason, each feature was scaled to make its mean to 0 and standard deviation to 1. *Third*, the MLP architecture used was rather simple. MLP had 3 layers, with 3 neurons in the hidden layer. Number of input and output neurons was selected according to the dataset. MLP and SLP experiments were repeated 5 times, and the final error was the mean of all errors from the 5 experiments. *Forth*, the *k*-NN algorithm had only 1 neighbour (1-NN). The reasons were these: a), one neighbour architecture is the simplest and easiest to understand and using more neighbours does not necessarily leads to the better classification accuracy. b) as more nearest neighbours *k*-NN has, the slower it is. Event with 1 neighbour run time of *k*-NN was approximately 2 hours.

Hypothesis #1. As it can be seen from the results (see Table 5.3), combined dataset increases all classifiers accuracy. The accuracy increases: for MLP by 2.6%, for SLP by 3.13% and for *k*-NN by 2.4%. An unexpected result was that SLP outperformed MLP. This means that the first hypotheses data is linearly separable.

In order to verify potential limits of the SLP based classifier we estimated a resubstitution error (apparent error rate obtained while classifying the training set vectors. Resubstitution error indicates how good or bad results are on the training set. The resubstitution error rate was 14.46. From well known symmetry of the resubstitution and expected classification errors (see e. g. (Raudys 2001), Section 6.3), the asymptotic classification error rate is approximately (14.5+17.6)/2≈16%. This estimate advocates that the SLP classification error rate could be improved only negligibly by 16%. We believe that the result could be improved if a

specific group of users would be considered. In such a case, we would have more homogenous group and "narrower dependences" between the outputs and inputs.

Hypothesis #2. As it can be seen from the results, combing URL and word features, the classification accuracy increases from 0.7 to 3.5 %. The accuracy increases: for MLP by 3.5%, for SLP by 1.4% and for *k*-NN by 0.8%.

Evaluation of resubstitution error for hypothesis #2 is (16.3+22.6)/2≈19%.

Table 5.3 MLP, SLP and k-NN performance on URL and URL + words datasets. Mean (over 5 experiments) error is presented in percents on the test set. The error's standard deviation is presented in brackets. The lower the number the better the result.

	MLP with URL	MLP with URL+Wrd	SLP with URL	SLP URL+Wrd	*k*-NN with URL	*k*-NN URL+Wrd
H#1	20.78(0.64)	**18.18**(0.27)	20.69(0.38)	**17.56**(0.26)	41.25	**38.84**
H#2	23.63(0.78)	**20.10**(0.54)	24.02(0.52)	**22.63**(1.89)	28.26	**27.50**

5.2.3. Evaluation

A point estimate is the "best" in a sense of the test set, In cross-validation error rate estimation the test set is of fixed size. In principle a large number of almost equally good estimates are possible; therefore, the test set estimate is a random one. In principle, perhaps this estimate was far from the best. Therefore interval of confidence estimates provides this sort of information. Instead of a single number, an interval is used which specifies a degree of confidence that this interval contains the unknown parameter. This interval called confidence interval and the upper and lower limits of the interval are called confidence limits (Hand et al. 2001). Confidence interval can be used to describe how reliable survey results are. Normal approximation was used to estimate the interval; it means that 95% of the probability lies within 1.96 standard deviations of the mean:

$$P \pm 1.96 \times \sqrt{\frac{P \times (1-P)}{n}}, \tag{5.1}$$

where P – estimate of the class error rate, n – test sample size. For the simplicity, very often one uses *rough calculations*; therefore do not multiply by the 1.96.

Confidence interval for hypothesis 1, where mean error is the lowest with SLP $17.56 \approx 18$:

$$17.56 \pm \sqrt{\frac{0.18 \times (1-0.18)}{27613}} = 17.56 \pm 2.29e^{-3} \times 100\% = 17.56 \pm 0.23\%,$$

Thus, 95% that the error rate falls into the interval $17.56 \pm 0.23\%$.

Confidence interval for hypothesis 2, where mean error is the lowest with MLP $20.10 \approx 20$:

$$20.1 \pm \sqrt{\frac{0.2 \times (1-0.2)}{27612}} = 20.1 \pm 2.41e^{-3} \times 100\% = 20.1 \pm 0.24\%,$$

Thus, 95% that the error rate falls into the interval $20.1 \pm 0.24\%$.

5.2.4. Limitations

Since text tags have to be included into the analysis process, there are some limitations of this method:

- if text tag does not exist, the web page is removed. By doing this, some valuable information can be lost;
- if instead of the text tag, reference to the link is indicated by the image or Java script.

5.3. Text for Results Presentation

5.3.1. Description of the Process

"The essence of Information Visualization is referred to the creation of an internal model or image in the mind of a user. Hence, information visualization is an activity that humankind is engaged in all the time" (Banissi 2000). The

knowledge discovery steps described by (Fayyad et al. 1996) does not involve information visualization as a part of the pattern discovery cycle. However, it is important to display the outcome in the understandable format (Hunphreys 1992; Chi 2002). This process called pattern analysis stage – when results of the mining process are presented in a form most readily understood by the end user/business analyst (Pabarskaite 2002).

Technical experts are comfortable with the knowledge discovered from web pages in technical terms. For example, a list of five most common pages through a web site being discovered might be presented in the form of URLs. The end user might find this technical form baffling. A non-technical end-user would likely find more useful the most common path represented in a form of page names such as Homepage instead of index.html and TOPIC search page instead of /xxx.html.

Text tags in HTML links carry valuable information. They can be used to replace unfriendly URLs. This is applicable in any application where URL is presented to the end user. This is often a case in web log mining. In novel approach all unfriendly links can be replaced by much more meaningful text tags.

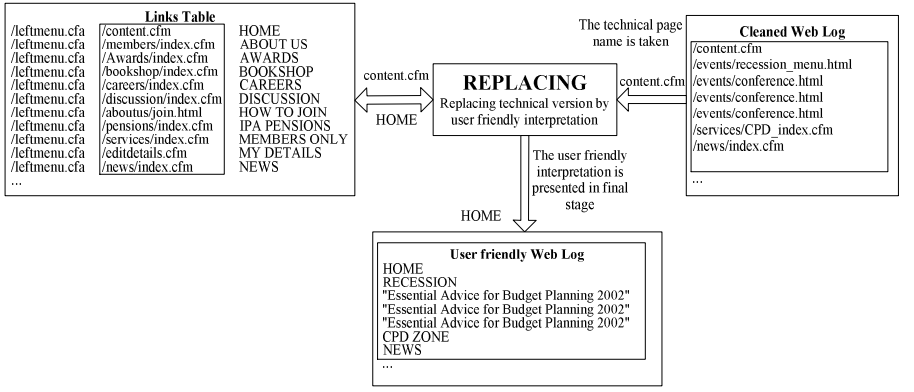

Fig 5.2. Replacing engine replaces link tags with text tags

For example, information presented for technical analyst might not be understandable by a business person. Technical interpretation of the page *index.html* is commonly understood by web site developers. The expression *index.html* might be more easily understood by a business person as *Homepage.* The implemented engine (see Fig 5.2) takes web pages from the cleaned records table called “Links” (see Chapter 3) and replaces them with text tags.

