Chapter 2
Advanced Techniques in Web Data Pre-processing and Cleaning

Pablo E. Román*, Robert F. Dell, and Juan D. Velásquez

Abstract. Central to successful e-business is the construction of web sites that attract users, capture user preferences, and entice them into making a purchase. Web mining is diverse data mining applied to categorize both the content and structure of web sites with the goal of aiding e-business. Web mining requires knowledge of the web site structure (hyperlink graph), the web content (vector model) and user sessions (the sequence of pages visited by each user to a site). Much of the data for web mining can be noisy. The origin of the noise comes from many sources, for example, undocumented changes to the web site structure and content, a different understanding of the text and media semantic, and web logs without individual user identification. There may not be any record of the number of times a specific page has been visited in a session as page is stored on a proxy or web browser cache. Such noise presents a challenge for web mining. This chapter presents issues with and approaches for cleaning web data in preparation for web mining analysis.

2.1 Introduction

The Web has become the primary communication channel for many financial, trading, and academic organizations. Large corporations such as Amazon and Google would not exist without the Web and they rely on web data as an important source of customer information. For more than 10 years, there have been numerous

Robert F. Dell
Naval Postgraduate School, Operations Research Department,
Monterey, California, USA
e-mail: dell@nps.edu

Pablo E. Román · Juan D. Velásquez
University of Chile, Department of Industrial Engineering,
Repblica 701, Santiago, Chile
e-mail: proman@ing.uchile.cl, jvelasqu@dii.uchile.cl

* Corresponding author.

J.D. Velásquez and L.C. Jain (Eds.): Advanced Techniques in Web Intelligence – 1, SCI 311, pp. 19–48.
springerlink.com

methods proposed for extracting knowledge from this web data [79, 70], where pre-processing of data is a critical step toward pattern identification.

Soft computing techniques such as Neural Networks, Fuzzy Logic, Bayesian methods, and Genetic Algorithm are commonly applied to web data to help cope with uncertainty and imprecision [70]. However, these soft computing techniques do not always take care of the time dependency and high dimensionality of web data, suggesting that one takes considerable care in avoiding the celebrated term "Garbage in, Garbage out" [40] . Web data can be placed into three categories: Web Structure, Web Content and Web Usage [67].

Web Structure Data corresponds to the hyperlink network description of a web site. This oriented graph in today's web 2.0 [97] becomes more dynamic and depends on a web user's actions [80]. Web 2.0 applications distinguish themselves from previous software because they take full advantage of dynamic pages and encourage social interaction such as in Facebook and Twitter. This structure was traditionally static [16] but new web applications suggest the need for a time dependent graph.

Web Content Data corresponds to the text content and relates to the field of information retrieval. There is a long history of research on both text retrieval and representation spanning more than fifty years. Vector Space Model or Bag of Word Model [68], are traditional representations, based on word frequencies of text. The focus of recent research is on more accurate representations of the semantic of text [101, 41, 49, 46, 90] and on coping with the more dynamic nature of text. Text now changes depending on both the time and circumstances. The semantic of a web page also changes due to multimedia objects. Parsing web pages, automatic interpretation of objects and pre-processing such information is part of ongoing research process [91, 98].

Web Usage Data corresponds to the trail of pages, also called a session, that tracks each web user while browsing a web site. Monitoring a web user's session can be a violation of privacy [22] and is forbidden by law in several countries [61]. For example, installing tracing software in the web user's browser is equivalent to spyware. There are other less intrusive ways of retrieving sessions; the most popular is by using a web log, a file that records each page retrieved from a web site. This data source is anonymous, but simultaneous interactions of web users makes identifying a session more complicated. With respect to the advancement of browser technology, for instance the Opera software, session retrieval has become more complicated as more sophisticated algorithms for pre-loading and buffering pages have been implemented. New studies [25, 26] relate to increasing the accuracy of sessionization.

Pre-processing web data has some well known issues [87]. Parsing problems relate to different version of HTML, DHTML and non-compliant codes. Dynamic content generated via embedded code like JavaScript or server side content generation render text content extraction unfruitful through usual parsing methods. Pages with frames produce different presentations for a user that can be interpreted as a "pageview" [70] that combines group pages and other objects together.

A pageview abstract is a specific type of user activity such as reading news, or browsing search results. Even when a session is considered as a sequence of pageviews, dynamic pages produce a large number of possible object combinations for each pageview. Log files are influenced by new browser technology (linkprefetching) [56], enhanced cache management and also by network caching devices such as proxies and parallel browsing consisting of tabs and popups.

The quality of pre-processing has been quantified using a set of measures for each data type. New measures have also been used in the case where uncertain results are obtained [25]. An efficient storage and data representation is also important because companies like Google have large data storage that incorporates most of the Web. This chapter, aims to explore the difficulty in obtaining quality data. In section 2.2 general statistical characteristics of web data are discussed. Section 2.3 presents the state the art in web graph retrieval. Section 2.4 expounds on the current methodology for retrieving text data. In Section 2.5 the process of session retrieval is analyzed and examples are provided. Finally section 2.6 provides a summary.

2.2 The Nature of the Web Data: General Characteristics and Quality Issues

Web data is not a random collection of users' interactions, pages and content. There is some regularity that remains constant across the sites and groups of web user sessions (e.g., [42]). This should impact pre-processing [25] and the evaluation of web mining.

Knowing prior web data regularities could significantly improve data mining. For example, for pattern recognition, semi-supervised clustering techniques have been popular in recent years. The results found by these algorithm are considerably better [75] because they use domain data descriptions that refine the search space where an unsupervised machine learning algorithm works [37]. The resulting subspace region is less of a factor with the additional domain information. In the field of data cleaning, deviations identify outliers or automatic anomaly detection [69].

2.2.1 Web Content

Web content information retrieval has been studied for many years. For example, the evaluation of a text for word frequencies. Today, Web content is based on multimedia in which rich text features enhance a web user's experience. In this context, extracting the semantics relies on the discovery of compact structures or a web object that represents a component in a web page.

Pure web texts do not differ from a traditional corpus and empirical studies reveal statistical regularities on the text distributions. The well known Zipf Law from 1932 [45, 68] states that in a corpus the ranked number of words per page is the power $\alpha \simeq 1$ (see equation 2.1) . Today, linguistics agree that this rule assumes speakers simplify communication by using a small pool of words that can be retrieved efficiently from their memory. Furthermore, the listeners simplify communication by selecting words with a single unambiguous meaning. Other models relate to a stochastic diffusion model for generalizing Zipf's Law [58] and cast a new perspective on the subject.

$$P(n) = \frac{n^{-\alpha}}{\sum_{k=1}^{\infty} k^{-\alpha}} \tag{2.1}$$

One of most celebrated distribution is the Heaps' Law [63] used in recent applications to describe the Internet. The heap law describes the number of unique word in a text as a power with exponent β representing its size or the number of word on a page. It has been found that $\beta = 0.52$ [62] has greater accuracy on internet pages. The study then estimates the number of n-tuples of words m having the expected number of hits on search engines, obtaining log-normal results. Also these distributions are used to measure similarity between pairs of words for grouping purposes. This kind of clustering result is useful for grouping terms on texts that capture the semantic for further processing more effectively.

Thanks to the advent of Web 2.0 applications , the content of web pages has become more dynamic. Updates on web sites have been a prolific subject of focus since web search engines are based on the accuracy of indexed terms and pages. The study [34] includes 151 millions Web pages browsed once a week within a three month period. The findings signified that 22% of pages were deleted, and 34.8% of pages had content updates (larger pages where more frequently updated). From other studies [78] the information longevity is not related to the extent of updates on the web page. It also reveals that dynamic changes can be placed into two main categories: Churn Updating behavior (33%) related to repeatedly overwriting content and Scroll Updating behavior (67%) related to lists of updates (e.g., news).

In a recent study [1] over 55,000 web pages with a diversity of visitation pattern indicated a higher dynamic content than previous studies. Consequently, further refinement of the web content representation and pre-processing must be taken into consideration in order to manage dynamic content. Some other studies reveal that in a one hour period 26% of the visited pages during the study were modified or updated in some way and 66% of those pages were visited the following day, of which 44% consisted of query parameters. Earlier studies, (e.g., [18]) report that only 23% of the pages consisted of dynamic content for the duration of one day.

2.2.2 Web Site Structure

Early studies of web site structure suggested a static graph [16] with pages as nodes and hyperlinks as edges. However, the link structure is as dynamic as content and develops exponentially [6]. Despite the changing structure, large-scale statistical analyses of the hyperlinks [5, 53] show a power law distribution ($p(x) \sim x^{-\alpha}$) has a good fit for several structural measures (categorized by the exponent α). The study [5] suggests that the number of pages per web site is a power law with exponent $\alpha \in [1.2, 1.6]$ [31]; the number of pages that have a given number of in-links have $\alpha \in [1.6, 2.1]$, the number of pages with a given number of out-links has a piecewise power law with $\alpha_1 \in [0.3, 0.7]$ and $\alpha_2 \in [1.9, 3.6]$. Finally the ranking number (Page

Rank) has $\alpha \in [1.8, 1.9]$. The distribution of the age of a page corresponding to the *last modified* value register is a Poisson process [19] with decay parameter $\lambda \in [1.6, 2.3]$.

