FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Reverse Engineering Static Content and Dynamic Behaviour of E-Commerce Sites for Fun and Profit

João Pedro Matos Teixeira Dias

DISSERTATION PLANNING

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Hugo Sereno Ferreira

Second Supervisor: Rui Peixoto

February 12, 2016

Reverse Engineering Static Content and Dynamic Behaviour of E-Commerce Sites for Fun and Profit

João Pedro Matos Teixeira Dias

Mestrado Integrado em Engenharia Informática e Computação

February 12, 2016

Abstract

Nowadays electronic commerce websites are one of the main transaction tools between on-line merchants and consumers or businesses. These e-commerce websites rely heavily on summarizing and analyzing the behavior of customers, making an effort to influence user actions towards the optimization of success metrics such as CTR (Click through Rate), CPC (Cost per Conversion), Basket and Lifetime Value and User Engagement. Knowledge extraction from the existing e-commerce websites datasets, using data mining and machine learning techniques, have been greatly influencing the Internet marketing activities.

When faced with a new e-commerce website, the machine learning practitioner starts a web mining process by collecting historical and real-time data of the website and analyzing/transforming this data in order to be capable of extracting information about the website structure and content and its users' behavior. Only after this process the data scientists are able to build relevant models and algorithms to enhance marketing activities.

This is an expensive process in resources and time since that it will depend always on the condition in which the data is presented to the data scientist, since data with more quality (i.e. no incomplete data) will make the data scientist work easier and faster. On the other hand, in most of the cases, data scientists would usually resort to tracking domain-specific events throughout a user's visit to the website in order to fulfill the objective of discovering the users' behavior and, for this, it is necessary code modifications to the pages themselves, that will result in a larger risk of not capturing all the relevant information by not enabling tracking mechanisms. For example, we may not know apriori that a visit to a Delivery Conditions page is relevant to the prediction of a user's willingness to buy and therefore would not enable tracking on those pages.

Within this problem context, the proposed solution consists of a tool capable of extracting and combining information about a e-commerce website through a process of web mining, comprehending the structure as well as the content of the website pages, relying mostly on identifying dynamic content and semantic information in predefined locations, complemented with the capability of, using the user's access logs, be capable of extracting more accurate models to predict the users future behavior. This will permit the creation of a data model representing an e-commerce website and its archetypical users that can be useful, for example, in simulation systems.

"Part of the inhumanity of the computer is that, once it is competently programmed and working smoothly, it is completely honest."

Isaac Asimov

Contents

CONTENTS

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

Abbreviations

WWW	*World Wide Web*
B2C	Business-to-consumer
B2B	Business-to-business
e-commerce	Electronic Commerce
CTR	Click-through rate
CR	Conversion rate
WUM	Web usage mining
WSM	Web structure mining
WCM	Web content mining
HTML	HyperText Markup Language
EDA	Exploratory Data Analysis
SVM	Supported Vector Machines
*k*NN	*k*-Nearest-Neighbour
GATC	Google Analytics Tracking Code
URL	Uniform Resource Locator
CSP	Contiguous Sequential Pattern
W3C	World Wide Web Consortium

Chapter 1

Introduction

Nowadays, thanks to the development of electronic commerce, a great percentage of the most basic of economic transactions - the buying and selling of goods - take place over electronic networks, mostly the Internet, and this business is still growing [Sta16].

When a user enters an e-commerce website there are used marketing and advertising mechanisms that tries to influence the user behaviour in order to increase profit. This mechanisms relay heavily on summarizing and analysing the behaviour of costumers. Data mining and machine learning techniques have been applied towards the development of this mechanisms, making great significance in Internet marketing activities, and improving e-commerce sales and profits.

There are a considerable number of companies today who develop and provide tools and mechanisms to e-commerce company owners in order to make them capable of improving their sales and take better data-driven decisions (i.e. marketing decisions). These tools are used with the objective to track and learn costumers habits and are most of the times complemented with automatic recommendation systems. This makes e-commerce company owners more capable of taking action in target marketing and advertisement, with for example, recommending the most interesting product for each visitor, in order to increase profits.

1.1 Motivation

When faced with a new e-commerce website, the machine learning practitioner typically spends a great amount of time collecting and analysing the static and dynamic data of the website. This process is essential to extract useful information about the website's structure, content and its users' behavior. Only after this process will the machine learning partitioner be able to build relevant models and algorithms to enhance online marketing activities. This is a great challenge because of the heterogeneous nature of the data, the semi-structured or even unstructured way that data is presented and the data existent is so vast that we can easily get overwhelmed by it [ZS08].

The process of extracting information from a Web site's structure, content and users is mostly a repetitive and semi-automated task, which is reflected in the necessity of allocating dedicated resources to it, and, in some cases, there is a need of analysing the website for a certain period of time after it comes to the machine learning practitioner for data collection proposes may imply a waste of time. Besides that there is always a greater risk of relevant knowledge that may exist between data sources. When improving this process to an automatic one we reduce the costs as well as the risks of losses of information.

1.2 Aim and Goals

The aim of this dissertation is to develop a methodology capable of, given an e-commerce website, extracting useful information from its structure, content and its typical users' behavior using web mining techniques, returning a consistent model representing the website content, relations between pages and its archetypical users. This methodology has the objective of reducing the time and resources required to analyse an e-commerce website and of improving the extraction of knowledge, making possible relationships between a website structure, content and usage more clear to the data scientist.

The expected final result of the present work is a functional prototype of a tool, as the methodology's *proof of concept*, capable of doing web mining and that is ready to be used by a data scientist in order to get a grasp of the site's layout and its users', aiming to make the data collection task more efficient, easy and concise.

1.3 Expected Contributions

We propose a study of a new methodology for extracting and representing the information present on a given e-commerce website. The website pages and connection between them are mapped into a data structure, as well the navigational habits from its users. This process will consist of a *all-in-one* approach, trying to mitigate possible data relationships losses from the typical, an partially separate, tasks used to retrieve this data.

The contributions of this dissertation can be summarize in the following points:

1. An *all-in-one* approach to extract information from an e-commerce website content, structure and its archetypical users;
2. A concise representation of this information on a consistent model in order to make the knowledge access and retrieval more simple and easy;
3. Mitigate possible flaws when collecting data, establishing relationships and connections that can be otherwise pass unnoticed to the data scientist.

1.4 Structure of this Dissertation

In Chapter 2, it's provided a contextualization and background review of the related work in web mining and user profiling fields, going through web mining taxonomy, user profiling techniques, e-commerce peculiarities and the importance success metrics on e-commerce. Chapter 3 presents, on one hand, the proposed methodology to retrieve a website static content and its representation, and on the other hand, the archetypical website users through the representation of statistical models, user flows and usage patterns. Concluding remarks and further work is presented in Chapter 4.

Chapter 2

Literature Review

2.1 E-commerce Overview

2.1.1 Introduction

Electronic commerce has become an essential tool for small and large businesses worldwide, not only as a tool of selling goods directly to consumers (B2C) or to other businesses (B2B), but also as a way of engaging them [EK08].

> "Electronic commerce is the process of buying, selling, transferring, or exchanging products, services, and/or information via computer networks, including the Internet." [TK11]

E-commerce is one of the innovations with most impact over the consumer habits, mostly due to his unique features. E-commerce benefits of its ubiquity nature, the WWW is always available and everywhere, through desktops, mobiles and tablets. Besides that, the richness of information presented to the user make e-commerce a great bridge between merchants and customers [Hyd05].

When talking about e-commerce business is obligatory to mention Amazon[1] and eBay[2], which were among the first Internet companies to make available electronic transactions to the users. But, as of today, this kind of transactions are available everywhere on the WWW, even through social networks like Facebook[3]. Associated with this, it's increasingly common that marketing and advertising campaigns run over the Internet too [Moh12].