Table 5.4 Replacing technical version of web pages with user attractive interpretation

Technical version of the web page	Textual interpretation
/content.cfm	Home
/ipacareers_html/factfile/listcategories.cfm	IPA recruiting agencies
/ipacareers_html/home_maillayer.cfm	Graduate recruitment (Flash)
/ipacareers/welcome.cfm	Graduate recruitment (HTML)
/services/main.cfm	MEMBERS ONLY
/members/map/searchoptions.cfm	IPA agencies
/services/training/cpd/loghome.cfm	CPD ZONE

Table 5.4 shows example of the applied methodology. The technical version of web pages was replaced by text tags extracted from HTML code. This modification is much more attractive and allows tracing results because text which users can see on the links used for results presentation.

5.3.2. Limitations

Replacing URLs with text tags is not straightforward process. There are numerous problematic situations. For example, there can be multiple text tags or there can be no text tag at all. Some of all problematic areas are addressed in this section.

5.3.2.1. Multiple Text Tags

In certain cases, the same link in various pages can have different text tags. For example, the root page in one place can be indicated as “Home” and in another place as “Home Page”. This creates a problem witch one to use. Several solutions exist. First, any/random/first text tag can be used. Next, text tags can be combined together.

5.3.2.2. Missing Text Tags

In reverse to previous problem, text can be missing. This can be a case, where link text is an image, special script or just meaningless word such as “click here”. In image case a simple solution can exist. Sometimes (quite often), images contain tag *alt*. This is textual tool tip that is displayed if user holds mouse on top of the image. This text can be used as a replacement instead of the link text tag. However, it is often a case that developers do not use or forget to set them and *alt* tags just do not exist or contain file name of the image.

There is no good solution if text tag does not exist and there is nothing to replace it. In these cases, usually, it is represented by the same unfriendly URL. In future there are some ideas how to overcome this problem. One can use not only text tags, but neighbour text as well. I.e. cases where links are placed near

meaningful text like: ”If you are interested in cars <click here>”, select the text which comes before <click here>. From another hand there is a risk of capturing unrelated or even harmful text and displaying it instead of a link.

5.3.2.3. Not HTML based Links

If web pages are highly scripted or use a lot of ActiveX or Java Applet components, it may be difficult to parse and extract text tags. In HTML case, it is pretty straightforward, however if link is implemented in some other way, the parsing engine must be enhanced. There should be added support for technologies, targeted web site uses.

5.3.2.4. Speed

Replacing page URL with text tag is pretty straightforward process. If it is preformed in database, the URL field must be indexed. In this case matching will be performed very quickly without noticeable effect in performance. In case of programming (e. g. C++) implementation, the array of URL strings must be sorted. That allows utilising fast binary search routine. The same as in database case, the decrease in performance while changing URLs into text should be small and unnoticeable.

5.3.2.5. Multilanguage pages

Whereas data mining tasks are language independent, using text in mining language plays an important part. Currently there is no methodology created to deal with this problem. All text mining techniques are applicable to English language (Tan 1999).

5.4. Summary of the Fifth Chapter

1. Combined web log and link text mining approach is proposed. Web usage data is combined with text on links. These topics have never been analysed together, so it is absolutely a new concept. Some researchers used web usage information together using text from the page content, but nether with link information. The idea is that users do not see web content while clicking on the link to the page, he can only look at the text on the hyperlink. This text significantly influences users’ further actions. Therefore text on hyperlinks is used to improve web log mining tasks.
2. Special data preparation steps were presented.

3. Empirical study was provided in order to test the new combined approach. Two real world problems which could help to improve web site user's retention have been analysed: a. users classification into different groups according to their registration status, b. predict users behaviour based on their interest to the web site.

4. The combined approach increases classification tasks accuracy. Classifications performed with MLP, SLP and k-NN demonstrated up to 3% reduction in misclassification error using user browsing history in conjunction with the text.

5. Empirical findings advocated that web site reorganisation could be considered, in order to improve the quality of pages less attractive to the visitors and more commercially valuable information might be placed into pages which attract users' retention.

6. Proposed simple yet quite useful technique to improve the way how results can be displayed. Traditionally, results are presented by showing URLs/links to various web pages on the site. The idea is to replace this technical information with more user friendly text. This text is taken from the links. For example, /site/prod.php is replaced by "Products". Despite of its limitations, technique is quite good result presentation improvement and can be used in any web log analysis software.

General Conclusions

1. An investigative study was undertaken to determine essential characteristics of web mining. Analysis was performed on a selection of web usage tools and methodologies proposed by research community. Examination number of web usage tools allowed compare results based on a set of standard techniques and ones proposed in these thesis.

2. Introduced a new web log data cleaning method. The method performs "link based" cleaning. This enables web log data practitioners to use information which represents actual users' activity. Proposed cleaning technique reduces the number of record. Moreover, "link based" technique imitates real clicks therefore in the pattern analysis stage easier to trace visitors navigational patterns.

3. Proposed specific data preparation process in order to compose fixed length vectors. This enables executing various prediction tasks to understand users' behaviour exploiting decision tree algorithm. Decision trees generate clear logical rules, understandable to humans. Therefore data analyst can identify situations typical for browsing termination and succession, and web site designer can reorganise web site information to be more convenient for web site visitors.

4. This dissertation introduced a combined approach which takes users browsing history and text from the links text to analyse users' behaviour. Explained text retrieval process, provided steps required to combine text and

users navigational information. Experiments proved the correctness of hypothesis that text can bring more value in order to understand users' patterns. Combined features (pages + text) method increased task classification accuracy by 3.1% (error rate from 20.8% decrease to 17.6%). Combined features (pages + text) method increased the task "will user returns to the site" accuracy by 3.5% (error rate from 23.6% decrease to 20.1%). According to the findings web site reorganisation could be revised, in order to improve the quality of pages less attractive to the visitors and more commercially valuable information might be placed into pages which keep users' retention.

5. Proposed more understandable approach for displaying web log mining results. Perception and interpretation of the results becomes clearer and more attractive because they appear as a text, which users see while browsing the actual web site.

References

AccrueHitList. [accessed 2003.09.11]. Available from Internet: <http://www.accrue.com/index.html>.

Active_Server_Pages. [accessed 2009.02.19]. Available from Internet: <http://www.asp.net/>.