An important observation taken from current studies is that the notion of a page is losing its importance as a fundamental information source in relation to its description of the overall web structure [6]. In todays dynamic web applications what is commonly described as a web page depends on the parameters that the application receives. Today, a complete web site could be served by a unique application file in which each page is represented by the URL's query parameters.

As stated by [30], the temporal component of the graph evolution has not been conclusively studied by the web mining community. They consider three levels of study: temporal ”single nodes”, temporal ”sub-graphs” and ”whole graph” analysis. The single node study examines how frequently a page is accessed during a time period in which there are no changes to a page. In this case, the data mining approach consists of clustering over page content. The subgraph level consists of finding the period of time where a small level of change has occurred within a structure based on graph properties such as order, size, components, ”Max Authorities”, ”Hub”, and ”Page Rank”. The ”whole graph” analysis focuses on identifying a set of measures for building a features vector of the graph. Like in the subgraph level, the features could be represented by basic graph properties (e.g., order and size) or derived properties (e.g., ”Max Hub” score and ”Average Hub” score). A current means of research is to explain the changes of such measures in time.

2.2.3 Web User Session

The celebrated Law of Surfing [42, 104, 43] was noticed ten years ago as a distribution that regularly fits the number of sessions of a given size. This distribution was recognized as an inverse Gaussian (see equation 2.2) whose parameters change depending on the Web site $(E[L] = \mu, Var[L] = \mu^3/\lambda)$. This law is also approximated by the Zipf's laws simplifying the calculation process as in log scale the fitting problem reduce to linear regression.

$$P(L) = \sqrt{\frac{\lambda}{2\pi L^3}} e^{-\lambda (L-\mu)^2/2\mu^2 L} \tag{2.2}$$

Some empirical studies have been performed on browsing customs [95, 13, 102], with recent changes on web user general behavior reported in [103]. The change in browsing habits can be explained by the highly dynamics environments and new internet services. The use of a browser's interface widget orchestrates the changes in frequencies. For example users backtracking on certain sites are declining. Earlier studies [106] report 30% back button activation. However, new studies [77] report only 14% revisitation patterns. This behavior is most likely also the outcome of new applications incorporating backtracking support within a web page. The use of the forward button also resulted in decrease usage; decreasing from 1.5% to 0.6%. The reload button usage has also decreased from 4.3% to 1.7%.

New windows and submitting activities reported an increase in total user's actions of ($\sim$ 10%), as they are key features of dynamic pages. Despite this, the link click stream seems to maintain its percentage of actions in web browsers at 44%. As reported in [77], search engines, now considered new browsing assisting tools, carried out 16% of page visits. Another new behavior also appears as navigating simultaneously with new windows and tabs, where an average of two pages are opened throughout a session. It confirms that parallel navigation is not the exception but the rule.

The same study [77] shows a heavy tail distribution of the average stay time on a page where a considerable fraction of the visits last a very short time. Revisiting pages or scanning behavior corresponds to a fraction of the entire user's action. Despite these observations, a direct correlation is found between time remaining on the site and the number of hyperlinks on the page (also with the number of words).

[3, 2] perform other empirical work on navigating of electronic documents like pdf and MS Word. Despite the many differences between users, they report some similarities. A power law distribution models the number of times a document opens. Some interesting patterns were discovered by mean of a tracking system on the user interface action. The period of time spent reading a section of the text appears to relate to the interest of the user and to the quality of content. In these cases, the main methods of navigating correspond to scrolling operations, where distances travelled in relation to session size are revealed to have a power law distribution. They report the same behavior on the mean percent of time in text regions with a heavy tail distribution. A general rule for human behavior seeking information seems to rule the interactions on internet and document seeking behavior.

Web usage research is mainly driven by click stream data (web user's sessions). A recent study on more precise biometric data in conjunction with Web Usage Data shows that click streams are a bias source of information [50, 38]. Study [38] uses eye monitoring techniques to focus on the click stream in a Google search result page. This reflects how web users' behave in relation to the texts and links. Eye tracking data provides important information about the web user's cognitive processing that have a direct relation to their actions. This study highlights that clickstreams are directly influenced by the order of the presented search result; the first links on a page are likely to be chosen as they tend to be inspected to a greater extent. The first visual portion of the web page also has a direct influence on the click decision as a web user prefers to process this first portion. In this way, sessions obtained from web logs do not directly reflect web user behavior. The visual disposition of the elements on the web page must also be taken into consideration.

2.2.4 Privacy Issues

Legal issues must be considered when collecting web usage data [44]. The extraction of information from web data can lead to the identification of personal data that can be illegal to collect. Any general solution to issues of what to collect must balance four forces: laws, social norms, the market, and technical mechanism.

Privacy relates to “The control of information concerning an individual,” “the right to prevent commercial publicity of one’s own name and image,” “the right to make certain personal and intimate decisions free from government interference,” and “to be free from physical invasion of one’s home or person” [73]. Certainly, a whole spectrum of solutions exists but all of them need to be tested under current laws and social norms. While one method may protect individuals, it could compromise social information (e.g., racial discrimination) and can lead to legal prosecutions.

Legislation on privacy control is still unclear [73]. The W3C organization produces a policy called the P3P Platform for Privacy Preferences [66]. This policy stipulates that information about web users must be protected by privacy laws and the users themselves informed about their rights. It suggests [73] a continuous process of negotiating, the inclusion of relevant third parties and an optimum or acceptable level of disclosure of personal information in an online environment.” Hence, an optimal level of privacy strives to return the privacy control to the web user, and declares that browsing preferences must determine the level of privacy.

A user’s privacy can be resolved by incorporating a “slider” on a web browser that controls behavior monitoring. The P3P protocol should be the mechanism that determines and illustrates the level of privacy a user is entitled to in each web site. The use of this protocol is increasing among web sites [85]. For instance, with a common session tracking mechanism (section 2.5.2.1) the P3P protocol defines the following cookie persistent or non-persistent, first party or third party, non-personal or personal. “Persistent” cookies correspond to a permanent cookie repository that can store all interaction within the web. “First party” cookies send information only to the web server of the page while “Third party” allows to distribute data to any other server. “Personal” cookies store any personal web user data. According to this protocol, the most privacy compliant cookie is a non-persistent, first party and non-personal. The worst privacy compliant cookie is a persistent, third party and personal. This information will be present on the P3P protocol and future non-compliant privacy tracking session methods will be banned by most users [73].

2.2.5 Quality Measures

Ensuring web data quality is important for data mining. We will be presented in this chapted different quality measures for different types of web data. Quality metrics also enable the implementation of pre-processing filters.

Web Structure Data

The discipline of graph mining [14] relates to the analysis of the web structure’s intrinsic graph properties. Data quality depends on the application where the data is used. For example, in web structure mining, large web crawling results need to be evaluated for the page rank value as a measure of the importance of a page [93]. The Page Rank value is an indicator of the visibility that a web page has on a search engine. Another metric corresponds to the hub and authority scores [14] that concentrates on out-going and in-going links, [84] discusses more indirect measures based on the web user action on the search results.

$$c_i = \begin{cases} \frac{n_i}{k_i} & k_i > 1 \\ 0 & k_i \in \{0,1\} \end{cases} \tag{2.3}$$

There are graph theoretical measures related to community structure [14] that are useful for controlling data quality. The clustering coefficient (Equation 2.3 where k_i is number of neighbors of node i and n_i the number of edges between them) reflects the degree of transitivity of a graph. Other measures relate to identifying the number of connected sub-graph components. Thus a disconnected web graph is an indicator of some kind of problem in the data collection or a recent change. Another important indicator is the resilience value [14]. This is obtained by finding the minimum cut-set cardinality of the graph. A cut-set is a set of links that split the graph into two components. A data set with a high resiliency value should be the most representative.

Web Content Data

Web pages are special cases in the context of text pre-processing, since HTML tags contain both embedded and miscellaneous information (e.g., advertisement and logos). It is estimated that an average of 13.2% of the embedded information on web pages is noise related [111]. A similar value found between vector model text content and a predefined set of topics provides an indication of the quality of the content data [111]. Filtering embedded information familiar with this measure helps with pre-processing text data.

Web Usage Data

The set of web users' sessions correspond to web usage data. The quality of such large data could be tested using the session size distribution, which follows a site independent distribution law [42]. For simplicity, this distribution is approximated by a power law [64]. The goodness of fit value to this distribution is a quality measure of the session set [25, 26, 27].

If real sessions are available (for example using logged users), then the "degree of overlap" between sessions can be compared [94]. It is defined as the maximal averaged fraction of real sessions that are recovered using the sessionization. Another wide spread measure [7] corresponds to the precision and recall scheme. The precision p is a measure of exactness that can be defined as $p = N_p/N_s$, where N_p is the number of maximal patterns positively recovered and N_s is the number of maximal patterns recovered from the sessionization. The recall measure is a measure of completeness that can be defined as $r = N_p/N_r$, where N_r is the number of correct maximal patterns. An accuracy value can be defined as a geometric mean $a = \sqrt{p * r}$. Greater accuracy means a more effective sessionization method. Simulation can also be used to obtain artificial sessions [94, 7]. Such simulations should be "human compliant" with statistical behavior following known strong regularities [42]. Some such simulation results are available [86].