E-commerce companies commonly resort to the Web personalization techniques as a way of doing target marketing over its visitors based on each visitor individual characteristics with the goal of improving the likelihood of an visitor generate profit [Hyd05].

[1] http://www.amazon.com/

[2] http://ebay.com/

[3] http://facebook.com/

2.1.2 E-commerce Metrics

For a long time now, business community knows how to measure performance on traditional commerce based on number of sales and profit and what to expect from clients paying attention to metrics like *Costumer Lifetime Value* and *Costumer Engagement*. When talking about the Internet, there is a need of using metrics beyond the traditional ones, called *e-metrics*, with the propose of measuring an Web site success and improving its performance. Basic concepts like page views and unique visitors are already deployed by almost every Web site, but more advanced metrics that take to account the loyalty of an visitor or typical users habits are now becoming essential in order to increase the user engagement [SC00].

In e-commerce Web sites, besides the common metrics used to measure a Web site performance, there are some unique metrics that can give us a better and more direct overview of the online business performance, as follows:

- *Conversion rate* that is given by the number of buyers (or paying customers) over the total number of Web site users;

- *Click-through rate* stands for the ratio between the users who click on a specific link to the total number of page visitors;

- *Costumer retention* its the measure that shows the ability of a Web site retain customers over the long term [Gef02];

- *Shopping cart abandonment rate* that gives the number of times in which an item was added to the shopping cart but the order was not completed.

In [PV12] *Conversion rate* and *Click-through rate* are presented as the primary metrics when validating recommendation systems present in a Web site. An overview of recommendation systems and its impact is detailed in 2.1.3.

2.1.3 Recommendation Systems

E-commerce is a very convenient way for people do their shopping. However, it is sometimes a difficult task for customers to select a valuable item over the great number of various products available on a Web site. On the other hand, most of the e-commerce Web sites depend on this personalization capabilities to improve sales and profits. Here is were the personalization capabilities of an Web site for each customer become essential, giving the user suggestions on products, refining search results and targeting advertises. Recommendation mechanisms goal is to influence user actions towards the optimization of success *e-metrics*, maintaining the customer satisfaction.

A recommender system usually depends on a three step process, starting with data retrieval, normally resorting to Web mining techniques, were we get to know the user preferences by analysing static content and user behaviour data, followed by computing and validating the recommendation using proper techniques and finalizing by presenting the recommendation results to the customer [WHF07].

As Wei et al. [WHF07] suggests, the typical recommendation mechanisms are split into three approaches: collaborative filtering, content-based filtering and hybrid.

Collaborative filtering is one of the most widely adopted and successful recommendation approach technique. This technique bases itself on building a customer preference model based on their previous iterations, thus distancing itself from techniques that are based on intrinsic consumer and product characteristics [ZDH07].

Content-based recommendation systems are systems that, by analysing items description and details, identify items that fit in the user particular interests based on their user profile. An user profile can contain a number of different types of information, but it should contain a model of user preferences - description of items that the user expressed interest about - and a history of the user's interactions with the Web site, this can include information like what other items the user has viewed [PB07].

Additionally, as suggested by Wei et al. [WHF07] there is a third type of recommendation mechanism which consist in a hybrid filtering approach. This approach have combined both the collaborative filtering as the content-based recommendation methods. There are various approach when combining this two methods but they can mostly be classified into three methods. On of them consists on introducing some component or feature of one approach into the other one as designed by [MMN01]. Another kind consists on combining the result of recommendation of each approach into a single recommendation as proposed by [CMG$^+$99]. At last, another approach is to present a comprehensive and unique model depending on other information. One example of this method is presented in [PUPL01], where is used a probabilistic model technique.

2.2 Web Mining

2.2.1 Introduction

Data mining, the process of extracting hidden predictive information from large data sources, has been used by companies in order to focus on most important information present in their data sets. Data mining techniques give companies the possibility of predict future user trends and behaviours, allowing business to take better and data-driven decisions for their future actions. Data mining communities categorize three different types of mining: data mining, Web mining, and text mining [KB00].

Data mining mainly deals with structured data organized in a databases while text mining mainly handles unstructured data/text. Web mining lies in between and deals with semi-structured and/or unstructured data. Web mining conciliates data mining and/or text mining techniques applying this concepts to the WWW, extracting useful, and sometimes hidden, information and patterns present in Web documents and Web activities. In this way, Web mining focus on data like the content of Web pages, user access information, hyperlinks and other resources (i.e. multimedia) in order to retrieve intrinsic proprieties between data objects [MC10].

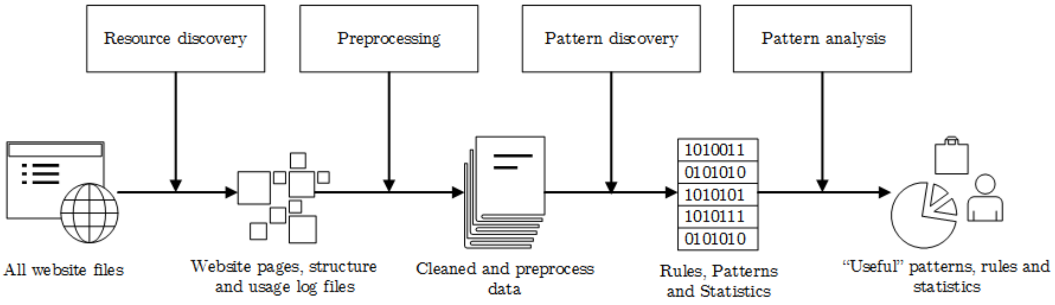

Resource discovery
Preprocessing
Pattern discovery
Pattern analysis
1010011
0101010
1010101
1010111
0101010
All website files
Website pages, structure and usage log files
Cleaned and preprocess data
Rules, Patterns and Statistics
"Useful" patterns, rules and statistics

Figure 2.1: Web mining process overview [MC10].

The WWW is a massive, explosive, diverse, dynamic, constantly growing and mostly unstructured data repository, which delivers an incredible amount of information, and also increases the complexity of dealing with the information from the different perspectives of the data scientist [KS10]. Due to this complexity, Web mining data present some challenges [CG05]:

- The Web is huge and Web pages are semi-structured or lack of structure at all;
- Web information tends to be diversity in meaning with lots of sources and ambiguity;
- The degree of quality of the information extracted;
- The reliability of knowledge retrieved from the information extracted.

Web mining can be considered a four-phase process, consisting of the data preparation, pattern discovery and pattern analysis phases, as identified in fig. 2.1 [MCS00]. After the information sources are discovered and collected, information is preprocessed with the objective of making the data more concise and accurate. The second stage, pattern discovery, consists on apply mathematical strategies like averages and means, as well as, data processing strategies like association rules, successive pattern discovery, clustering, and classification are applied to retrieve patterns in the data. Finally, there is the need of analysing and select the useful patterns out of all patterns found. A more detailed overview on taxonomy of Web mining is presented in 2.2.2, and an overview on Web mining processes and its phases is present over the 2.2.3 and 2.2.4 topics.

2.2.2 Web Mining Taxonomy

The taxonomy for Web mining has evolve. Initially, as stated by Cooley et al. [CMS97], Web mining was considered to be of two types: Web content mining and Web usage mining. As of today, Web mining is classified into three different categories, namely: Web content mining, Web structure mining and Web usage mining, as is shown in diagram 2.2.