Adomavicius, G. 1997. Discovery of Actionable Patterns in Databases: The Action Hierarchy Approach, in *Proc. of the Knowledge discovery and data mining*. 111–114.

Afergan, M.; Darnell, R.; Farrar, B.; Jacobs, R. L.; Medinets, D.; Mullen, R.; Foghlú, M. Ó. 1996. Web Programming Desktop Reference, 1104 p.

Agrawal, R.; Imielinski, T.; Swami, A. 1993. Mining association rules between sets in large databases, in *Proc. of the Conference on Management of Data (ACM SIGMOD)*. 207–216.

Agrawal, R.; Srikant, R. 1994. Fast Algorithms for Mining Association Rules, in *Proc. of the 20th VLDB*. 487–499.

Agrawal, R.; Srikant, R. 1995. Mining Generalized Association Rules, in *Proc. of the 21st VLDB Conference*. 407–419.

Agrawal, R.; Srikant, R. 1995. Mining Sequential Patterns, in *Proc. of the Data engineering*. 3–14.

Ahonen, H.; Heinonen, O.; Klementtinen, M.; Verkamo, A. 1998. Applying data mining techniques for descriptive phrase extraction in digital document collections, in *Proc. of the Advances in Digital Libraries (ADL'98)*. 2–11.

Ahonen, H.; Heinonen, O.; Klementtinen, M.; Verkamo, A. 1999. Finding co-occurring text phrases by combining sequence and frequent set discovery, in *Proc. of the 16th International Joint Conference on Artificial Intelligence IJCAI-99, Workshop on Text Mining*. 1–9.

Allchin, D. 2001. Error Types, in *Perspectives on Science*. 9(1): 38–58.

Analog. [accessed 2008.01.18]. Available from Internet: <http://www.analog.cx/>.

Angoss. [accessed 2003.11.14]. Available from Internet: <http://www.angoss.com/angoss.html/>.

Ansari, S.; Kohavi, R.; Mason, L.; Zheng, Z. 2001. Integrating E-Commerce and Data Mining: Architecture and Challenges, in *Proc. of the Data mining*. 27–34.

Arimura Hiroki; Abe Jun-ichiro; Hiroshi, S.; Setsuo, A.; Ryoichi, F.; Shinichi, S. 2000. Text Data Mining: Discovery of Important Keywords in the Cyberspace, in *Proc. of the Kyoto International Conference on Digital Libraries*. 121–126.

Arocena, G. O.; Mendelzon, A. O. 1998. WebOQL: Restructuring Documents, Databases and Webs, in *Proc. of the Data engineering*. 24–35.

Balabanovic, M.; Shoham, Y.; Yun, Y. 1995. An Adaptive Agent for Automated Web Browsing, in *Visual Communication and Image Representation*. 6(4): 12–22.

Banissi, E. 2000. Information Visualization, in *Encyclopedia of computer science and technology*. 42(27): 93–112.

Bdchner, A. G.; Mulvenna, M. D.; Anand, S. S.; Hughes, J. G. 1999. An Internet-enabled Knowledge Discovery Process, in *Proc. of the International database conference; Heterogeneous and internet databases*. 13–30.

Berendt, B.; Mobasher, B.; Nakagawa, M.; Spiliopoulou, M. 2002. The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis, in *Proc. of the 4th WebKDD 2002 Workshop, ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD'2002)*. 159–179.

Berendt, B.; Spiliopoulou, M. 2000. Analysis of navigation behaviour in web sites integrating multiple information systems, in *VLDB*. 9(1): 56–75.

Berry, M. J. A.; Linoff, G. 1997. Data mining Techniques: For Marketing, Sales, and Customer Support, *John Wiley & Sons*. 454 p.

Boley, D.; Gini, M.; Gross, R.; Han, E. H.; Hastings, K.; Karypis, G.; Kumar, V.; Mobasher, B.; Moore, J. 1999. Partitioning-based clustering for Web document categorization, in *Decision Support Systems*. 27(3): 329–341.

Boley, D.; Han, E.; Hastings, K.; Gini, M.; Gross, R.; Karypis, G.; Kumar, V.; Mobasher, B.; Moore, J. 1997. Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering, in *Proc. of the 7th Workshop on Information Technologies and System (WITS'97)*. 25–31.

Bonchi, F.; Giannotti, F.; Gozzi, C.; Manco, G.; Nanni, M.; Pedreschi, D.; Renso, C.; Ruggieri, S. 2001. Web Log Data Warehousing and Mining for Intelligent Web Caching, in *Elsevier Science*. 39(2): 165–189.

Bright, L.; Gruser, J. R.; Raschid, L.; Vidal, M. E. 1999. A wrapper generation toolkit to specify and construct wrappers for web accessible data sources (WebSources), *Crl Publishing Ltd*. 437 p.

Brin, S.; Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine, in *Computer Networks and Isdn Systems*. 30(1–7): 107–117.

Buzikashvili, N. 2007. Sliding window technique for the web log analysis, in *Proc. of the WWW 2007*. 1213–1214.

Carriere, S. J.; Kazman, R. 1997. WebQuery: searching and visualizing the Web through connectivity, in *Computer Networks and Isdn Systems*. 29(8/13): 1257–1267.

Catledge, L. D.; Pitkow, J. E. 1995. Characterizing browsing strategies in the World-Wide Web, in *Computer Networks and Isdn Systems*. 27(6): 1065.

Chakrabarti, S. 2000. Data mining for hypertext: A tutorial survey, in *ACM SIGKDD Explorations*. 1(2): 1–11.

Chen, M. S.; Park, J. S.; Yu, P. S. 1996. Data Mining for Path Traversal Patterns in a Web Environment, in *Proc. of the Distributed computing systems*. 385–393.

Chen, Y.; Yu, Y.; Zhang, W.; Shen, J. 2008. Analyzing User Behavior History for constructing user profile, in *Proc. of the IEEE International Symposium Medicine and Education*. 343–348.

Chi, E. H. 2002. Improving Web Usability Through Visualization, in *IEEE Internet Computing*. 6(2): 64–71.

Clementine. [accessed 2005.09.03]. Available from Internet: <http://www.spss.com/clementine/>.

Colin, W. 2004. Information visualization, *Morgan Kaufmann*. 347 p.

Cooley, R.; Mobasher, B.; Srivastava, J. 1997a. Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns, in *Proc. of the IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'97)*. 2–7.

Cooley, R.; Mobasher, B.; Srivastava, J. 1997b. Web Mining: Information and Pattern Discovery on the World Wide Web, in *Proc. of the Tools with artificial intelligence*. 558–567.

Cooley, R.; Mobasher, B.; Srivastava, J. 1999. Data Preparation for Mining World Wide Web Browsing Patterns, in *Knowledge and Information Systems*. 1(1): 5–32.