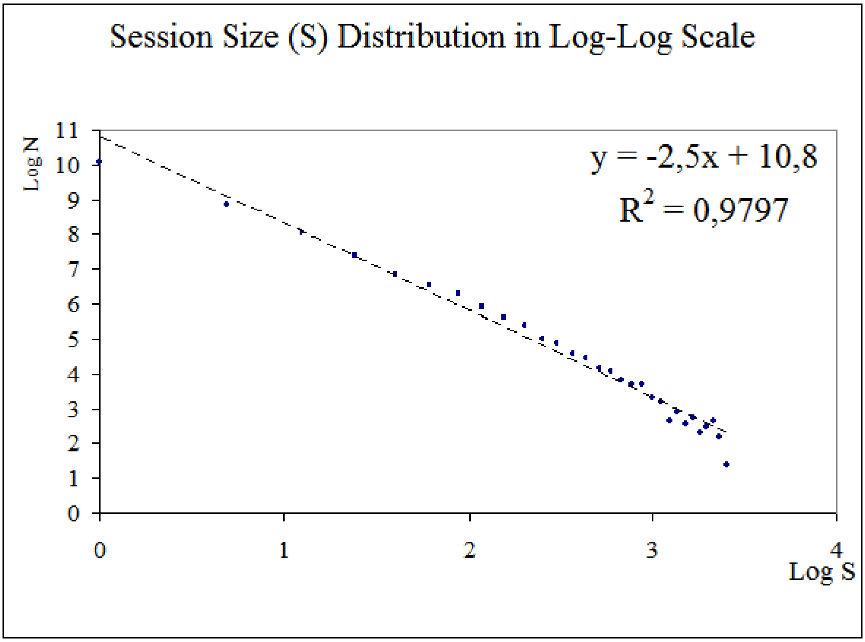

Fig. 2.1 Session size distribution shows an approximate power law for a cookie sessionization. Data extracted April-June 2009 from http://dii.uchile.cl web site

General Measures
Data mining algorithms are influenced by data quality [107]. For instance, a neural network is particularly sensitive to this issue as reported in [108]. One way to handle this is by using anomaly detection algorithms. Because an anomaly is based on parameter settings it can be considered as a metric of data quality. Support vector machines (SVM) have been largely used for these purposes [100]. SVM can learn complex regions and locate outliers. The fraction of outliers provides a measure of the quality of the data. This general measuring principle can be applied to web feature vectors as defined in section 2.5.1. Furthermore, finding the outliers implies a mechanism for data cleaning.

2.3 Transforming Hyperlinks to a Graph Representation

With respect to the advent of search engines, the field of structure retrieval has been widely studied. The web structure as a graph representation has it own specific data mining process [14]. Large scale retrieval schema spanning the Web has been implemented successfully. Web page structure must be retrieved by sampling web pages following the observed hyperlink structure. This process is called Web

Crawling [12] and involves the storage of the hyperlink structure and web page content.

2.3.1 Hyperlink Retrieval Issues

A major difficulty for hyperlink retrieval is large volumes and the frequent rate of change regarding web data [1]. The quantity of data slows the complete retrieval process. A large crawling of the web must select an update strategy such as selecting the pages most likely to change first. The crawler must have a set of policies regarding: the page selection method, the revisit schedule, a politeness policy, and parallel processing. Several strategies are available based on incomplete information: Breadth search ordering, Page rank based, prediction of changing pages [53]. Other issues relate to the imperfect mapping between URL and the page that is visually seen by users [87]. A common HTML structure like frameset, groups a set of URLs in the same virtually presented page. Hence a frameset produces confusion in the process of mapping URLs to web pages. Nevertheless, a solution considers the group of web pages as a “pageview” object [70].

2.3.2 Crawler Processing

The high rate of web page modification [1] suggests that the time of page retrieval could be near the time of page content expiration. It is important to prioritize the most important pages for retrieval, and that requires a measure of importance for the selection algorithm. The revisiting policy should seek to maintain the freshness of the pages, and once predicting an updated page, it is inserted on the pile for crawling (Figure 2.2). The strategy for revisiting pages could be using the same frequency for all (Uniform case), proportional to its registered updating frequency, or futher estimation for the time of page obsolescence. A recent method based on the longevity of web pages [78] optimizes the cost of page retrieval using a generative model. Web pages are modelled as an independent set of content regions of three types: Static, Churn Content, Scroll Content. Such content types have specific lifetime probability distributions that are fitted with observed data. The page lifetime is then estimated using this distribution. The retrieval mechanism must ensure is does not overload the web server. This is called the web crawler’s politeness policy, if they are not compliant, the retrieval program could be blocked on the network by a third parties. The visit interval setting for the same web server should have a lower limit (usually 10 seconds) that ensures a minimal impact on the web server. This produces a slow retrieval rate because a web site can have anywhere from a thousand to a million pages. Finally a parallel algorithm reduces the processing time but should be designed to avoid retrieving the same page twice.

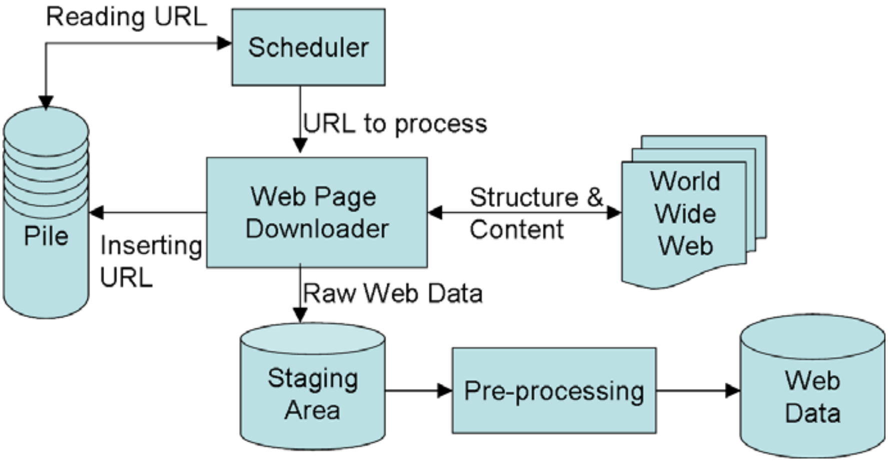

Fig. 2.2 High level Crawler Structure

2.3.3 Large Sparse Distributed Storage

A crawler that inserts data directly into a database results in poor performance [12]. Retrieved data is therefore stored on a intermediary format for further processing. Recent advances on storage for distributed systems [17] give some insight into efficiently connecting a web crawler with structured data storage. The Google's Bigtable storage engine is a distributed storage engine for peta byte size. It is implemented over several projects including web indexing. It consists of a three dimensional map of strings that are indexed by row, column and timestamp. A hyperlink structure fits well in this schema because a row can represent the page of a hyperlink, the column can correspond to the pointed page, and the timestamp represents the link when it's retrieved. This storage facility is implemented using the Google File System and operates over a shared pool of servers. It is built on top of the Google SSTable file format that provides a persistent ordered immutable map from keys to values.

2.4 Transforming Web Content into a Feature Vector

The information presented on web pages is complicated by text and multimedia content. This information is used for page classification [83], page summarization [89], entity extraction [98] and semantic processing. This data needs to be filtered because many objects present on web pages are not related with the stated objectives (e.g., advertising). Furthermore text content by itself is noisy, for example sentences contain many word that have a poor influence on its semantic content. Text algorithms are also dependent on the text's language. [111] present a collection of text preprocessing algorithms for data mining purposes.

2.4.1 Cleaning Web Content

Web content is full of noisy items such as navigation sidebars, advertisement, copyright, and notices that can mask the principal content of a web page. Automatic detection methods, [15] claim that noise consist of more than 50% of the total content. An earlier method, [24], segments a web page into blocks and separates blocks into informative and non-informative categories. The segmentation is realized by using the DOM (W3C's Document Object Model), which decomposes the web page into a hierarchical tree structure. The objective of this method is to identify nodes from this structure which are non-informative. A node is non-informative if all its sub nodes are non-informative. The method consists of building a classifier for this property. Once all the nodes from the DOM tree representation of a web page have been labeled as non-informative the top level non-informative nodes are removed.

At the sentence level, cleaning consists of removing each Stop Word. A Stop Word corresponds to illatives, pronouns and others words that do not contain much semantic value individually. This kind of preprocessing is fundamental for the ”Bag of Word“ model where the semantic is approximated by a set of words without considering the sentence and paragraph structure semantic. An additional step is known as Porter Stemming that reduces each word to it root [82].

In English, the algorithm for stemming is simple because it corresponds to identifying the suffix and removing it. In other languages, it can be more complicated so specific stemming algorithms in several languages exist [92]. Another approach for stemming sentences consists of clustering words based on a corpus statistic [8]. The algorithm is trained on a corpus that finally defines a representative word from a set of words, the stemmed text has been shown to produce better results for automatic classification.

The changing content of a web page [1] must be managed using time window processing [47] where it is assumed that content remains constant during a defined period of time. This implies a content feature vector update using a temporal index.

A promising direction on text pre-processing is the Word Sense Disambiguation technique [96] (WSD). It addresses the problem of selecting the most appropriate sense for a word with respect to its context. The technique consists of selecting the most appropriate meaning for each word using a semantic model of the text. It was reported that WSD boosted the further information retrieval and data mining processing [105].