As mentioned by Patel et al. [PCP11], Web mining is split in categories that symbolizes the different emphasis and different ways of obtaining information, although the differences between them are narrowing since the categories are all interconnected [KS10]. In Web content mining knowledge is automatically retrieve from Web pages content, as analysed in 2.2.2.1. Web structure

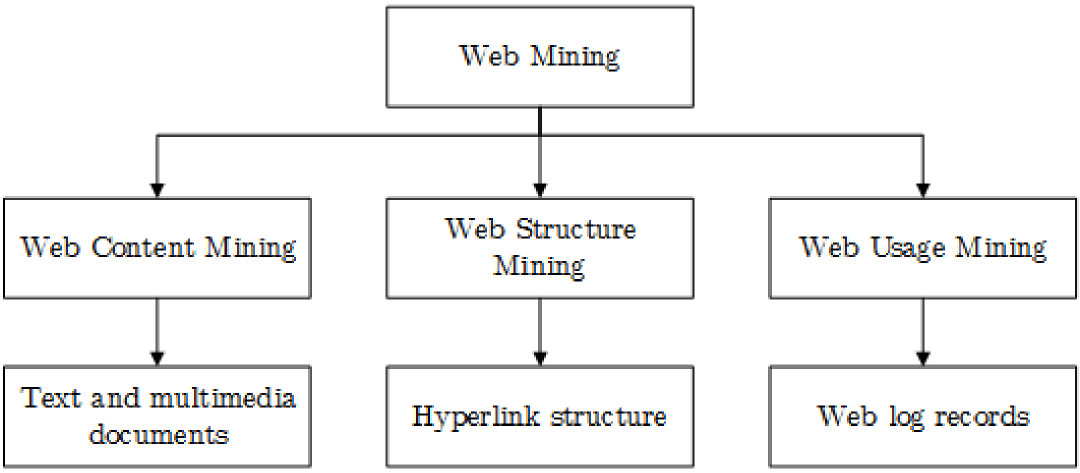


Figure 2.2: Web mining taxonomy [SA13].

mining useful information is extracted from hyperlinks and it show how pages are connected one with another, which is presented in 2.2.2.3. Finally, Web usage mining helps define the behaviour of visitors and classify them into groups, as shown in 2.2.2.2.

2.2.2.1 Web Content Mining

Web content mining (WCM) is concerned with the retrieval of information from a Web site and its Web pages, usually HTML documents, into more structured forms and indexing the information to retrieve it more easily and quickly. This content can be text, image, audio, video, meta-data, hyperlinks and other multimedia resources. As refereed in [JK12], as the WWW grown, the information available on it increased, becoming difficult for users to retrieve useful information from the massive amounts of content, making impracticable the extraction of knowledge from this data sources manually.

2.2.2.2 Web Usage Mining

In Web usage mining (WUM) the main objective is to find user access patterns from Web usage logs. Generally, all visitor's actions are recorded as log files for various proposes including user behaviour analysis, comparison between expected and actual Web site usage, the Web site performance and adjustment of the Web site with respect to the users' interests [PCP11].

As refereed by Patel et al. [PCP11], there are mainly two data sources used for Web usage mining. Web Server Data consists on the data contained in the Web server logs, result of the user interaction with the Web site. This log files may contain useful information characterizing the users' behavior in the Web site. These log files normally contains data like IP addresses, page references, and access time of the user. The other data source is Application Level Data which consists on records of various kinds of events in an application, such as mouse clicks.

2.2.2.3 Web Structure Mining

Web structure mining (WSM) is the technique of analysing the structure of the hyperlinks within the Web itself. This technique takes advantage of analysing a Web site pages a directed labelled graph whose nodes are the documents or pages and the edges are the hyperlinks between them [Für02]. This hyperlinks can be in-links, out-links and co-citation (two pages that are both linked to by the same page).

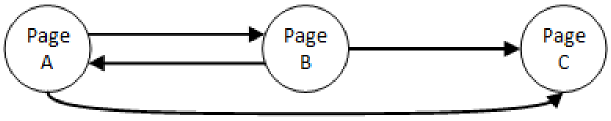


Figure 2.3: Directed graph example with 3 nodes and 3 edges.

This directed graph structure on the Web is called as *Web Graph*. A graph, as defined by David et al. [DJ10], is a way of specifying relationships between a collection of items, where exists a set of items called *nodes* with certain pairs of this objects connected by links called *edges*. A graph G consists of two sets V and E, where the set V is a finite, non-empty set of vertices, that in this case are the Web pages, and the set E is a set of pairs of vertices being these pairs called edges, that here are the hyperlinks between pages. The notation $V(G)$ and $E(G)$ represent the sets of vertices and edges, respectively of graph G. The Web is considered a directed graph which consist of a set of nodes, as defined before, together with a set of directed edges, where each directed edge is a link from one node to another, with the direction being important, like the example present in fig. 2.3 [Rav10].

Although Web structure mining is a relative new research field, link analysis is an old research area [SA13]. The constantly growing interest in Web mining has resulted in a growth of the research in structure analysis and these efforts resulted in a different research area called Link Mining [Get03]. Link Mining results of the intersection of the work in link analysis, hypertext and Web mining, relational learning and inductive logic programming, and graph mining [CG05].

As suggested in [CG05], there are some tasks of link mining which are applicable to the WSM, as follows:

- *Link-based Classification*: focus on the prediction of the category of a Web page, based on words that occur on the page, links between pages, anchor text, HTML tags and other possible attributes found on the Web page;
- *Link-based Cluster Analysis*: consists in finding naturally occurring sub-classes. The data is segmented into groups, where similar objects are grouped together, and dissimilar objects are grouped into different groups. Different than the previous task, link-based cluster analysis is unsupervised and can be used to discover hidden patterns from data;
- *Link Type*: prediction the existence of links, such as the type of link between two entities or the purpose of a link;

- *Link Strength*: technique of associating links with weights;
- *Link Cardinality*: used to predict the number of links between objects.

Some of the most known algorithms in Web structure mining area are the PageRank and HITS algorithm. PageRank [PBMW99], used by Google, calculates the importance of Web pages relying on the link structure of the Web, giving weights to the Web pages according to the number of inbound and outbound link presented on ti. In HITS concept [Kle99] are identified two kinds of pages from the Web hyperlink structure: authorities (pages with good sources of content) and hubs (pages with good sources of links). For a given query, HITS will find authorities and hubs.

2.2.3 Data Collection and Preprocessing

The main goal in data collection stage is to gather resources and retrieve data from Web pages and Web log files.

In one hand, Web pages contains information like text and multimedia that are essential for Web content mining, as is referred in section 2.2.2.1. On the other hand, the hyperlinks presents in this pages are essential for Web structure mining as mentioned in section 2.2.2.3.

Web log files are another information source which can be of mainly two sources: Application Level, which can include data retrieved on the server side, the client side and the proxy side, and Web Server Level. This logs records the users' behaviour very clearly, being essential for Web usage mining, as stated in 2.2.2.2.

2.2.3.1 Web Server Logs

```
192.168.1.133 - - [26/Jan/2016:15:42:49 -0700] "GET /informatica/ HTTP/1.1" 200
    1895 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:43.0) Firefox/43.0"
192.168.1.110 - - [26/Jan/2016:15:42:50 +0000] "GET /escritorio/ HTTP/1.1" 200 843
    "-" "Mozilla/5.0 (Windows NT 6.3; WOW64) Chrome/47.0.2526.111 Safari/537.36"
192.168.1.133 - - [26/Jan/2016:15:43:00 -0700] "GET /informatica/" 200 1895 "-" "
    Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:43.0) Gecko/20100101 Firefox/43.0"
192.168.1.110 - - [26/Jan/2016:15:43:05 +0000] "GET /escritorio/Papelaria/" 200 843
     "-" "Mozilla/5.0 (Windows NT 6.3; WOW64) Chrome/47.0.2526.111 Safari/537.36"
192.168.1.187 - - [26/Jan/2016:15:43:11 -0700] "GET /desporto/Garmin-225 HTTP/1.1"
    200 6936 "-" "Mozilla/5.0 (Windows NT 10.0; WOW64) Chrome/47.0 Safari/537.36"
```

Listing 2.1: Example server log file excerpt.