Cooley, R.; Mobasher, B.; Srivastava, J. 2000. Automatic Personalization Based on Web Usage Mining, in *Proc. of the Communications of the ACM*. 142–151.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; Slattery, S. 2000. Learning to construct knowledge bases from the World Wide Web, in *Artificial Intelligence*. 118(1-2): 69–113.

Dai, H.; Luo, T.; Mobasher, B.; Nakagawa, M.; Sung, Y.; Wiltshire, J. 2000. Discovery of Aggregate Usage Profiles for Web Personalization, in *Proc. of the Mining For E-Commerce Workshop (WebKDD'2000, held in conjunction with the ACM-SIGKDD on Knowledge Discovery in Databases KDD'2000)*. 61–82.

Dai, H.; Mobasher, B. 2003. A Road map to More Effective Web Personalization: Integrating Domain Knowledge with Web Usage Mining, in *Proc. of the International Conference on Internet Computing 2003 (IC'03)*. 58–64.

Data_Mining_Tools. [accessed 2009.02.07]. Available from Internet: <http://www.rulequest.com/see5-info.html>.

DataMiningSuite. [accessed 2008.09.14]. Available from Internet: <http://www.datamining.com/dmsuite.htm>.

Deerwester, S.; Dumais, S.; Furnas, G.; Landauer, T. 1990. Indexing by latent semantic analysis, in *American Society for Information Science*. 41(6): 391–407.

Duda, R. O.; Hart, P. E.; Stork, D. G. 2000. Pattern classification, 637 p.

Dumais, S.; Platt, J.; Heckerman, D.; Sahami, M. 1998. Inductive Learning Algorithms and Representations for Text Categorization, in *Proc. of the Information and knowledge management*. 148–155.

Dyreson, C. 1997. Using an incomplete data cube as a summary data sieve, in *Bulletin of the IEEE Technical Committee on Data Engineering*. (March): 19–26.

EIAA_Mediascope. [accessed 2009.01.25]. Available from Internet: <http://www.eiaa.net/>.

Facca, F. M.; Lanzi, P. L. 2005. Mining interesting knowledge from weblogs: a survey, in *Data & Knowledge Engineering* 53(3): 225–241.

Famili, A.; Shen, W. M.; Weber, R.; Simoudis, E. 1997. Data Preprocessing and Intelligent Data Analysis, in *Intelligent Data Analysis*. 1(1): 3–23.

Faulstich, L.; Spiliopoulou, M.; Winkler, K. 1999. A Data Mining Analyzing the Navigational Behaviour of Web Users, in *Proc. of the Workshop on Machine Learning User Modeling of the ACAI'99 International Conf.* 44–49.

Faulstich, L. C.; Pohle, C.; Spiliopoulou, M. 1999. Improving the Effectiveness of a Web Site with Web Usage Mining, in *Proc. of the KDD Workshop WEBKDD'99*. 142–162.

Faulstich, L. C.; Spiliopoulou, M. 1998. WUM: A Tool for Web Utilization Analysis, in *Proc. of the EDBT Workshop WebDB'98*. 212.

Fayyad, U.; Haussler, D.; Stolorz, P. 1996. KDD for Science Data Analysis: Issues and Examples, in *Proc. of the Knowledge discovery & data mining*. 50–56.

Fayyad, U. M. 1996. Advances in knowledge discovery and data mining, *AAAI Press: MIT Press*. 543 p.

Feldman, R.; Fresko, M.; Kinar, Y.; Lindell, Y.; Lipshtat, O.; Rajman, M.; Schler, Y.; Zamir, O. 1998. Text Mining at the Term Level, in *Principles of Data Mining and Knowledge Discovery*. 1510(4): 65–73.

Fernandez, M.; Florescu, D.; Kang, J.; Levy, A. 1997. STRUDEL: A Web Site Management System, in *Proc. of the Management of data: SIGMOD 1997; proceedings ACM SIGMOD international conference on management of data*. 549–552.

Fifteen_years_of_the_web. [accessed 2009.02.02]. Available from Internet: <http://news.bbc.co.uk/2/hi/technology/5243862.stm/>.

Fleishman, G. Web Log Analysis, Who's Doing What, When? [accessed 2009.01.25]. Available from Internet: <http://www.webdeveloper.com/>.

Florescu, D.; Levy, A.; Mendelzon, A. 1998. Database Techniques for the World-Wide Web: A Survey, in *Sigmod Record*. 27(3): 59–74.

Fong, J.; Hughes, J. G.; Zhu, H. 2000. Online Web Mining Transactions Association Rules using Frame Metadata Model, in *Proc. of the First International Conference on Web Information Systems Engineering (WISE'00)*. 2121.

Frank, E.; Paynter, G. W.; Witten, I. H.; Gutwin, C.; Nevill-Manning, C. G. 1999. Domain-Specific Keyphrase Extraction, in *Proc. of the International Joint Conference on Artificial Intelligence*. 668–673.

Freitag, D. 1998. Information extraction from HTML: application of a general machine learning approach, in *Proc. of the AAAI*. 517–523.

Fukunaga. K. 1990. Introduction to Statistical Patterns Recognition, *Academic Press, NY*. 591 p.

Glassman, S. 1994. A caching relay for the World Wide Web, in *Proc. of the 1st World wide web conference*. 165-174.

Gruser, J. R.; Raschid, L.; Vidal, M. E.; Bright, L. 1998. Wrapper Generation for Web Accessible Data Sources, in *Proc. of the Cooperative information systems*. 14–23.

Han, J.; Chiang, J.; S.Chee; Chen, J.; Chen, Q.; Cheng, S.; Gong, W.; Kamber, M.; Liu, G.; Koperski, K.; Lu, Y.; Stefanovic, N.; Winstone, L.; Xia, B.; Zaiane, O. R.; Zhang, S.; Zhu, H. 1997. DBMiner: A System for Data Mining in Relational Databases and Data Warehouses, in *Proc. of the CASCON'97:Meeting of Minds*. 249–260.

Han, J.; He, Y.; Wang, K. 2000. Mining Frequent Itemsets Using Support Constraints, in *Proc. of the Very Large Data Bases (VLDB'00)*. 43–52.

Han, J.; Mortazavi-Asl, B.; Pei, J.; Zhu, H. 2000. Mining Access Pattern efficiently from Web logs, in *Proc. of the Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'00)*. 396–407.

Han, J.; Xin, M.; Zaiane, O. R. 1998. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs, in *Proc. of the Conf. on Advances in Digital Libraries*. 19–29.

Hand, D.; Mannila, H.; Smyth, P. 2001. Principles of data mining, 594 p.

Hasan, M.; Golovchinsky, G.; Noik, E.; Charoenkitkarn, N.; Chignell, M.; Mendelzon, A. O.; Modjeska, D. 1995. Browsing local and global information, in *Proc. of the CASCON, Ontario, Canada*. 27–41.