2.4.2 Vector Representation of Content

The simplest text representation consist of a vector $V = [\omega_i]$ of real components indexed by word. This representation comes from the Bag of Word abstraction. Despite the approximation, this model provides remarkable positive results in the information retrieval [68]. The value of each component of the vector ω_i is called a weight for word i. There are several weighting schema for words (Term), the most common of which is the TF-IDF schema. Table 2.1 presents the most common weightings.

Table 2.1 Common weighting Schema. Index i is for a unique term, index k for a unique document. A document k is represented by a vector $[\omega_{ik}]_{i=1,\ldots,N_k}$, where $\omega_{ik} = f(term\ i\ in\ document\ k)$. n_{ik} is the times the term i appears in the document k and n_{ik}^{-} correspond to the negative number of appearances predicted for the word i according to a trained algorithm. N_k is the number of terms in documents k

$f(.)$	Calculation
$binary$	$binary(\omega_i) = 1$ if the term i is present on the document k or 0 if not.
tf_k	Term frequency on the document k, $tf_k = n_{ik}/N_k$, n_{ik}/N_k is the frequency of the term i on the document k.
$logtf_k$	$logtf_k = log(1+tf_k)$
ITF	Inverse Term Frequency, $ITF = 1-1/(1+tf)$
idf	Inverse Document Frequency, $idf = log(N/n_i)$, where n_i is the number of document having the term ω_i and N the total number of document.
$tf.idf$	$tf.idf = tf * idf$ [68]
$logtf.idf$	$logtf.idf = log(1+tf)idf$
$tf.idf-prob$	The probabilistic approximation of the value $tf.idf$ using an estimator for idf
$tf.chi$	Use of the χ^2 feature selection measure, $tf.chi = tf * \chi^2$
$tf.rf$	$tf.rf = tf * log(1+n_{ik}/max\{1, n_{ik}^{-}\})$, where rf is the relevance frequency [59, 60]
$tf.ig$	$tf.ig = tf * ig$, where ig is the information gain (Kullback Leibler divergence [33]).
$tf.gr$	$tf.gr = tf * gr$, where gr is the information gain ratio [74].
$tf.OR$	$tf.OR = tf * OR$, where OR is the Odds Ratio [36].

The replacement of a document by a "Bag of Word" inevitably involves a loss of information. For instance, "The Matrix" and "Matrix" represent a film and a mathematical term. Ontology gives a more correct description of the semantic of objects on a web page. Once the ontology annotation on web objects is complete it is possible to transform it into a vector representation [11]. But first an automatic semantic annotation should be performed [81].

2.4.3 Web Object Extraction

Extracting information automatically or semiautomatically from web data has become more difficult as web sites have adopted multimedia technologies to enhance both their content and their presentation. Some of the most successful sites provide video streaming and picture sharing. Unlike text, the content of multimedia formats within web pages can be understandable only to humans. At the most, some technical information such as the color of a histogram or wave patterns for pictures and sounds can be obtained automatically. Given this, a different approach to data extraction for these formats must be adopted.

The use of metadata to describe the content of any multimedia format allows the creation of automatic or semiautomatic ways of extracting information. This enables the webpage to be described as a series of objects brought together in an ordered manner similar to a structured manner in which text and multimedia formats are displayed within a page.

An object displayed within a webpage is termed a Web Object. One definition is found in [32], a web object corresponds to text and multimedia content that a user identifies as a compact unity. Using this definition every part of a webpage can be describe as an object, a picture a sound or even a paragraph. An advantage here is that the content of a website is described not by the site itself, but in the metadata used to define the Web Objects within it.

Another significant advantage of using Web Objects is that any two objects can be easily compared. This can be achieved by defining a similarity measure that uses the metadata that describes the content of an object. This enables complex comparisons between objects that do not require the same format. For example if a webpage contains only one picture and the accompanying text describes the picture in detail, the metadata for both the picture and the text focus on the content rather than on the format. Therefore by using a similarity measure both objects can be discovered as equivalent.

The development of Web Object techniques is focused mainly through the user's point of view, as they are able to describe a website taking into consideration both the content and appearance of a web page rather than only the data which it contains. Different ways have been developed to describe web pages based on how the user perceives a particular page [10].

A large degree of research has recently been carried out in the field of Web Objects; in this section some of these are described focusing in mainly four areas: Web site Key object identification [32], Web Page Element Classification [10], Named Objects [91] and Entity extraction from the Web.

Web site Key objects Identification: In this work [32], Web Objects are defined using a specially created metadata model and a similarity measure to compare two objects. Data mining reconstructs the user sessions from the web server log and objects are created from the web-pages of a site. By inspecting the users' content preferences and similar sessions (clustering), website key objects can be found which reflect the objects within a site that captivate a user's attention.

Web Page Element Classification: This work [10] creates a method for detecting the interesting areas in a webpage. This is accomplished by dividing a webpage in visual blocks and detecting the purpose of each block based on their visual features.

Named Objects: The manner in which users' interpret a web page lies in the focus of this work [91] where a user's perception of a web page is obtained through the user's intention. Web Design Patterns are then selected based on a user's intentions. These named objects are used as the basis of mining methods which enables Web Content Mining.

Entity Extraction from the Web: A Web knowledge extraction system is created in this work [28, 35], which uses Concepts, Attributes and Entities as input data. By modelling this using ontology, facts from generic structures and formats are extracted. Subsequently a self supervised learning algorithm automatically estimates the precision of these structures.

2.5 Web Session Reconstruction

Sessionization is the process of retrieving web user sessions from web data. Web usage mining highly depends on the correct extraction of Web User sessions [23]. Several methods exist and can be classified according to the level of web user personal data (privacy protection).

Proactive methods directly retrieve a rich set of information concerning the operation of a web user. Examples are cookies based sessionization methods where personal data and activities are stored and retrieved. Other proactive methods consist of installing a tracking application on a user's computer that enables the capture of each interaction with the browser interface. In these cases, privacy issues are raised, and in some countries it is forbidden by law. A further possibility lies in the use of login information to track the web user's actions. In this case, a disclaimer agreement is required between the company and the web user in order to enable the tracking of personal information.

Reactive sessionization corresponds to indirect ways to obtain anonymous sessions. The primary source of data for reactive sessionization is the server Web Logs which contains all activities of all web users excluding personal identifiers. Several heuristics have been used to reconstruct sessions from web logs as individuals can not be uniquely identified. Recently, integer programming has also been used to construct sessions and conduct additional analyses on sessions [25, 26].

Click stream analysis has contributed to new browser technology such as link prefetching [56, 54] from the Mozilla Firefox browser [21] together with enhanced web page caching from the Opera browser [4]. Link prefetching corresponds to the loading of links in the background that can be visited in the future allowing a faster browsing experience. In this case, the access register in the log corresponds to a machine operation which does not reflect human behavior. This increases the difficulty of constructing sessions from web logs.

Sessionization based on web logs passes several processing phases [23]. Firstly, data acquisition is performed during a web server's operation, where for each HTTP request a register is recorded in the web log. During data collection, files are selected and scanned and information stored in temporal repositories for further transformation. Data cleaning consists of selecting only the valid registers: this includes discarding robots, exploits and worm attacks; and finally only html files are selected (discarding multimedia, images, etc). The chunking process [25] splits the large set of valid log registers into smaller sections based on the same IP, agent combination and a time threshold between consecutive registers. The last partition does not ensure unique sessions since multiple web users can share the same IP and agent, but it simplifies further processing. A chunk is the main data unit through which a sessionization algorithm retrieves the web user's trails.

2.5.1 Representation of the Trails

Sessions have different representations [29] depending on what further data mining is to be conducted. At least three historical representations of a session have been used.

Weight usage per pages
Following earlier work on web user profiling [72], sessions are processed to obtain a weight vector $V = [w_1, ..., w_n]$ of normalized time of visits for each page. The vector's dimension is n corresponding to the number of pages in the web site, and the index relates to the corresponding page. Vector entries are zero when the page is not visited and proportional to the time spent on the page when the user visited it. This representation is useful for web user's profiling based on clustering. The weight can be tested with many other representative measures. A binary weighting scheme could be used ($w_i \in \{0, 1\}$) that is suitable for association rule analysis [71]. In this case, the weight takes value one if the page is visited during the session. Another variation makes use of prior knowledge of each page [72].

Graph of sessions
[39] Considers the time spent on each page for similarity calculations. The representation of a session is given by the sequence of visited pages. A weighted graph is constructed using a similarity measure and an algorithm of sequence alignment [39]. This representation is further processed using a graph clustering algorithm for web user classification.

Considering text weighting
Other works [99, 48] take into account the semantics of text and the time used. An important page vector is defined containing the page (text content) and the percentage of time spent on each page sorted by time usage and selecting only a fixed number of pages with maximum time usage. This representation is generalized for multimedia content using web objects. The clustering of important page vectors are perform via the SOFM (Self Organized Feature Map) [57] algorithm obtaining categorizations for web user.

2.5.2 Proactive Sessionization

Proactive sessionization directly records a web user's trails but there are some implementation subtilties.A variety of proactive session retrieval methods are described.

2.5.2.1 Cookie Based Sessionization Method

Cookies are a data repository in the web browser that can be read and updated by an embedded program on the web page. This mechanism, usually used for persistence purposes, can uniquely identify the web user by storing a unique identifier and sending this information to the web server (Figure 2.3).