There are mainly four formats of logs at the server level available to capture the activities of the user on a Web site, namely: Common Log File Format (NCSA), Combined Log Format, Extended Log Format (W3C) and IIS Log Format (Microsoft). All of this formats are in ASCII text format, and are used to act as health monitor for the Web sites and are main source of user access data and user feedback. Each line of an access log is representative of a *hit* on the server. This server *hit* is not the same as an Web page hit, since each file loaded in a Webpage *hit* (multimedia content and other Web resources) corresponds to an entry on the Web server access log. The information present in all formats is pretty similar, and, using as example the Combined Log Format (used by *Apache HTTP Server*) [Fou08]:

LogFormat "%*h* %*l* %*u* %*t* \"%*r*\" %>*s* %*b* \"%{Referer}*i*\" \"%{User-agent}*i*\""

Taking the first line in the listing 2.1 as example, in this format, by order we got:

- 192.168.1.133 (%*h*): IP address of the client (remote host) that requested the content.
- - (%*l*): Client machines *identd* information. This is most of the times empty, showing up as an "-" that indicates missing data.
- - (%*u*): Username information that is only filled when accessing password-protected content.
- $[26/Jan/2016:15:42:49-0700]$ (%*t*): Time stamp with time-zone information corresponding to the visit as it is received by the server.
- "GET / HTTP/1.1" (\"%*r*\"): The HTTP request done. In this case correspond to a "GET" request.
- 200 (%>*s*): HTTP response status code. Different results are given depend on the user privileges and request type (i.e. access protected content gives code "500").
- 1895 (%b): Size of the content transferred in bytes.
- "-" (\"%{Referer}*i*\"): Referrer URL corresponds to the page the visitor was on when they clicked to come to the current page. There are few User Agents (explained below) who send this information to the server.
- "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:43.0) Firefox/43.0" (\"%{User-agent}*i*\"): The User Agent is the representation of whatever software (normally browser) was used to access the content.

Data collected from Server logs generally features incomplete, redundant and ambiguous data. For a more efficient process is essential to filter this noise using preprocessing techniques, resulting more accurate and concise data. Data preprocessing consists on data cleaning, user identification, user session identification, access path supplement and transaction identification [MC10].

According to Mobasher et al. [MCS00] and Li Mei et al. [MC10] data cleaning task is usually site-specific, and involves tasks such as merging logs from multiple servers and parsing of the logs. Also consists on remove Web log redundant and inconsistent data which is not associated with the useful data, reducing the scope of data objects. After the data cleaning there is essential to do user identification in order to identify the users' unique. This can be obtained using cookie technology, user identification techniques and heuristic rule.

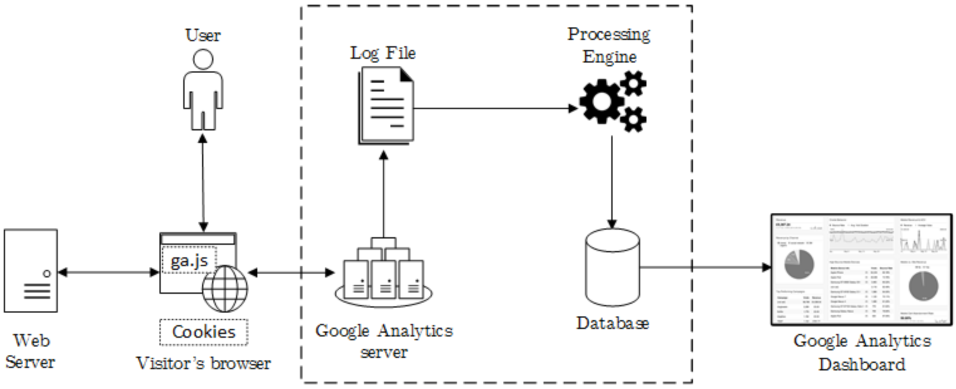


Figure 2.4: Google Analytics functional scheme [Cut10].

User session identification consists on dividing each user's access information into separated sessions. This can be archived using time-out estimation approach, which means that when the time interval between the page requests exceeds a given value that user has started a new session [MCS00]. Also, due to the widespread use of the page caching technology, the users' access path can sometimes be incomplete. To compensate this, path completion technique is used to add the missing requests to the user session information. This technique can resort to the Web site topology in order to complete the missing paths [MC10].

The transaction identification is based on the user's session recognition, and its purpose is to divide or combine transactions according to the demand of data mining tasks in order to make it appropriate for demand of data mining analysis [MC10].

2.2.3.2 Application Layer Tracking

Additionally to the Web server access logs, there exists information retrieved at the application level, sometimes through logs too, but with the objective to record more information about the user interaction with the Web site. This is accomplished generally resorting to tracking domain-specific events throughout the user's visit to the Web site, using built-in tracking scripts on the Web pages (page-tagging) and, sometimes, resorting also to HTTP cookies. One of the most used tools for this kind of data retrieval is Google Analytics[4], but there are others like Piwik[5], Yahoo! Web Analytics[6] and Clicky[7].

Using as example the Google Analytics process, as shown in fig. 2.4, starts by the placing of a JavaScript snippet (*page tag*) in the Web pages that we want to track (*Google Analytics Tracking Code - ga.js*). The data collection starts when the user requests a page from a certain Web server, and the browser process the response data. In this phase the browser may contact also other servers that may contain parts of the requested page, like multimedia resources and scripts (as is

[4] http://analytics.google.com/
[5] http://piwik.org/
[6] http://Web.analytics.yahoo.com/
[7] http://clicky.com/

the case of *ga.js*). Once GATC is loaded it starts identifies attributes of the visitor and her browsing environment, such as how many times that user as visited the Web site, where he came from, his operating system, his Web browser, among other information. After collecting this base data, the GATC sets or updates a number of first-party cookies where it stores information about the user. After this steps the data collected is sent to Google with the information that a visitor has viewed a certain page on the Web site and additional data like events, *pageview* duration among others that can be selected in Google Analytics options like e-commerce data. When Google Analytics server receives the data it stores the data in, for example purposes only, a *logfile*. Each line in this file contains numerous attributes of the information sent to Google, including:

- When the data was collected (date and time);
- Where the visitor came from (i.e. referring Web site or search engine);
- How many times the visitor has been to the site (number of visits);
- Where the visitor is located (geographic location);
- Who the visitor is (IP address).

After this information is stored in the *logfile*, the data collection process is complete. Now the process continues on the Google Analytics processing engine which parses and interpreters the *logfile*. During processing, each line in the *logfile* is split into pieces, one piece for each attribute of the entry in the file. Google Analytics turns each piece of data into a data element called a *field*. Later on, the fields are transformed into dimensions. For example, the IP address becomes the Visitor IP field and the city that the visitor is visiting from becomes the Visitor City field and the City dimension. This transformation is needed since Google Analytics will use fields to manipulate the data and dimensions to build the reports using pattern discovery techniques, as presented in 2.2.4. After this process the data is saved in a database and if an user request a report, the appropriate data is retrieved from the database and sent to the browser.

2.2.3.3 Web Crawlers

For collecting data from pages and structure the most common tool used are Web crawlers. A Web crawler is a program that, given one or more base URLs, downloads the Web pages associated with these URLs, extracts any hyperlinks contained in them that have not been encountered before, and recursively continues to download the Web pages identified by these hyperlinks [Naj09].