Hasan, M.; Vista, D.; Mendelzon, A. 1995. Visual web surfing with Hy+, in *Proc. of the CASCON*. 218–227.

Haykin, S. S. 1999. Neural networks: a comprehensive foundation, 591 p.

Hearst, M. A. 1999. Untangling Text Data Mining, in *Proc. of the ACL, Annual Meeting-Association for Computational Linguistics*. 3–10.

Herder O.; Obendorf H.; H., W. 2006. Data Cleaning Methods for Client and Proxy Logs, in *Proc. of the WWW Workshop Proceedings: Logging Traces of Web Activity: The Mechanics of Data Collection*. 45–57.

Hernandez, M. A.; Stolfo, S. J. 1995. The Merge/Purge Problem for Large Databases, in *Proc. of the Proceeding of the 1995 ACM SIGMOD: International Conference on Management of Data*. 127–138.

Huang, X.; Peng, F.; An, A.; Schuurmans, D. 2004. Dynamic Web log session identification with statistical language models, in *Journal of the American Society for Information Science and Technology*. 55(14): 1290–1303.

Hunphreys, G. W. 1992. Understanding vision, *Blackwell*. 472 p.

Infoshare. [accessed 2002.03.24]. Available from Internet: <www.infoshare.ltd.uk/>.

Internet_Corporation_for_Assigned_Names_and_Numbers. [accessed 2008.10.15]. Available from Internet: <http://www.icann.org/>.

Internet_Systems_Consortium. [accessed 2009.02.02]. Available from Internet: <https://www.isc.org/>.

Ivancsy, R.; Juhasz, S. 2007. Analysis of Web User Identification Methods, in *Proc. of the WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*. 485–492.

Jain, N.; Han, E.; Mobasher, B.; Srivastava, J. 1997. Web Mining: Pattern Discovery from World Wide Web Transactions, in *Proc. of the Int. Conference on Tools with Artificial Intelligence*. 558–567.

JavaServer_Pages(TM)_Technology. [accessed 2007.06.01]. Available from Internet: <www.java.sun.com/>.

Jeffry Dwight; Erwin, M. 1996. Special Edition Using CGI, 287 p.

Jicheng, W.; Yuan, H.; Gangshan, W.; Fuyan, Z. 1999. Web Mining: Knowledge Discovery on the Web, in *Ieee International Conference on Systems Man and Cybernetics.* 2): 137–141.

Kanth Ravi, K. V.; Ravada, S. 2002. Personalization and Location–based Technologies for E–commerce Applications., in *Journal for e–commerce tools and applications*. 1(1): 1–5.

Kargupta, H.; Hamzaoglu, I.; Stafford, B. 1997. Scalable, Distributed Data Mining – An Agent Architecture, in *Proc. of the Knowledge discovery and data mining*. 211–214.

Kato, H.; Hiraishi, H.; Mizoguchi, F. 2001. Log summarizing agent for web access data using data mining techniques, in *Proc. of the Ieee Intelligent Systems and Their Applications*. 2642–2647.

Kleinberg, J. M. 1998. Authoritative Sources in a Hyperlinked Environment, in *Proc. of the Discrete algorithms*. 668–677.

Konopnicki, D.; Shmueli, O. 1995a. W3QS: A Query System for the World-Wide Web, in *Proceedings of the International Conference on Very Large Data Bases*.): 54-65.

Kosala, R.; Blockeel, H. 2000. Web Mining Research: A Survey, in *SIGKDD Explorations*. 2(1): 1–15.

Krishnapuram, R.; Joshi, A.; Nasraoui, O.; Yi, L. 2001. Low-Complexity Fuzzy Relational Clustering Algorithms for Web Mining, in *Ieee Transactions on Fuzzy Systems*. 9(4): 595–607.

Kushmerick, N.; Weld, D. S.; Doorenbos, R. 1997. Wrapper Induction for Information Extraction, in *Proc. of the Artificial intelligence*. 729–737.

Lakshmanan, L. V. S.; Sadri, F.; Subramanian, I. N. 1996. A Declarative Language for Querying and Restructuring the Web, in *Proc. of the research issues in data engineering*. 12-23.

Langley, P. 1999. User Modeling in Adaptive Interfaces, in *Proc. of the User modeling*. 357–370.

Lee, Y.-S.; Yen, S.-J. 2008. Incremental and interactive mining of web traversal patterns, in *Information Sciences*. 178(2): 287–306.

Li, H.; Lee, S.-Y.; Shan, M.-K. 2006. DSM-PLW: single-pass mining of path traversal patterns over streaming web click-sequences, in *Journal of Computer and Telecommunications Networking* 50(10): 1474–1487

Li, J.; Zaiane, O. R. 2004. Combining Usage, Content, and Structure Data to Improve Web Site Recommendation, in *Proc. of the E-commerce and web technologies*. 305–315.

Li, W. S.; Shim, J.; Candan, K. S.; Hara, Y. 1998. WebDB: A Web Query System and Its Modeling, Language, and Implementation, in *Proc. of the Research and technology advances in digital libraries*. 216–227.

Lin, I. Y.; Huang, X. M.; Chen, M. S. 1999. Capturing User Access Patterns in the Web for Data Mining, in *Proc. of the Tools with artificial intelligence*. 345–348.

Luotonen, A.; Altis, K. 1994. World-Wide Web proxies, in *Computer Networks and ISDN Systems*. 27(2): 147–154.

Lup Low, W.; Li Lee, M.; Wang Ling, T. 2001. A knowledge-based approach for duplicate elimination in data cleaning, in *Information Systems*. 26(8): 585–606.

Madria, S. K.; Bhowmick, S.; Ng, W. K.; Lim, E. P. 1999. Research Issues in Web Data Mining, in *Proc. of the Data warehousing and knowledge discovery*. 303–312.

Mannila, H.; Toivonen, H.; Verkamo, I. 1995. Discovering Frequent Episodes in Sequences, in *Proc. of the Knowledge discovery & data mining*. 210–215.

Markov, Z.; Larose, D. T. 2007. Data mining the web, 218 p.

Maruster, L.; Faber, N. R.; Jorna, R. J.; R., H. 2008. Analysing agricultural users' patterns of behaviour: The case of OPTIRas^{TM}, a decision support system for starch crop selection, in *Agricultural Systems.* 98(3): 159–166.

Masseglia, F.; Teisseire, M.; Poncelet, P. 2002. Real Time Web Usage Mining with a Distributed Navigation Analysis, in *Proc. of the 12th International Workshop on*

Research Issues in Data Engineering: Engineering e-Commerce/ e-Business Systems (RIDE'02). 169–174.

Mauldin, M. L.; Leavitt, J. R. 1994. WEb agent related search at the center for Machine Translation, in *Proc. of the SIGNIDR Meeting*. 12–18.