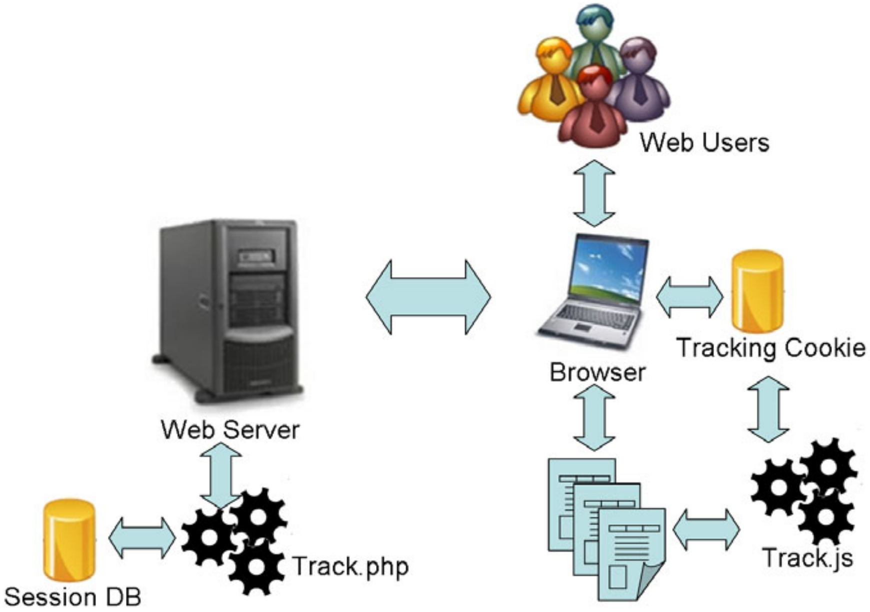

Fig. 2.3 Cookie Sessionization: An embedded script (Track.js) updates the cookie with a unique identification per each user and sends it along with the current URL to the web server web application (Track.php) for recording the page on the session's database

The method of session processing with cookies is simple in appearance but has it own issues. Security and privacy issues (already mentioned) cause web users to often disable the cookie facilities of their web browsers. Two DOM methods are available for cookie processing, the "onload" method that can be replaced by direct execution of code on HTML and the "onbeforeunload" that is more restricted for browsers executed at the moment of leaving the page. Entering and leaving a web page could be recorded on the web server in order for it to calculate the sequence of pages and time of visit foreach page. A range of browser execution policies provide different pattern access to the same session. Heuristic methods should be used to obtain sessions for some undetermined cases (Table 2.2). For example, if it was registered as an enter event on a page and then a leaving event from another page, the time of visit for both page could be approximated as half of the whole period. A more precise heuristic could estimate the visit time based on similar known sessions.

A futher refining method could be the loading time of the web page. There are 11 combinations of events from the previous page to the the next page; impression occurs in some cases when the page corresponds to the first or last page.

Table 2.2 Simple heuristic for visit time estimation with cookie (1: the event is registered, 0: if not)

Page 1		Page 2		Page 3		
(E)IN	(F)OUT	(A)IN	(B)OUT	(C)IN	(D)OUT	TIME Page 2
1	1	1	1	1	1	B-A
1	1	1	0	1	1	C-A
0	1	1	1	0	1	B-A
1	0	1	1	1	0	B-A
1	1	0	1	1	1	B-F
0	0	1	1	0	0	(D-A)/2
1	0	1	0	1	0	C-A
0	1	0	1	0	1	B-F
1	0	0	1	1	0	(B-E)/2
0	0	1	0	0	0	Indeterminate
0	0	0	1	0	0	Indeterminate

Despite the problem of security and privacy of cookies some advances have been made in order to secure its usage [109]. This consists of automatic validation of cookies for the safety of the web user, based on automatic classification algorithms. Others protocols like P3P [66, 85] relate to the web site declaration of a cookie's degree of private data retrieval.

2.5.2.2 Tracking Application

A spyware application monitors and stores events on the host machine. They can hold links to criminal activity because they can be used to retrieve personal credit card numbers and passwords. These extreme tracking applications are a world wide internet security problem, but some applications have been designed for scientific purposes, for example, AppMonitor [2] that tracks low level events such as mouse clicks on Windows OS for Word and Adobe Reader.

2.5.2.3 Logged Users

Web applications with login authentication can maintain a register of a user's trails on a proprietary database. This can be implemented by storing each page visit during the web user's visit. This is the simplest and most reliable way to track sessions but requires client permission and authentication.

2.5.3 Reactive Sessionization

Sessions obtained from log files are anonymous because web user identity does not appears explicitly in the registers. Of course, this complicates the identification of a web user's trail (Reactive Sessionization). Web users with a similar profile could be accessing the same part of a site at the same time resulting in registers that appears shuffled on Logs. Additionally, web page caching (e.g., Back Button) from browsers and proxy servers can contribute to the cause of log registers going missing. Untangling sessions from Log files requires some assumptions like the maximum time spent by web users on a session, the web site's topology compliance and the semantic content of web pages.

2.5.3.1 Traditional Heuristic Sessionization

Web Logs maintain four important fields for each register: The web user's IP address, the date and time of the request, the url requested before of the current (when activated), and the browser agent identification (Agent) [110]. Using this data a number of processing methods for session retrieval have been used previously. Traditional heuristics start from a partition of the Log register set by grouping according to the same IP number and Agent description. An IP number and agent can not uniquely describe a web user since Internet Service Provider (ISP) multiplex the same IP number over several users employing the network address protocol (NAT).

Time Oriented Heuristic
One of the most popular methods for session retrieval is based on a maximum time a user stays on a web site. After partitioning by IP and Agent, the register is further sliced considering the accumulated time of the visit. There is a tradition of using 30 minutes for the time window [94]. This time is based on empirical experience of the web data mining community. Nowadays, this number does not carry an explanation, but it does seem to provide reasonable results [48]. Another less used approach is to limit the time spent per page.

Web Site Topology Oriented Heuristics
The site topology motivates another heuristic where web users strictly follow the hyperlink structure of a web site. If a register can not be reached from the last register in a web log, it begings a new session [20, 94]. The heuristic scans registers with the same IP and Agent, starting a new session each time a register can not be followed by the previous (Figure 2.5). Of course, this does not uniquely identify the individual path and such a heuristic encounters difficulty when two or more users follow the same path more or less at the same time [25]. This is the case for web sites that have frequently accessed content with only a fewer ways of accessing it. For example, the financial news from a news web site. When a browser or proxy cache is activated, "path competition" can be used to reconstruct the missing registers and conforming a session [94] by selecting the shortest path for the missing registers. If

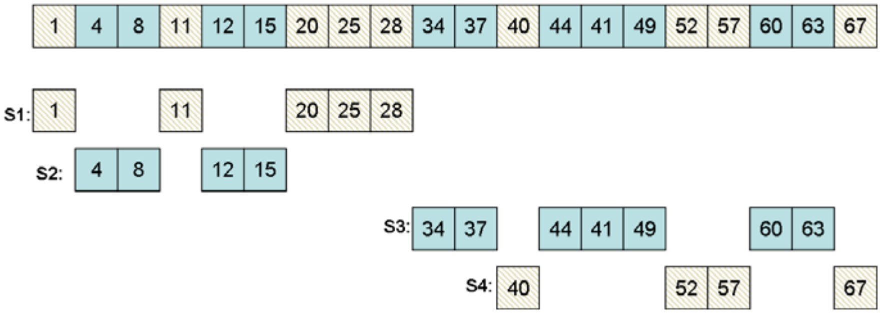

Fig. 2.4 Time Based Sessionization example: a log register indexed by time (second) is segmented in two groups (IP/Agent) generating four session. A timeout occurs after registers 15, 28 and 63

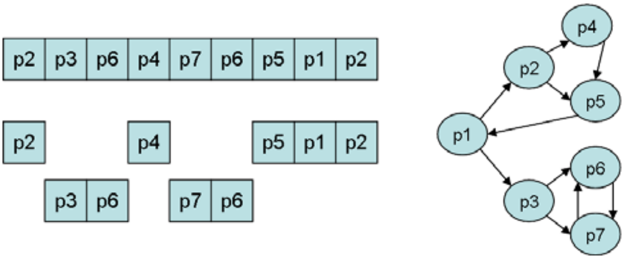

Fig. 2.5 Topology Based Sessionization example: Given the sequence of pages in the log register (left hand) and the web page's structure (right hand) if a page (p3) does not follows the web site hyperlink structure then it start a new session (p4,p7,p5)

the referred field of the log file is activated, the path competition heuristic can be enhanced because the previous pages are provided [65].

2.5.3.2 Ontology Based Sessionization

This method consists of enriching web registers with the semantic information of the visited URL. It is based on the assumption that each web user has a defined purpose that is reflected in the sequence of pages which one visited. The ontology description of each URL must be stated before [52, 51] and a semantic distance matrix Δ_{ij} is calculated from the URL i to j. Sessions are constructed by including the URL that has the nearest semantic distance. Another approach consists of defining predefined sub-trails [55] categorized by ontology.