In practice, retrieving the content of a single Web page is an easy and fast process. Here is a simple crawler that uses the command-line tool wget:

```
wget -r --html-extension --convert-links --mirror --progress=bar --no-verbose --no-
    parent --tries=5 -level=5 $http://example.com/
```

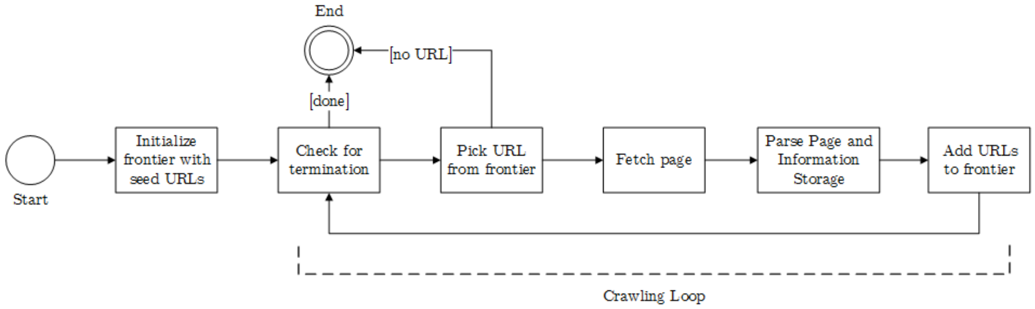


Figure 2.5: Flow of a basic sequential Web crawler [PSM04].

The basic concepts of a sequential Web crawler is shown in fig. 2.5 [PSM04]. The crawler maintains list of unvisited URLs known as the *frontier*. This list is initialized with seed URLs (one or more). Each crawling loop the program chooses the next page to crawl from the *frontier* list, fetch the corresponding content, parse the content in order to extract the URLs and store all the information/content needed, and finally adding the unvisited URLs to the *frontier*. The process can finish when a certain number of pages have been crawled or when the frontier is empty.

As suggested in section 2.2.2.3, the Web can be seen as a large graph with pages at its nodes and hyperlinks as its edges. Because of this, Web crawling can be considered as a graph search problem.

One of the most important components of the crawler is the *frontier*, the to-do list of a crawler, that contains the URLs of unvisited pages. In graph search terminology the *frontier* is an *open list* of unexpanded (unvisited) nodes. The *frontier* can be implemented as a FIFO queue in which case we have a breadth-first crawler that can be used to blindly crawl the Web. The URL to crawl next comes from the head of the queue and the new URLs are added to the tail of the queue. The problem of using this kind of queue is that we need to resort to linear search to find out if a newly extracted URL is already in the frontier, and, in certain cases this can be costly. On alternative approach to the *frontier* is the use of an hash-table [PSM04].

Other alternative is to implement the *frontier* as a priority queue resulting in a preferential crawler, also known as a best-first crawler. The priority queue may be a dynamic array that is always kept sorted by the estimated score of unvisited URLs. At each step, the best URL is picked from the head of the queue. Every time a page is fetched, the URLs are extracted from it and scored based on some heuristic. Sometimes, a crawler may encounter a *spider trap* that leads it to a large number of different URLs that refer to the same page. One way to alleviate this problem is by limiting the number of pages that the crawler accesses from a given page [PSM04].

Next, we need to fetch the HTML page using an HTTP client which sends an HTTP request for a page and reads the response. Once a page is fetched, we need to parse its content to extract information that will be used to fill the frontier. Parsing may imply simple URL extraction or more complex processes of tidying up the HTML content in order to analyse the HTML tag tree. Parsing might also involve steps to convert the extracted URL to a canonical form, remove stopwords from the page's content and stem the remaining words. This process of identify HTML tags and associated attribute-value pairs in a given HTML document is known as URL extraction and canonicalization [PSM04].

The most important component on URL extraction and canonicalization as well as other kinds of information extraction from HTML pages, is that the parser needs to be able to deal with messy markup and resilient to errors. It is important to consider besides the normal than ASCII text, it is also common to run across Unicode URLs and content. Additionally, it is common to find *spider traps* that look like dynamic content but are actually an infinite generation of links.

After having the HTML content is necessary to navigate through the page and the DOM structure to find specific parts of the page that are important or relevant, and then converting that information in a structured form like JSON or other formal data schema in order to storage it in some kind database [Mat12].

2.2.4 Pattern Discovery and Analysis

Pattern discovery consists on retrieving effective, novel, potentially, useful and ultimately understandable information and knowledge using mining algorithms. The methods for pattern discovery include, among other techniques, exploratory data analysis, classification analysis, association rule discovery, sequential pattern discovery and cluster analysis.

The pattern analysis complements the pattern discovery process by focusing on filter and choose the most useful and interesting patterns found during the pattern discovery, in order to selecting the most valuable models for the business [MC10].

2.2.4.1 Exploratory Data Analysis

Exploratory data analysis (EDA) is an approach to analysing data sets with the goal of learn about its main characteristics, general patterns and tendencies. EDA generally resorts to methods like graphical displays and suitable summaries to get a gasp about the most relevant content of the data [GHP]. This allows a deeper overview of the data, retrieve relationships between variables and reveal interesting subsets of the data sets.

In order to apply EDA methods, the data is aggregated in measurable variables such as days, sessions, visitors or country of origin. The insights on the data, like most visited pages by day, average view time of a page, average length of a path through a site, most frequently accessed pages, common entry and exit page, can be extracted with the application of typical statistics methods over the data. Although, this analysis can be superficial and let some important knowledge undiscovered, the information retrieved can be used for guidance on future work over the data, improving the efficiency and, possibly, improving the results of the data mining.

As stated by Gentleman et al. [GHP], there are essential four themes for EDA, namely *Revelation*, *Resistance*, *Residuals* and *Reexpression*. *Revelation* bases itself on the use of suitable graphical display in order to look for patterns present in the data. *Resistance* methods are applied in order to mitigate the problem of extreme observations that can deviate from general patterns,

Table 2.1: Example of a pageview transaction matrix.

Users / Pages	A	B	C	D	E	F
user0	27	62	39	0	0	95
user1	0	6	60	145	84	0
user2	200	0	0	0	0	42
user3	3	29	98	14	60	55
user4	87	0	32	0	81	62
user5	0	2	0	0	62	0
user6	27	58	82	24	74	54
user7	0	0	0	92	0	75
user8	88	20	61	69	45	84
user9	48	40	98	47	0	24

making results more insensible to this observations. When fitting data in simple models such as a line, sometimes the useful knowledge is not fitted in this line, but are the *residuals* that show up as deviations from the line drawn. This *residuals* we often learn about data patterns that are difficult to see by the initial data displays. Additionally, like mentioned before, sometimes there is a need to change the scale of the data or *reexpress*, since the choose measure scale can hide some of the data patterns.

2.2.4.2 Clustering Analysis and Visitor Segmentation

Clustering is a data mining technique that aim to classify user or data items with similar characteristics in groups [MC10]. A cluster is a collection of records that are similar between themselves and dissimilar to records in another clusters. In the Web domain there are mainly two types clusters that can be found: user clusters and page clusters.

After mapping of user transactions into a multi-dimensional space as vectors of pageviews, as shown in table 2.1, standard clustering algorithms like *k*-means can be applied in order to partition this data in groups that have similarities to each other. The *k*-means algorithm decision is based on a measure of distance or similarity among the vectors. After applying any cluster algorithm, the clusters retrieved should be analysed in order to see if they are useful. This process of analysis is need due to the fact that clusters my sometimes have thousands of data points and, because of this, do not provide an aggregated view of common user patterns.

One straightforward approach to create an aggregate view of each cluster is to compute the centroid (or the mean vector) for each cluster. Using the centroid its possible to calculate the distance of each point to its centroid and filter the ones that are most significant in a given cluster, generally the ones that are more close to the centroid point. The resulting set of vectors can be viewed as an aggregate user profile accurately representing the interests or behaviour of a significant group of users [Liu07].