McBryan, O. A. 1994. Genvl and WWWW: Tools for taming the web, in *Proc. of the First Intl. WWW Conference*. 15–29.

Mecca, G.; Atzeni, P.; Masci, A.; Merialdo, P.; Sindoni, G. 1998. The ARANEUS Web-Base Management System, in *Sigmod Record*. 27(2): 544-546.

Mendelzon, A. O.; Mihaila, G. A.; Milo, T. 1996. Querying the World Wide Web, in *Proc. of the Parallel and distributed information systems*. 80–91.

MINEit. [accessed 2001.11.21]. Available from Internet: <http://www.mineit.com/products/easyminer/>.

Mitchell, T. 1997. Machine Learning, 414 p.

Mobasher, B.; Dai, H.; Luo, T.; Nakagawa, M. 2002. Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks, in *Proc. of the International conference on data mining*. 669–672.

Mobasher, B.; Dai, H.; Luo, T.; Sun, Y.; Zhu, J. 2000. Integrating Web Usage and Content Mining for More Effective Personalization., in *Proc. of the International Conference on Electronic Commerce and Web Technologies.* 165–176.

Montgomery, A. L.; Faloutsos, C. 2001. Identifying Web Browsing Trends and Patterns, in *Computer*. 34(7): 94-95.

Mulvenna, M.; Norwood, M.; Buechner, A. 1998. Data–Driven Marketing, in *Electronic Markets*. 8(3): 32–35.

Musciano, C.; Kennedy, B. 2004. HTML & XHTML: The Definitive Guide, 932 p.

Nahm, U. Y.; Mooney, R. J. 2000. Using Information Extraction to Aid the Discovery of Prediction Rules from Text, in *Proc. of the Knowledge Discovery in Databases*. 51–58.

net.Analysis. [accessed 2003.05.02]. Available from Internet: <http://www.netgen.com/>.

NetGenesis. [accessed 2003.05.04]. Available from Internet: <http://www.netgen.com/>.

OnlineDictionary. [accessed 2009.02.02]. Available from Internet: <http://onlinedictionary.datasegment.com/word/Hypernym/>.

Pabarskaite, Z. 2002. Implementing advanced cleaning and end–user interpretability technologies in Web log mining, in *Proc. of the 24th International Conference on Information Technology Interfaces (IEEE Cat. No.02EX534)*. 109–113.

Pabarskaite, Z. 2003. Decision trees for web log mining, in *Intelligent Data Analysis*. 7(2): 141–154.

Pabarskaite, Z.; Raudys, A. 2002. Advances in Web usage mining, in *6th World Multiconference on Systemics, Cybernetics and Informatics*. 11(2): 508–512.

Pabarskaite, Z.; Raudys, A. 2007. A process of knowledge discovery from web log data: Systematization and critical review, in *Journal of Intelligent Information Systems*. 28(1): 79–104.

Padmanabhan, B.; Tuzhilin, A. 1999. Unexpectedness as a measure of interestingness in knowledge discovery, in *Decision Support Systems*. 27(3): 303–318.

Padmanabhan, B.; Tuzhilin, A. 2000. Small is Beautiful: Discovering the Minimal Set of Unexpected Patterns, in *Proc. of the International conference on knowledge discovery and data mining; KDD 2000*. 54–63.

Perato, L.; Al Agha, K. 2001. Mobile Agents for Improving the Web Access in the UMTS System, in *Proc. of the Ieee Vehicular Technology Conference*. 2599–2603.

Perkowitz, M.; Etzioni, O. 1997a. Adaptive Sites: Automatically Learning from User Access Patterns, in *Proc. of the 6th int. WWW conference*. 22–26.

Perkowitz, M.; Etzioni, O. 1997b. Adaptive Web sites: An AI challenge, in *Proc. of the IJCAI'97*. 16–20.

Perkowitz, M.; Etzioni, O. 1998. Adaptive Web sites: Automatically Synthesizing Web Pages, in *Proc. of the AAA/98*. 727–732.

Perkowitz, M.; Etzioni, O. 1999. Towards Adaptive Web Sites: Conceptual Framework and Case Study, in *Proc. of the Eighth International World Wide Web Conference*.

Perkowitz, M.; Etzioni, O. 2000. Towards Adaptive Web Sites: Conceptual Framework and Case Study., in *Artificial Intelligence.* 118(1–2): 245–275

Peterson, T.; Pinkelman, J. 2000. Microsoft OLAP Unleashed, 950 p.

PHP_Hypertext_Preprocessor. [accessed 2002.10.02]. Available from Internet: <http://www.php.net/>.

Piatetsky-Shapiro, G.; Matheus, C. J. 1994. The Interestingness of Deviations, in *Proc. of the Knowledge discovery in databases.* 25–36.

Pirjo, M. 2000. Attribute, Event Sequence, and Event Type Similarity Notions for Data Mining., Doctoral thesis. Department of Computer Science.

Pirolli, P.; Pitkow, J.; Rao, R. 1996. Silk from a Sow's Ear: Extracting Usable Structure from the Web, in *Proc. of the Human factors in computing systems: Common ground; CHI 96*. 118–125.

Pitkow, J. 1997. In Search of Reliable Usage Data on the WWW, in *Proc. of the The Sixth International World Wide Web Conference*. 451–463.

Pitkow, J.; Bharat, K. 1994a. WEBVIZ: a tool for World-Wide Web access log analysis, in *Proc. of the First International World Wide Web Conference*. 35–41.

Pitkow, J.; Margaret, R. 1994c. Integrating Bottom-Up and Top-Down Analysis for Intelligent Hypertext, in *Proc. of the Intelligent Knowledge Management*. 12–17.

Pitkow, J.; Pirolli, P. 1999. Mining Longest Repeating Subsequences to Predict World Wide Web Surfing, in *Proc. of the Internet technologies and systems; USENIX symposium on internet technologies and systems.* 139–150.

Pitkow, J.; Recker, M. 1994b. A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns, in *Proc. of the Second International World Wide Web Conference*. 2–8.

Pitkow, J.; Recker, M. 1995. Using the Web as a Survey Tool: Results from the Second WWW User Survey, in *Proc. of the Third International World Wide Web Conference*. 809–822.

Pyle, D. 1999. Data Preparation for Data Mining, *Morgan Kaufman Publishers Inc.* 540 p.

Quinlan, J. R. 1993. C4.5: programs for machine learning, 302 p.

Quinlan, J. R. 1996. Improved use of continues attributes in C4.5, in *Journal of Artificial Intelligence Research*. 4(2): 77–90.

Raudys, S. 2001. Statistical and Neural Classifiers: An integrated approach to design, *Springer-Verlag*. 312 p.

Raudys, S.; Pabarskaite, Z. 2004. Fixed non-linear combining rules versus adaptive ones, in *Proc. of the Artificial Intelligence and Soft Computing - ICAISC 2004*. 260–265.