2.5.3.3 Integer Programming Models

Integer programming is a well known technique used in engineering and operation research for modeling problems that have a combinatorial nature. Substantial progress has been archieved in this particular area in the last 10 years at improving solution time. In the past problemas that took months to resolve can now be solved in minutes [9]. The sessionization problem is inherently combinatoria. Two optimization models for sessionization are presented in this section [25]. The optimization models group log registers from the same IP address and agent. Additional restrictions ensure the link structure of the site is followed in any constructed session. Unlike the traditional heuristics that construct the sessions one at a time, optimization models constructs all sessions simultaneously. Each constructed session from a web server log is an ordered list of log registers where each register can only be used once in only one session. In the same session, register $r1$ can be an immediate predecessor of $r2$ if: the two registers share the same IP address and agent; a link exists from the page requested by $r1$ to the page requested by $r2$; and the request time for register $r2$ is within an allowable time window.

Maximizing a total reward per session

The Sessionization Integer Program (SIP) considers a binary set of variables $\{X_{ros}\}$ that have value one if register r is part of session s in the position o and zero otherwise. If maximizing the number of sessions is considered then the solution is equal to the number of log registers. In this case, a session with only one register satisfies all the previous restriction. Earlier studies on integer programming sessionization [25, 26] shows that considering a linear combination of variables as an objective function for maximization provides results in accordance within the Zipf law for session size (Section 2.2).

$$Maximize \quad \textstyle\sum_{ros} C_{ro} X_{ros}$$

$$Subject\ to:$$

$$\sum_{os} X_{ros} = 1 \qquad \forall r \tag{2.4}$$

$$\sum_{r} X_{ros} \leqslant 1 \qquad \forall o,s \tag{2.5}$$

$$X_{r,o+1,s} \leqslant \textstyle\sum_{r' \in bpage} X_{r'os} \quad \forall r,o,s \tag{2.6}$$

$$X_{ros} \in \{0,1\} \qquad \forall r,o,s$$

$$X_{ros} = 0 \qquad \forall s,\ r \in first,\ o > 1$$

The concise formulation of the optimization problem is given in the previous set of equations with a set of restriction from 2.4 to 2.6. The set $bpage_r$ correspond to the set of register r' that can be immediately before the register r in the same session. This is based on the hyperlink precedence of pages, the IP address matching between r and r', the agent matching and time precedence. The set $first$ contains register labels that can be first in a session, and can easily be formed by the condition $r \in first\ if\ bpage_r = \phi$.

The objective function can be understood as a total reward of $\sum_{r,o\ L} C_{ro}$ per session of size L. The constant C_{ro} should be a monotonic function in o and should reward larger session to avoid the attraction of singleton sessions.

Minimizing the number of sessions
Sessions can be represented as a network flow problem in a directed graph of nodes representing web log registers. The authors in [27] construct a bipartite cardinality matching where (in polynomial time) a solution corresponds to the minimum number of possible sessions can be found for a given web log.

2.5.3.4 Session Analysis

Using integer programming for extracting web user sessions from a web log has the advantage of being a flexible framework for encountering variations of additional information regarading some sessions. Some example follow.

The maximum number of copies of a given session
Finding the maximum number of copies of a given session enable the exploration of any fixed sequence of visited pages. Those patterns are interesting for the study of user habits on a web site. Specific patterns can be tested and ranked by the maximum number of possible sessions for a given web log. Higher ranked patterns should be considered as the most likely sequence of pages from a web log. The formulation correspond to a network flow problem with side constraints [27]. This can be solved quickly, allowing solution of a large number of different patterns within a reasonable time.

The maximum number of sessions of a given size
By adjusting $C_{ro} = 0\ \forall\ r,o \neq l$ and $C_{ro} = 1\ \forall\ r,o = l$, one can find the maximum number of sessions of a given size l. This can be used as a measure for the capacity of a web site to capture the attention of the web user.

The maximum number of sessions that pass by a given page in a given order
Some pages on the web site are considered important for navigation and should be present in a large number of sessions. The maximal reward per session can be modified by adjusting $C_{ro} = 1$ when the register r relates to the page and o is the order, and $C_{ro} = 0$ otherwise.

Considering the effect of cache devices on web logs
As presented in section 2.2 the use of the back button in web browsing produces missing registers in web logs. The maximal reward per session can be modified by introducing a new variable Y_{ros} corresponding to the use of a back button action [26]. A session in this case could be identified to include a session with back button usage and constraints can be added on the total number (or percent) of sessions which includes the use of the back button.

2.5.4 Sessions in Dynamic Environments

Static assumptions for web usage mining are a common hypothesis for data preprocessing. However in web application like CRM and recommender systems dynamic content is not the exception but the rule. In this dynamic context, the notion of dynamic URL becomes a valid representation [76]. Query parameter values from the URL and application database contribute to map the content with the dynamic URL, and convert it into a hierarchical semantically structure $(i, j)|i : parent, \ j : child$ of semantic label.

The evolution of a web site produces changes in the behavior of the web users. Thus, web user sessions should be comparable to a period of time where changes can be considered minimal. Those periods $\{T_1, ..., T_k\}$ depend on web site managers' updating procedures which need to be defined. Then, a given period of sessionization has to be performed using the available semantic labels in the page. Periods of sessionization can then be compared to the analysis of the user profile evolution [76]. The use of semantic labelling provides a comparison between the sessions belonging to different version of the same web site.

2.5.5 Identifying Session Outliers

An important phase of data pre-processing is data cleaning. A first stage of the sessionization process cleans the Log file erasing register from robot, viruses/worms, hacking attempt and others. But in general not all of this unwanted data can be removed. The sessions set could be refined detecting different modes of uncommon behavior [88, 52]. Some recent studies [88] use the 1% tail of the Malanobis distance to locate sessions indicating unusual modes of behaviors.Others studies relate to semantic characteristics of the sessions [52] which are obtained using an Ontology-Oriented heuristic.

2.6 Summary

Web data is a complex and noisy data source, yet empirical regularity rules its statistical description. Several pre-processing techniques have been developed in the last ten years to support web mining and address the changing characteristics of the Web. The primary web data extracted are the hyperlink structure, the content, and the usage of a web site. All have some collection issues. A web crawler collects the hyperlink structure but the time changing characteristic of the Web must be handled with page selection methods, revisit schedules, a politeness policy, and parallel processing. The challenges with web content are determining a weighting scheme as well as dealing with the visual representation and the dynamism of a website. Web usage data can be obtained indirectly from web logs or by direct retrieval. Integer programming has recently shown promise as an indirect method. Data preparation is a significant effort and a necessary cornerstone for web mining.

Acknowledgements. This work was supported partially by the FONDEF project DO8I-1015 and the *National Doctoral Grant* from *Conicyt Chile*. We are grateful to Luis E. Dujovne for very fruitful discussion on Web Site Key Objects.

References

1. Adar, E., Teevan, J., Dumais, S., Elsas, J.: The web changes everything: understanding the dynamics of web content. In: WSDM 2009: Proceedings of the Second ACM International Conference on Web Search and Data Mining, pp. 282–291. ACM Press, New York (2009)
2. Alexander, J.: Understanding and improving navigation within electronic documents. Ph.D. thesis, University of Canterbury, Christchurch, New Zealand (2009)
3. Alexander, J., Cockburn, A.: An empirical characterisation of electronic document navigation. In: GI 2008: Proceedings of graphics interface 2008, pp. 123–130. Canadian Information Processing Society, Toronto (2008)
4. ASA, O.S.: Opera browser, `http://www.opera.com`
5. Baeza-Yates, R., Castillo, C., Efthimiadis, E.: Characterization of national web domains. ACM Transactions on Internet Technology 7(2) (2007)
6. Baeza-Yates, R., Poblete, B.: Dynamics of the chilean web structure. Comput. Netw. 50(10), 1464–1473 (2006)
7. Bayir, M., Toroslu, I., Cosar, A., Fidan, G.: Smart miner: a new framework for mining large scale web usage data. In: WWW 2009: Proceedings of the 18th international conference on World wide web, pp. 161–170. ACM Press, New York (2009)
8. Bhamidipati, N.L., Pal, S.K.: Stemming via distribution-based word segregation for classification and retrieval. IEEE Transactions on Systems, Man, and Cybernetics, Part B 37(2), 350–360 (2007)
9. Bixby, R.E.: Solving real-world linear programs: A decade and more of progress. Operations Research 50(1), 3–15 (2002)
10. Burget, R., Rudolfova, I.: Web page element classification based on visual features. In: Asian Conference on Intelligent Information and Database Systems, vol. 0, pp. 67–72 (2009)
11. Castells, P., Fernandez, M., Vallet, D.: An adaptation of the vector-space model for ontology-based information retrieval. IEEE Trans. on Knowl. and Data Eng. 19(2), 261–272 (2007)
12. Castillo, C.: Effective web crawling. Ph.D. thesis, University of Chile, Santiago, Chile (2004)
13. Catledge, L.D., Pitkow, J.E.: Characterizing browsing strategies in the world-wide web. In: Computer Networks and ISDN Systems, pp. 1065–1073 (1995)
14. Chakrabarti, D., Faloutsos, C.: Graph mining: Laws, generators, and algorithms. ACM Comput. Surv. 38(1), 2 (2006)
15. Chakrabarti, D., Kumar, R., Punera, K.: Page-level template detection via isotonic smoothing. In: WWW 2007: Proceedings of the 16th international conference on World Wide Web, pp. 61–70. ACM Press, New York (2007)
16. Chakrabarti, S., Dom, B.E., Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., Kleinberg, J.: Mining the web's link structure. Computer 32(8), 60–67 (1999)

17. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: a distributed storage system for structured data. In: OSDI 2006: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation, p. 15. USENIX Association, Berkeley (2006)
18. Cho, J., Garcia-Molina, H.: The evolution of the web and implications for an incremental crawler. In: VLDB 2000: Proceedings of the 26th International Conference on Very Large Data Bases, pp. 200–209. Morgan Kaufmann Publishers Inc., San Francisco (2000)
19. Cho, J., Garcia-Molina, H.: Estimating frequency of change. ACM Trans. Internet Technol. 3(3), 256–290 (2003)
20. Cooley, R., Mobasher, B., Srivastava, J.: Data preparation for mining world wide web browsing patterns. Knowledge and Information Systems 1, 5–32 (1999)
21. Corporation, M.: Mozilla firefox browser, `http://www.mozilla.org`
22. Coull, S.E., Collins, M.P., Wright, C.V., Monrose, F., Reiter, M.K.: On web browsing privacy in anonymized netflows. In: SS 2007: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, pp. 1–14. USENIX Association, Berkeley (2007)
23. Das, R., Turkoglu, I.: Creating meaningful data from web logs for improving the impressiveness of a website by using path analysis method. Expert Syst. Appl. 36(3), 6635–6644 (2009)
24. Debnath, S., Mitra, P., Pal, N., Giles, C.L.: Automatic identification of informative sections of web pages. IEEE Trans. on Knowl. and Data Eng. 17(9), 1233–1246 (2005)
25. Dell, R.F., Román, P.E., Velásquez, J.D.: Web user session reconstruction using integer programming. In: Procs. of The 2008 IEEE/WIC/ACM International Conference on Web Intelligence, Sydney, Australia, pp. 385–388 (2008)
26. Dell, R.F., Román, P.E., Velásquez, J.D.: User session reconstruction with back button browsing. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based and Intelligent Information and Engineering Systems. LNCS, vol. 5711, pp. 326–332. Springer, Heidelberg (2009)
27. Dell, R.F., Román, P.E., Velásquez, J.D.: Optimization models for construction of web user sessions. Working Paper (2010)
28. Demartini, G., Firan, C.S., Iofciu, T., Nejdl, W.: Semantically enhanced entity ranking. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) WISE 2008. LNCS, vol. 5175, pp. 176–188. Springer, Heidelberg (2008)
29. Demir, G.N., Goksedef, M., Etaner-Uyar, A.S.: Effects of session representation models on the performance of web recommender systems. In: ICDEW 2007: Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop, pp. 931–936. IEEE Computer Society Press, Washington (2007)
30. Desikan, P., Srivastava, J.: Mining temporally evolving graphs. In: Mobasher, B., Nasraoui, O., Liu, B., Masand, B. (eds.) WebKDD 2004. LNCS (LNAI), vol. 3932, pp. 1–17. Springer, Heidelberg (2004)
31. Dill, S., Kumar, R., Mccurley, K., Rajagopalan, S., Sivakumar, D., Tomkins, A.: Self-similarity in the web. ACM Trans. Internet Technol. 2(3), 205–223 (2002)
32. Dujovne, L.E., Velásquez, J.D.: Design and implementation of a methodology for identifying website keyobjects. In: Velásquez, J.D., Ríos, S.A., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based and Intelligent Information and Engineering Systems. LNCS, vol. 5711, pp. 301–308. Springer, Heidelberg (2009)
33. Eguchi, S., Copas, J.: Interpreting kullback-leibler divergence with the neyman-pearson lemma. J. Multivar. Anal. 97(9), 2034–2040 (2006)

34. Fetterly, D., Manasse, M., Najork, M., Wiener, J.: A large-scale study of the evolution of web pages. In: WWW 2003: Proceedings of the 12th international conference on World Wide Web, pp. 669–678. ACM Press, New York (2003)
35. Gaugaz, J., Zakrzewski, J., Demartini, G., Nejdl, W.: How to trace and revise identities. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 414–428. Springer, Heidelberg (2009)
36. Ghani, R., Jones, R., Mladenic, D.: Mining the web to create minority language corpora. In: CIKM 2001: Proceedings of the tenth international conference on Information and knowledge management, pp. 279–286. ACM Press, New York (2001)
37. Görnitz, N., Kloft, M., Brefeld, U.: Active and semi-supervised data domain description. In: ECML PKDD 2009: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 407–422. Springer, Heidelberg (2009)
38. Granka, L., Feusner, M., Lorigo, L.: Eye monitoring in online search. In: Hammoud, R., Ohno, T. (eds.) Passive Eye Monitoring, Signals and Communication Technology, Part VI, pp. 347–372. Springer, Heidelberg (2008)
39. Gündüz, C., Özsu, M.T.: A web page prediction model based on click-stream tree representation of user behavior. In: KDD 2003: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 535–540. ACM Press, New York (2003)
40. Hand, D.: Statistics and data mining: intersecting disciplines. SIGKDD Explor. Newsl. 1(1), 16–19 (1999)
41. Hensman, S.: Construction of conceptual graph representation of texts. In: HLT-NAACL 2004: Proceedings of the Student Research Workshop at HLT-NAACL 2004, vol. XX, pp. 49–54. Association for Computational Linguistics, Morristown (2004)
42. Huberman, B., Pirolli, P., Pitkow, J., Lukose, R.M.: Strong regularities in world wide web surfing. Science 280(5360), 95–97 (1998)
43. Huberman, B., Wu, F.: The economics of attention: maximizing user value in information-rich environments. In: ADKDD 2007: Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising, pp. 16–20. ACM Press, New York (2007)
44. Iachello, G., Hong, J.: End-user privacy in human-computer interaction. Found. Trends Hum.-Comput. Interact. 1(1), 1–137 (2007)
45. Ipeirotis, P., Gravano, L.: When one sample is not enough: improving text database selection using shrinkage. In: SIGMOD 2004: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pp. 767–778. ACM Press, New York (2004)
46. Janzen, S., Maass, W.: Ontology-based natural language processing for in-store shopping situations. In: ICSC 2009: Proceedings of the 2009 IEEE International Conference on Semantic Computing, pp. 361–366. IEEE Computer Society, Washington (2009)
47. Jatowt, A., Ishizuka, M.: Temporal multi-page summarization. Web Intelli. and Agent Sys. 4(2), 163–180 (2006)
48. Velásquez, J.D., Palade, V.: Adaptive web sites: A knowledge extraction from web data approach. IOS Press, Amsterdam (2008)
49. Jin, W., Srihari, R.K.: Graph-based text representation and knowledge discovery. In: SAC 2007: Proceedings of the 2007 ACM symposium on Applied computing, pp. 807–811. ACM, New York (2007)

50. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., Gay, G.: Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. ACM Trans. Inf. Syst. 25(2), 7 (2007)
51. Jung, J.J.: Ontology-based partitioning of data steam for web mining: A case study of web logs. In: ICCS 2004, 4th International Conference, Proceedings, Part I, June 6-9, 2004, Kraków, Poland, pp. 247–254 (2004)
52. Jung, J.J., Jo, G.S.: Semantic outlier analysis for sessionizing web logs. In: ECML/P-KDD Conference, pp. 13–25 (2004)
53. Ke, Y., Deng, L., Ng, W., Lee, D.: Web dynamics and their ramifications for the development of web search engines. Comput. Netw. 50(10), 1430–1447 (2006)
54. Khan, J.I., Tao, Q.: Exploiting webspace organization for accelerating web prefetching. Web Intelli. and Agent Sys. 3(2), 117–129 (2005)
55. Khasawneh, N., Chan, C.: Active user-based and ontology-based web log data preprocessing for web usage mining. In: 2006 IEEE / WIC / ACM International Conference on Web Intelligence (WI 2006), Hong Kong, China, pp. 325–328. IEEE Computer Society, Los Alamitos (2006)
56. Kim, Y., Kim, J.: Web prefetching using display-based prediction. In: WI 2003: Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence, p. 486. IEEE Computer Society, Washington (2003)
57. Kohonen, T.: Self-organized formation of topologically correct feature maps, pp. 509–521 (1988)
58. Kryssanov, V., Kakusho, K., Kuleshov, E., Minoh, M.: Modeling hypermedia-based communication. Information Sciences 174(1-2), 37–53 (2005)
59. Lan, M., Tan, C.L., Low, H.B., Sung, S.Y.: A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In: WWW 2005: Special interest tracks and posters of the 14th international conference on World Wide Web, pp. 1032–1033. ACM Press, New York (2005), http://doi.acm.org/10.1145/1062745.1062854
60. Lan, M., Tan, C.L., Su, J., Lu, Y.: Supervised and traditional term weighting methods for automatic text categorization. IEEE Trans. Pattern Anal. Mach. Intell. 31(4), 721–735 (2009)
61. Langford, D.: Internet ethics. MacMillan Press Ltd., Basingstoke (2000)
62. Lansey, J.C., Bukiet, B.: Internet search result probabilities, heaps' law and word associativity. Journal of Quantitative Linguistics 16(1), 40–66 (2005)
63. Leijenhorst, D.V., der Weide, T.V.: A formal derivation of heaps' law. Inf. Sci. Inf. Comput. Sci. 170(2-4), 263–272 (2005)
64. Levene, M., Borges, J., Loizou, G.: Zipf's law for web surfers. Knowl. Inf. Syst. 3(1), 120–129 (2001)
65. Li, Y., Feng, B., Mao, Q.: Research on path completion technique in web usage mining. In: International Symposium on Computer Science and Computational Technology, vol. 1, pp. 554–559 (2008)
66. Linn, J.: Technology and web user data privacy: A survey of risks and countermeasures. IEEE Security and Privacy 3(1), 52–58 (2005)
67. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications), 1st edn. (2007); corr. 2nd printing edn. Springer, Heidelberg (2009)
68. Manning, C.D., Schutze, H.: Fundation of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
69. Maynor, D.: Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research, 1st edn. Syngress (2007)