One of the most common and used alternatives to *k*-means algorithm is the DBSCAN (Density-based Spatial Clustering of Applications with Noise) that given a set of points in some space, groups points with many nearby neighbours, marking as outliers points that lie alone in low-density regions (whose nearest neighbours are too far away) [EKSX96]. One of the main advantages compared to the k-means is that DBSCAN does not require one to specify the number of clusters in the data a priori.

When clustering techniques are applied to Web content data, the result may be collections of pages or products related to the same topic or category. On other hand, when cluster algorithms are applied to Web usage data, items that are commonly accessed or purchased together can be automatically organized into groups [MCS00]. A variety of stochastic methods have been proposed for clustering of user transactions, and more generally for user profiling. Recent work on this methods shows that mixture models are capable of capture more complex and dynamic user behaviour, result of the interaction with large and very dynamic Web sites. This data can be to complex to be modelled using basic probability distributions such as a normal distribution. Essentially, each user can have different types of behaviour corresponding to different tasks, and each behaviour can be modelled by a different distribution [Liu07].

Mixture models, such as mixture of Markov models, assume that there exists k types of user behaviour (or k user clusters) in the data, and each user session is assumed to be generated by a generative process that models the probability distribution of the observed variables as well as the hidden variables. Initially, a user cluster is chosen with some probability. Then, the user session is generated from a Markov model with parameters specific to that user cluster. A Markov model is a stochastic model used on modelling randomly changing systems, where it is assumed that future states depend only on the actual state and not on the sequence of events that preceded it. After this, it is used the Expectation–Maximization[8] algorithm, to learn the proportion of users assigned to each cluster as well as the parameters of each Markov model [CHM+00]. The resultant user models are very flexible. For example, a mixture of first-order Markov models is capable of probabilistically cluster user sessions based on similarities in navigation behaviour and, also, characterize each type of user behaviour, thus capturing popular navigation paths or characteristics of each user cluster. Besides that, mixture models have limitations too. Each individual observation (i.e. user session) is generated from only one component model, the probability assignment to each component only measures the uncertainty about this assignment, thus limiting the model ability of capturing complex user behaviour, and, besides that, may result in model overfitting. Overfitting occurs, generally, when a model is excessively complex, such as being to detailed (i.e. too many parameters) relatively to the number of observations.

2.2.4.3 Association and Correlation Analysis

Association rule discovery and statistical correlation analysis is useful for finding groups of items that are purchased together or set of pages that are commonly accessed. This enables Web sites

[8] The expectation–maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables [B+98].

Table 2.2: Example of a market basket transactions.

TID	Items
1	$\{Milk, Bread\}$
2	$\{Bread, Diapers, Beer, Eggs\}$
3	$\{Milk, Diapers, Beer, Cola\}$
4	$\{Bread, Milk, Diapers, Beer\}$
5	$\{Bread, Milk, Diapers, Cola\}$

to provide effective cross-sale product recommendations or, even, better organize the Web site structure and content towards the reflection of typical user actions.

Statistical correlation is a statistical technique which tells us if two variables are related. Each two random variables or two datasets related between themselves are considered statistical dependent. Statistical correlation analysis consists on the analysis of any broad class of statistical relationships involving dependence. Formally, *dependence* refers to any situation in which two random variables are not probabilistic independent. There are several correlation coefficients, often denoted ρ or r, measuring the degree of correlation. The most common of these is the Pearson correlation coefficient, which is sensitive only to a linear relationship between two variables [Exp09].

Association analysis is an approach for discovering interesting relationships hidden in large data sets. The relations retrieved can be represented in association rule or frequent item sets. For instance, by the transactions sample shown in table 2.2 we can retrieve the rule $\{Beer, Bread\} \Rightarrow \{Milk\}$, that states that there is a strong relation between customers who buys bread, milk and beer, since many customers who buy beer and bread also buy milk [TSK05].

As stated by Mei et al. [MC10], in Web domain, the association rules discovery is principally important on Web on extracting the relevant rules from the access information logs and finding the inter-relationship by analysing the potential linkages between users access to Web pages.

Formally, let $I = \{i_1, i_2, ..., i_d\}$ be the set of all items in the market basket data, and $T = \{t_1, t_2, ..., t_N\}$ be the set of all transactions. Each transaction t is represented as a binary vector, with $t[k] = 1$ if t bought the item, $I[k] = 0$ otherwise. Let X be a set of some items in I. We say that transaction t satisfies X if for all items I_k in X, $t[k] = 1$.

An association rule is express in the form $X \rightarrow Y[sup, conf]$, where X and Y are set of items, *sup* is the support of the itemset $X \cup Y$ representing the probability that X and Y occur together in a transaction, and *conf* is the confidence of the rule, defined by $sup(X \cup Y)/sup(X)$, representing the conditional probability that Y occurs in a transaction given that X has occurred in that transaction. One other parameter of interest is the *lift* that measures the performance of an association rule at predicting or classifying cases as having an enhanced response, measured against a random choice targeting model. *Lift* is a value that indicate us information about the increase in probability of the consequent (Y) given the antecedent (X).

One of the most common approaches for association discovery is the *a priori* algorithm, which consists on identifying the frequent individual items in a dataset and trying to extend them to larger itemsets as long as the itemsets appear sufficiently often. The frequent itemsets determined by the algorithm are then used to determine association rules, based on their confidence and support levels [Liu07].

2.2.4.4 Classification and Prediction

Classification analysis process consists on mapping a data item into one of predefined categories. In the Web domain this consists mainly on attributing an user profile into one of the established categories of users [MC10]. To accomplish this is necessary to extract and select the features that best describe the proprieties for each class or category. Classification can be archived by using a set of supervised learning algorithms such as decision trees, Naive Bayesian, k-nearest neighbour and Supported Vector Machines. Additionally is possible to use previous known clusters and association rules for classification of new users [Liu07]. Normally it is used the previously discovered clusters and association rules, as shown previously in section 2.2.4.2 and 2.2.4.3, as base classes for the classification algorithms. For example, a classification model can be built to classify users according to their tendency to buy or not, taking into account features such as users' demographic characteristics, as well their typical navigational patterns.

One of the most important application of classification and prediction techniques in the Web domain is in collaborative filtering technique which is an essential component of many recommendation systems, as presented in 2.1.3. Most recommender systems that use collaborative filtering are based on k-Nearest-Neighbour classification algorithm to predict user ratings or purchase intentions by measuring the correlations between a current (target) user's profile (which may be a set of item ratings or a set of items visited or purchased) and past user profiles, in order to find users in the dataset with similar interests and characteristics [HKTR04].

The k-Nearest-Neighbour (kNN) classification algorithm bases itself on comparisons between the recorded activity for a given user and the historical records T of other users, searching for the top k users who have similar interests. The kNN algorithm measures the similarity between the given user active session u and each past transaction vector v (where $v \in T$). The top k most similar transactions to u are considered to be the neighbourhood for the session u [Pet09]. Once proximities are calculated, the most similar users are selected and this information is used to recommend items that were not already accessed or purchased by the active user u.