Riloff, E. 1995. Little Words Can Make a Big Difference for Text Classification, in *Proc. of the Sigir '95*. 130–136.

Roberts, S. 2002. Users are still wary of cookies, in *Computer Weekly*. 345(4): 24-28.

Salton, G.; McGill, M. 1983. Introduction to Modern Information Retrieval, *McGraw Hill*. 267 p.

Sarukkai, R. R. 2000. Link prediction and path analysis using Markov chains, in *Computer Networks.* 33(5): 377–386.

SAS_Webhound. [accessed 2008.02.01]. Available from Internet: <http://www.sas.com/products/webhound/index.html>.

Savola, T.; Brown, M.; Jung, J.; Brandon, B.; Meegan, R.; Murphy, K.; O'Donnell, J.; Pietrowicz, S. R. 1996. Using HTML, 1043 p.

Schechter, S.; Krishnan, M.; Smith, M. D. 1998. Using path profiles to predict HTTP requests, in *Computer Networks and Isdn Systems*. 30(1-7): 457–467.

Schmitt, E.; Manning, H.; Yolanda, P.; Tong, J. 1999. Measuring Web Success, 23(2): 2–7.

Scott, S.; Matwin, S. 1999. Feature Engineering for Text Classification, in *Proc. of the Machine learning*. 379–388.

Shen, Y.; Xing, L.; Peng, Y. 2007. Study and Application of Web-based Data Mining in E-Business, in *Proc. of the Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*. 812–816.

Silberschatz, A.; Tuzhilin, A. 1995. On Subjective Measures of Interestingness in Knowledge Discovery, in *Proc. of the Knowledge discovery & data mining*. 275–281.

Soderland, S. 1997. Learning to Extract Text-based. Information from the World Wide Web, in *Proc. of the KDD-97 Conference*. 251–254.

Spertus, E. 1997. ParaSite: mining structural information on the Web, in *Computer Networks and Isdn Systems*. 29(8/13): 1205–1215.

Spiliopoulou, M. 1999. Managing interesting rules in sequence mining, in *Proc. of the 3rd European Conf. on Principles and Practice of Knowledge Discovery in Databases PKDD'99*. 554–560.

Spiliopoulou, M.; Mobasher, B.; Berendt, B.; Nakagawa, M. 2003. A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis, in *INFORMS Journal on Computing*. 15(2): 171–190.

Srikant, R.; Agrawal, R. 1996. Mining Sequential Patterns: Generalizations and Performance Improvements, in *Proc. of the Extending database technology*. 3–17.

Srivastava, J.; Cooley, R.; M., D.; P.N, T. 2000. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data, in *SIGKDD Explorations*. 1(2): 12–23.

Tan, A. H. 1999. The state of the art and the challenges, in *Proc. of the Pacific Asia Conference on Knowledge Discovery and Data Mining PAKDD'99, Workshop on Knowledge Discovery from Advanced Databases*. 65–70.

Tauscher, L.; Greenberg, S. 1997. How people revisit web pages: empirical findings and implications for the design of history systems, in *International Journal of Human Computer Studies*. 47(1): 97–138.

Webtrends. [accessed 2009.02.02]. Available from Internet: <http://www.webtrends.com/>.

Weiss, S. M.; Apte, C.; Damerau, F. J.; Johnson, D. E.; Oles, F. J.; Goetz, T.; Hampp, T. 1999. Maximizing Text-Mining Performance, in *Ieee Intelligent Systems and Their Applications*. 14(4): 63–69.

Wilson, R. 1996. Introduction to Graph Theory, *Addison Westley Longman Higher Education*. 187 p.

WUM. [accessed 2003.04.08]. Available from Internet: <http://wum.wiwi.hu-berlin.de/>.

Xiao, J.; Zhang, Y. 2001. Clustering of Web Users Using Session–based Similarity Measures, in *Proc. of the IEEE, Computer Networks and Mobile Computing*. 223–228.

Xiao, Y.; Dunham, M. H. 2001. Efficient mining of traversal patterns, in *Data & Knowledge Engineering*. 39(2): 191–214.

Yang, Q.; Wang, H.; Zhang, W. 2002. Web-log Mining for Quantitative Temporal-Event Prediction, in *IEEE Computational Intelligence Bulletin*. 1(1): 10–18.

Yi, J.; Sundaresan, N. 2000. Metadata Based Web Mining for Relevance, in *Proc. of the International database engineering and applications symposium*. 113–121.

Yun, C. H.; Chen, M. S. 2000. Mining Web Transaction Patterns in an Electronic Commerce Environment, in *Proc. of the 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*. 216–219.

Yun, C. H.; Chen, M. S. 2000. Using Pattern-Join and Purchase-Combination for Mining Web Transaction Patterns in an Electronic Commerce Environment, in *Proc. of the 24th International Computer Software and Applications Conference*. 99–104.

Zorrilla, M. E.; Alvarez, E. 2008. MATEP: Monitoring and Analysis Tool for E-Learning Platforms, in *Proc. of the Advanced Learning Technologies*. 611–613.

List of Published Works on the Topic of the Dissertation

In the reviewed scientific publications

Pabarskaite, Z.; Raudys, A. 2007. A process of knowledge discovery from web log data: Systematization and critical review, in *Journal of Intelligent Information Systems*. 28(1): 79–104. ISSN 0925-9902. (Thomson ISI Web of science)

Raudys, S; Pabarskaite, Z. 2004. Fixed Non-linear Combining Rules versus Adaptive Ones, in *Lecture Notes in Computer Science*. Springer-Verlag. 260–265. ISSN 0302-9743. (Thomson ISI Web of science)

Pabarskaite, Z. 2003. Decision trees for web log mining, in *Intell. Data Anal.* 7(2): 141–154. ISSN 1088-467X.

In the Proceedings of the conferences included into international databases

Pabarskaite, Z.; Raudys, A. 2002. Advances in Web usage mining, in *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*. 508–512. ISBN 980-07-8150-1.

Pabarskaite, Z. 2002. Implementing advanced cleaning and end-user interpretability technologies in Web log mining, in. *Proc. of the 24th International Conf. on Information Technology Interfaces (ITI 2002)*. 109–113. ISSN 1330-1012.

Appendixes

Appendix A. A brief history of the Internet

The Internet has evolved over the past 25 years to become an incredibly important communications medium. What follows is a brief history of the Internet (Fifteen_years_of_the_web; Internet_Corporation_for_Assigned_Names_and_Numbers).

1969 – The U.S. Department of Defence starts researching a new networking project. The first node in the network is established at UCLA and, soon after, nodes are set up at Stanford Research Institute, UCSB, and the University of Utah.