70. Mobasher, B.: Web usage mining. In: Liu, B. (ed.) Web Data Mining: Exploring Hyperlinks, Contents and Usage Data, ch. 12. Springer, Heidelberg (2006)
71. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Effective personalization based on association rule discovery from web usage data. In: WIDM 2001: Proceedings of the 3rd international workshop on Web information and data management, pp. 9–15. ACM Press, New York (2001)
72. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Discovery and evaluation of aggregate usage profiles for web personalization. Data Min. Knowl. Discov. 6(1), 61–82 (2002)
73. Moloney, M., Bannister, F.: A privacy control theory for online environments. In: HICSS 2009: Proceedings of the 42nd Hawaii International Conference on System Sciences, pp. 1–10. IEEE Computer Society, Washington (2009)
74. Mori, T.: Information gain ratio as term weight: the case of summarization of ir results. In: Proceedings of the 19th international conference on Computational linguistics, pp. 1–7. Association for Computational Linguistics, Morristown, NJ, USA (2002)
75. Nadeax, D.: Semi-supervised named entity recognition: Learning to recognize 100 entity types with little supervision. Ph.D. thesis, University of Ottawa, Ottawa, Canada (2007)
76. Nasraoui, O., Soliman, M., Saka, E., Badia, A., Germain, R.: A web usage mining framework for mining evolving user profiles in dynamic web sites. IEEE Trans. on Knowl. and Data Eng. 20(2), 202–215 (2008)
77. Obendorf, H., Weinreich, H., Herder, E., Mayer, M.: Web page revisitation revisited: implications of a long-term click-stream study of browser usage. In: CHI 2007: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 597–606 (2007)
78. Olston, C., Pandey, S.: Recrawl scheduling based on information longevity. In: WWW 2008: Proceeding of the 17th international conference on World Wide Web, pp. 437–446. ACM Press, New York (2008)
79. Pal, S.K., Talwar, V., Mitra, P.: Web mining in soft computing framework: Relevance, state of the art and future directions. IEEE Transactions on Neural Networks 13, 1163–1177 (2002)
80. Peña-Ortiz, R., Sahuquillo, J., Pont, A., Gil, J.: Dweb model: Representing web 2.0 dynamism. Comput. Commun. 32(6), 1118–1128 (2009)
81. Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A.: Kim – a semantic platform for information extraction and retrieval. Nat. Lang. Eng. 10(3-4), 375–392 (2004)
82. Porter, M.F.: An algorithm for suffix stripping. Electronic Library and Electronic Systems 40, 211–218 (2006)
83. Qi, X., Davison, B.: Web page classification: Features and algorithms. ACM Comput. Surv. 41(2), 1–31 (2009)
84. Radlinski, F., Kurup, M., Joachims, T.: How does clickthrough data reflect retrieval quality? In: CIKM 2008: Proceeding of the 17th ACM conference on Information and knowledge management, pp. 43–52. ACM Press, New York (2008)
85. Reay, I.K., Beatty, P., Dick, S., Miller, J.: A survey and analysis of the p3p protocol's agents, adoption, maintenance, and future. IEEE Transactions on Dependable and Secure Computing 4, 151–164 (2007)
86. Román, P.E., Velásquez, J.D.: Dynamic stochastic model applied to the analysis of the web user behavior. In: 6th Atlantic Web Intelligence Conference, AWIC 2009, Prague, CZECH Republic, pp. 31–40 (2009)
87. Rugaber, S., Harel, N., Govindharaj, S., Jerding, D.: Problems modeling web sites and user behavior. In: WSE 2006: Proceedings of the Eighth IEEE International Symposium on Web Site Evolution, pp. 83–94. IEEE Computer Society Press, Washington (2006)

88. Sadagopan, N., Li, J.: Characterizing typical and atypical user sessions in clickstreams. In: WWW 2008: Proceeding of the 17th international conference on World Wide Web, pp. 885–894. ACM Press, New York (2008)
89. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34(1), 1–47 (2002)
90. Shehata, S.: A wordnet-based semantic model for enhancing text clustering. In: ICDMW 2009: Proceedings of the 2009 IEEE International Conference on Data Mining Workshops, pp. 477–482. IEEE Computer Society, Washington (2009)
91. Snásel, V., Kudelka, M.: Web content mining focused on named objects. In (IHCI) First International Conference on Intelligent Human Computer Interaction, pp. 37–58. Springer, India (2009)
92. Soares, M.V.B., Prati, R.C., Monard, M.C.: Improvement on the porter's stemming algorithm for portuguese. IEEE Latin America Transaction 7(4), 472–477 (2009)
93. Spaniol, M., Denev, D., Mazeika, A., Weikum, G., Senellart, P.: Data quality in web archiving. In: WICOW 2009: Proceedings of the 3rd workshop on Information credibility on the web, pp. 19–26. ACM Press, New York (2009)
94. Spiliopoulou, M., Mobasher, B., Berendt, B., Nakagawa, M.: A framework for the evaluation of session reconstruction heuristics in web-usage analysis. Informs Journal on Computing 15(2), 171–190 (2003)
95. Tauscher, L., Greenberg, S.: Revisitation patterns in world wide web navigation. In: Procs. of the Conference on Human Factors in Computing Systems, Atlanta, USA, pp. 22–27 (1997)
96. Tsatsaronis, G., Varlamis, I., Nørvg, K.: An experimental study on unsupervised graph-based word sense disambiguation. In: Gelbukh, A. (ed.) Computational Linguistics and Intelligent Text Processing. LNCS, vol. 6008, pp. 184–198. Springer, Heidelberg (2010)
97. Ullrich, C., Borau, K., Luo, H., Tan, X., Shen, L., Shen, R.: Why web 2.0 is good for learning and for research: principles and prototypes. In: WWW 2008: Proceeding of the 17th international conference on World Wide Web, pp. 705–714. ACM Press, New York (2008)
98. Urbansky, D., Feldmann, M., Thom, J.A., Schill, A.: Entity extraction from the web with webknox. In: 6th Atlantic Web Intelligence Conference (AWIC), Prague, Czech Republic (2009)
99. Velásquez, J.D., Yasuda, H., Aoki, T., Weber, R., Vera, E.: Using self organizing feature maps to acquire knowledge about visitor behavior in a web site. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS, vol. 2773, pp. 951–958. Springer, Heidelberg (2003)
100. Wang, J., Wu, X., Zhang, C.: Support vector machines based on kmeans clustering for real time business intelligence systems. Int. J. Bus. Intell. Data Min. 1(1), 54–64 (2005)
101. Wang, Y., Hodges, J.: Document clustering with semantic analysis. In: HICSS 2006: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, p. 54.3. IEEE Computer Society, Washington (2006)
102. Weinreich, H., Obendorf, H., Herder, E., Mayer, M.: Off the beaten tracks: exploring three aspects of web navigation. In: WWW 2006: Proceedings of the 15th international conference on World Wide Web, pp. 133–142. ACM Press, New York (2006)
103. Weinreich, H., Obendorf, H., Herder, E., Mayer, M.: Not quite the average: An empirical study of web use. ACM Trans. Web 2(1), 1–31 (2008)
104. White, R.W.: Investigating behavioral variability in web search. In. Proc. WWW, pp. 21–30 (2007)

105. Wittek, P., Darányi, S., Tan, C.L.: Improving text classification by a sense spectrum approach to term expansion. In: CoNLL 2009: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, pp. 183–191. Association for Computational Linguistics, Morristown (2009)
106. Won, S., Jin, J., Hong, J.: Contextual web history: using visual and contextual cues to improve web browser history. In: CHI 2009: Proceedings of the 27th international conference on Human factors in computing systems, pp. 1457–1466. ACM Press, New York (2009)
107. Yan, X., Zhang, C., Zhang, S.: Toward databases mining: Pre-processing collected data. Applied Artificial Intelligence 17(5-6), 545–561 (2003)
108. Yu, L., Wang, S., Lai, K.: An integrated data preparation scheme for neural network data analysis. IEEE Transactions on Knowledge and Data Engineering 18, 217–230 (2006)
109. Yue, C., Xie, M., Wang, H.: Automatic cookie usage setting with cookiepicker. In: DSN 2007: Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 460–470. IEEE Computer Society Press, Washington (2007)
110. Zawodny, J.D.: Linux apache web server administration. Sybex, 2 edn. (2002)
111. Zhang, Z., Chen, J., Li, X.: A preprocessing framework and approach for web applications. J. Web Eng. 2(3), 176–192 (2004)