Decision tree induction technique consists on the generation of a decision tree and performing classification on the given data using it. A decision tree is a tree in which each non-leaf node denotes a test on an attribute of cases, each branch corresponds to an outcome of the test, and each leaf node denotes a class prediction [CKK02], and is the result of a process of categorisation and generalisation of a given set of data. A typical data record comes in the form $(x,Y) = (x_1, x_2, x_3, ..., x_k, Y)$. The dependent variable, Y, is the target variable that we are trying to classify. The vector x is composed of the input variables, x_1 until x_n, that are used for that classification. Sometimes, the tree learning process can create over-complex trees that do not generalise well from the training data, resulting in the overfitting of the tree. To avoid this problem its

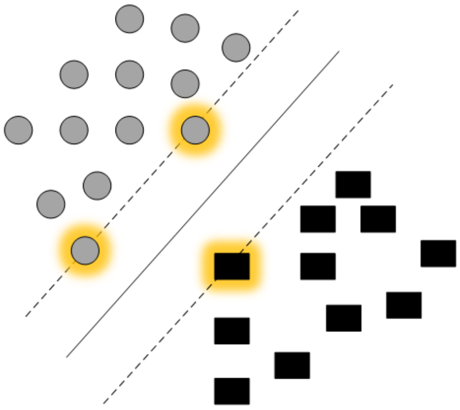


Figure 2.6: Example of an SVM classification with the best plane which maximizes the margin.

common to use pruning techniques, capable of reducing the size of the tree by removing parts of the tree that provide little power to classify instances.

Another technique for classification is Naive Bayesian which bases itself on Bayes' theorem. Naive Bayes classification can predict a class membership probabilities, such as the probability that a given record belongs to a particular class. Let $X = x_1, x_2, ..., x_n$ be a sample, whose components represent values made on a set of n attributes. In Bayesian terms, X is considered an *evidence*. Let H be some hypothesis, such as that the data X belongs to a specific class C. For classification proposes, the objective is to find probability that sample X belongs to class C, given that we know the attribute description of X [Leu07].

Supported Vector Machines (SVM) is also another supervised learning algorithm, useful for recognizing subtle patterns in complex datasets. This algorithm performs discriminative classification, learning by example, to predict the classifications of previously unseen data. The approach, as described by Bennett et al. [BC00], is systematic, reproducible, and properly motivated by statistical learning theory. Training involves optimization of a convex cost function in such way that there are no false local minima to complicate the learning process. SVM bases itself over three fundamental principals: *margins*, *duality* and *kernels*. This technique can be used for simple linear classifications and easily extend for more complex tasks. SVM method tries to maximize the distance or *margin* between the support planes for each class, in order to find the plane furthest from both sets, known as hyperplane, as shown in fig. 2.6. To accomplish this the support planes are pushed apart until they bump into a small number of data points from each class, known as support vectors, highlighted in fig. 2.6. *Duality* is the mathematical programming concept which states that the supported vectors found when maximizing the margin between parallel supporting planes is are the same ones found when using the bisecting method to find the closest points in the convex hull approach. *Kernels* method are part of SVM as they use of kernel functions, which enable SVM to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between all the pairs of data in the feature space. This operation is often computationally cheaper and it is

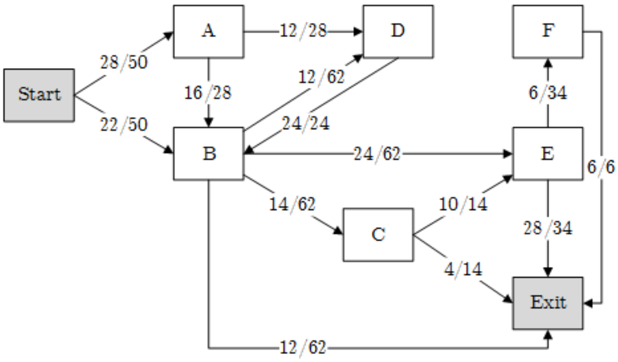


Figure 2.7: Example of an user navigational trails.

Figure 2.8: Frequency of occurrences of each transaction.

Transaction	Frequency
A, B, E	10
B, D, B, C	4
B, C, E	10
A, B, E, F	6
A, D, B	12
B, D, B, E	8

known as *kernel trick*. SVM principal advantages are the modularity and being almost immune to the curse of dimensionality and overfitting.

2.2.4.5 Sequential and Navigational Patterns Analysis

The sequential and navigational patterns analysis attempts to find inter-session patterns such that the presence of a set of items is followed by another item in a time-ordered set of sessions, giving us the causation relations between data. Other examples of temporal analysis that can be made on these patterns include trend analysis, change point detection, or similarity analysis. In the Web context, these techniques are employed to capture the Web page trails that are often visited by users, in the order that they were visited.

Sequential patterns are sequences of items that frequently occur in a sufficiently large proportion of sequential transactions. Formally, a sequence $\langle s_1 s_2 \ldots s_n \rangle$ occurs in a transaction $t = \langle p_1, p_2, ..., p_m \rangle$ (where $n \leq m$) if there exist n positive integers $1 < a_1 < a_2 < ... < a_n \leq m$, and $s_i = p_{ai}$ for all i. We say that $\langle c_{s1}, c_{s2} \ldots c_{sn} \rangle$ is a contiguous sequence in t if there is an integer $0 \leq b \leq m - n$, and $cs_i = p_{b+i}$ for all $i = 1$ to n. Contiguous sequential patterns (CSP) are patterns where each pair of adjacent items, s_i and s_{i+1}, must appear consecutively in a transaction t which supports the pattern. The CSP patterns are used to capture frequent navigational paths among user trails. General sequential pattern are used to represent more common navigational patterns within the site [Mob06].

One approach to modelling this type of user flow trough the Web site is using a Markov model in which each page (or a category) is represented as a state and the transition probability between two states represents the likelihood that an user will navigate from one state to the other [Mob06]. For example we can calculate the probability of a given user will make an order, given that she has visit the delivery conditions page.

Formally, a Markov model is characterized by a set of states $\{s_1, s_2, ..., s_n\}$ and a transition probability matrix, $[Pr_{i,j}]_{n*n}$, where $Pr_{i,j}$ represents the probability of a transition from state s_i to state s_j. This make Markov models very adapted for predictive modelling based on time-series events. Each state represents a contiguous subsequence of prior events. The order of the Markov model corresponds to the number of prior events used in predicting a future event. So, a *kth*-order

Markov model predicts the probability of next event by looking the past k events. Higher-order Markov models generally provide a higher prediction accuracy since they use an larger number of prior events. As example of a set of transactions that can be model using Markov chain consider the transactions presented in fig. 2.7, consisting on the pages A, B, C, D, E and F. For each transaction the frequency of occurrences of that transaction is given in the table 2.8. The (absorbing) Markov model for this data is also given in table 2.8. The transitions from the *start* state represent the prior probabilities for transactions starting with pageviews A and B. The transitions into the *final* state represent the probabilities that the paths end with the specified originating pageviews. For example, the transition probability from the state B to E is $24/62 = 0.387$ since out of the 62 occurrences of B in transactions, E occurs immediately after B in 24 cases.

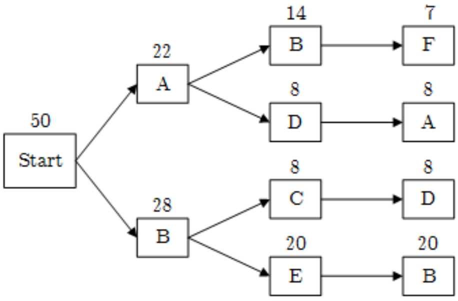


Figure 2.9: An example of modelling navigational trails in an aggregate tree.

Another way of representing contiguous navigational paths is by inserting each path into a tree structure. This method, as presented by Spiliopoulou et al. [SF98], is part of the Web Utilization Miner system, extracts the visitor trails from Web logs and aggregates them by merging trails with the same prefix into a tree structure called *aggregate tree*. This aggregate tree is a *trie*, where each node corresponds to the occurrence of a page in a trail. Common trail prefixes are identified, and their respective nodes are merged into a *trie* node. This node is annotated with a support value which consists on the number of visitors having reached the node across the same trail prefix. The main advantage of this method is that the search for navigational patterns can be performed very efficiently and the confidence and support for the navigational patterns can be readily obtained from the node annotations in the tree. For example, from the *trie* present in fig. 2.9, considering the navigational path $< A, B, F >$, the support for this sequence can be computed as the support of the last page in the sequence, F, divided by the support of the root node: $7/50 = 0.14$, and the confidence of the sequence is the support of F divided by the support of its predecessor, B, or $7/14 = 0.5$. If there are multiple branches in the tree containing the same navigational sequence, then the support for the sequence is the sum of the supports for all occurrences of the sequence in the tree and the confidence is updated accordingly.