1971 – The number of connected nodes reaches 15 as additional government and education institutions are brought online. The ability to send e–mail over the Internet is introduced.

1972 – Telnet is introduced to permit remote host access over the Internet.

1973 – The U.S. Defence Advanced Research Projects Agency begins work on the "Internetting Project" to research ways to link different kinds of packet networks. FTP, the File Transfer Protocol, is introduced.

1977 – E–mail specifications are formalized.

1983 – Name server technology is developed at the University of Wisconsin.

1984 – DNS, the Domain Name Service, is introduced.

1986 – The U.S. National Science Foundation starts developing NFSNET, a major Internet backbone. NNTP, the Network News Transfer Protocol, is introduced to enhance the performance of Usenet news.

1987 – Number of hosts connected to the Internet tops ten thousand.

1988 – Internet "worm" cripples the Internet, affecting over 60,000 hosts. IRC, Internet Relay Chat, is introduced.

1989 – Number of hosts connected to the Internet tops one hundred thousand.

1991 – Gopher is introduced. World Wide Web is released by CERN, the European Laboratory for Particle Physics, located near Geneva, Switzerland.

1992 – Number of hosts connected to the Internet tops one million.

1993 – The InterNIC is created to handle directory and domain registration services.

1995 – The World Wide Web becomes the service generating the most Internet traffic. InterNIC starts charging an annual fee for domain name registrations.

1997 – Domain name business.com sold for $150 000.

1998 – The first Google office was opened in the garage of California. The first community of Blogs appears.

2000 – Developed technology for active server pages, the birth of Napster for exchanging music files, the spread of wireless technologies. The appearance of the new protocol IPv6 which allows each Internet user to have his own digital address. The peak of “doc.com” technologies.

2001 – The first Act of the digital offences signed by the European Council. Wikipedia is founded.

2002 – 162 128 493 number of Internet nodes (node can be a computer or some other device, such as a printer. Every node has a unique network address). Blogs became popular.

2003 – the first official election over the Internet in Switzerland.

2004 – 285 139 107 number of Internet nodes. “Google“ is launched.

2005 – “Youtube.com“ is launched for storing video records.

2006 – 439 286 364 number of Internet nodes.

WWW expansion

By mid-1993, there were 130 sites on the www. After half of the year, there were over 600. By 1997 there were almost 100,000 Web sites in the world. For the first few months of its existence, the Web was doubling in size every three months. Even in 1997–1998, its doubling rate was less than five months. Table A.1 shows just how quickly the Web has grown over its three-year history.

Table A.1 The growth of the www

Data	Number of web sites
1993.6	130
1993.12	623
1994.6	2 738
1994.12	10 022
1995.6	23 500
1996.1	90 000
1997	342 081
2000	20 000 000
2005	75 615 362
2006	92 615 362

Appendix B. The most popular software for analysing web logs

This appendix lists the most popular up to date software packages for analyzing web log data. They are: Accrue Software, Autonomy systems, Amadea, Angoss, BlueMartini, Blossom software, Clementine, Quadstone, Data Mining Suite, **OK-log**, Lumio, Megaputer WebAnalyst, MicroStrategy, net.Analysis, **NetTracker**, Prudsys, SAS Webhound, Torrent WebHouse, **Webtrends**, XAffinity(TM), XML Miner, **123LogAnalyser**, Caesius Software, WUM (non commercial), Analog (non commercial), **Funel**. Several of the packages were available for free evaluation (see Table B.1). Therefore it was possible to examine their cleaning capabilities.

Most existing web log mining software companies do not provide complete software packages, but solutions and many programming and adjusting work has to be done on the site according customers requirements (e. g. NetGenesis). Therefore some software packages are not available for evaluation and testing.

Table B.1 Available to test software analysis tools and cleaning methods they use

Software	Cleaning methods
OK-log	Dynamic pages are not included
NetTracker	No filter is implemented
Webtrends	The following files might be eliminated: Html files *.doc *.cgi csripts Compressed files Image files Audio files Video files Cold fusion pages Active server pages
123LogAnalyser	Possible to define file extensions to exclude
Funel	Possible to define file extensions to exclude

Appendix C. Abbreviations in Lithuanian

Table C.1 Most used abbreviations in Lithuanian

ANGLIŠKAI	LIETUVIŠKAI
Web	Žiniatinklis
Web log mining	Žiniatinklio žurnalo įrašų gavyba
Web site	Internetinė svetainė
Web page	Tinklalapis, tinklapis (puslapis)
Proxy, proxy server	Įgaliotasis (atstovaujantysis) serveris, esantis tarp kompiuterio arba vietinio tinklo ir išorinio tinklo
Web server	Žiniatinklio serveris (stotis)
Cookie	Slapukas – tai duomenų rinkinys, kurį sukuria svetainė ir įrašo į lankytojo kompiuterį
User, visitor	Žiniatinklio naudotojas, lankytojas
Hosting	Priegloba – tai paslaugų arba vietos suteikimas kam nors kitam savame kompiuteryje, savoje svetainėje, duomenų saugykloje
URL	Universalaus išteklio adresas, jo vieta pasauliniame tinkle. Išteklis gali būti tinklalapis, duomenų failas, programa
Script file	Scenarijaus failas (jis užkoduotas skriptų kalba) – programa, sudaryta iš interpretavimui skirtų komandų
HTML protokolas	Hipertekstų parsiuntimo protokolas žiniatinklio duomenims persiųsti. Apibrėžia HTTP serverio ir kliento naršyklės sąveiką
Hipertekstinė kalba	Kalba hipertekstams rašyti, pvz., HTML
Flash/mirginti	Keisti grafinės sąsajos spalvinius elementus
JavaScript	Scenarijų kalba, skirta interaktyvioms svetainėms projektuoti
Frame	Kadras – atskiru dokumentu pateikiama tinklalapio dalis (stačiakampė, gali būti apvesta rėmeliu). Tinklalapis gali turėti daug kadrų

Židrina PABARŠKAITĖ

ENHANCEMENTS OF PRE–PROCESSING, ANALYSIS AND PRESENTATION TECHNIQUES IN WEB LOG MINING

Doctoral Dissertation

Technological Sciences,
Informatics Engineering (07T)

ŽINIATINKLIO ĮRAŠŲ GAVYBOS PARUOŠIMO, ANALIZĖS IR REZULTATŲ PATEIKIMO NAUDOTOJUI TOBULINIMAS

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,

INFORMATIKOS INŽINERIJA (07T)

zidrina@pabarska.com

2009 04 21. 15,0 sp. l. Tiražas 20 egz.
Vilniaus Gedimino technikos universiteto
leidykla „Technika“, Saulėtekio al. 11, LT-10223 Vilnius,
http://leidykla.vgtu.lt
Spausdino UAB „Biznio mašinų kompanija“,
J. Jasinskio g. 16A LT-01112 Vilnius