2.3 User Profiling

2.3.1 Introduction

User profiling are an essential part of any personalized recommendation system [Bou13]. The quality of the user profile affects the quality of recommendations directly. Only a system that understands user's requirements and interests is capable of recommended satisfactory information to the user. In order to describe interests exactly, relevant information about the user characteristic and interests should be collected and record and manage it through model building.

There are mainly three key areas in user profiling as mentioned by Bouneffouf [Bou13], namely: the background of the user (acquired knowledge in different subjects), the user objectives and his interests. The background concerns all the information related to the user past experiences, including how the user is familiar with the working environment of the Web site. Objectives consists on the users needs, for example what he search for. Finally, the user interests consists on the pages that the user has visited or other interactions that the user have on the Web page (time spend on a page, scroll and click events or even printing/saving a page).

User profiling is typically of two kinds: behaviour-based or knowledge-based [HF08]. Behaviour-based approaches resorts to monitor techniques in order to get a gasp about the user behaviour activities generally in an unobtrusive way, commonly using machine-learning techniques to discover patterns in their behaviour. On other side, knowledge-based approaches design static models for users and match users to the closest model dynamically. Questionnaires and interviews are often used to obtain this user knowledge.

2.3.2 User Profile Representation

Modelling the user's profile consists of designing a structure for storing all the information which characterizes the user and describes his interests, his background and his objectives. User profiles are generally represented as sets of weighted keywords, semantic networks, or weighted concepts, or association rules.

2.3.2.1 Keyword-based Profiles

Table 2.3: An example of a keyword-based user profile.

Technology	*Weight*	0.60	0.72	0.45	0.33	...
	Keyword	Laptop	Smartphone	Keyboard	Screen	...
Sports	*Weight*	0.33	0.80	0.75	0.61	...
	Keyword	Football	Running	Ball	Rugby	...
Music	*Weight*	0.37	0.45	0.55	0.23	...
	Keyword	Rock	Flute	Orchestra	Symphony	...

Keywords is the most common approach on representation of user profiles, since they can be automatically extracted from Web pages and/or provided directly by the user. Keywords have generally associated weights that are numerical representations of the keyword importance in the user profile. Each keyword can represent a topic of interest or keywords can be grouped in categories to reflect a more standard representation of user's interests. An example of a weight keyword-based user profile is shown in table 2.3.

One approach to give weights to keywords is the use of tf*idf weighting scheme. In this schema each profile is represented in the form of a keyword vector, and the retrieved Web pages by the system in response to a search are converted to similar weighted keyword vector. Created vectors are then compared to the profile using the cosine formula, and only the corresponding pages for those vectors that are closest to the profile are then passed on to the user [TPSA07].

Besides the simplicity of implementation of keyword-based profiles, the use of several vectors to represent the profile permits to take into account the different interests and their evolution through time. On other side the default version of this representation is in the lacks of structure and semantic (no connexion between terms). Additionally, one of the main drawbacks in keyword-based profiles is that many words have multiple meanings (polysemy) which can conduct to inaccurate profiles since the keywords in the profile are ambiguous [GSCM07].

2.3.2.2 Ontologies Representation

An ontology, as specified by Middleton et al. [MSD04], is a conceptualisation of a domain into a human-understandable, but machine-readable format, representation of entities, attributes, relationships, and axioms. Ontologies can, for example, be a rich conceptualisation of the working domain of an e-commerce Web site, representing the main concepts and relationships of the customers activities. These relationships could represent isolated information such as an customer last purchased item, or they could represent an activity such as the set of visited pages on a session. Ontologies are in this way used to refer to the classification structure and instances within a knowledge base.

Ontology-based user-profiling approaches have been used to take advantage of the knowledge contained in ontologies instead of attempting user-profile acquisition. This representation, as stated on [GA05], allows to overcome the limitations of the connexion representation by presenting the user's profile in the form of a concepts hierarchy. Each class in the hierarchy represents the knowledge of an area interesting to the user. The relationship (generalization / specification) between the elements of the hierarchy reflects a more realistic interest of the user. But this approach have some problems related to the heterogeneity and diversity of the user's interests (i.e. users may have different perceptions of the same concept, which leads to inaccurate representations).

2.3.2.3 Semantic Network Profiles

Semantic network profiles, as presented by Bouneffouf [Bou13], appears an representation solution capable of address the polysemy problem present in some representations. In this approach

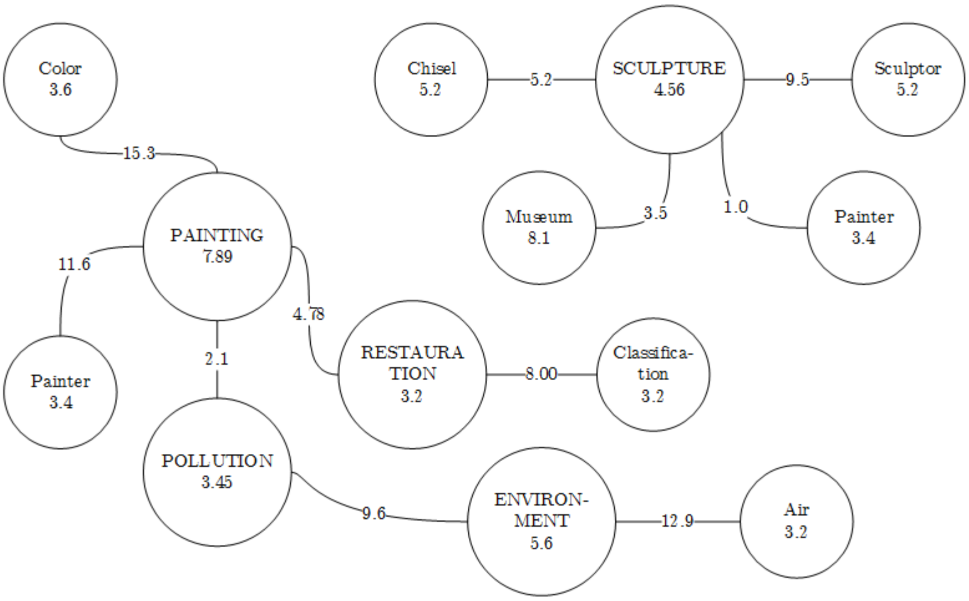


Figure 2.10: An example of an user profile based on semantic networks.

the user profiles are represented by a weighted semantic network in which each node represents a concept. One of the approaches to build this representation is the Hybrid User Modeling System, proposed by Micarelli et al. [MS04]. In this system each user profile consists on three components, a header that includes the user's personal data, a set of stereotypes and a list of interests. The stereotype consists on a prototypical representation of the user, containing a set of interests represented by a frame of slots. Each one of this slots comprises three concepts: *domain*, *topic* and *weight*. The *domain* identifies an user's area of interest, the *topic* is the specific term that the user used to identify the interest, and a *weight* that represents the user's degree of interest on the topic. The user model is, in this way, a structure embodying the *semantic links* and *justification links*, as well as *domain*, *topic*, and *weight*.

The semantic links include lists of keywords co-occurring in a page associated with the slot and the respective degree of affinity with the topic. The profile is given as a set of semantic networks, and each slot is a *planet* - a single and representative weighted term for a given concept - and *semantic links* as *satellites* - subsidiary nodes linked to the *planets* that represent additional weighted keywords associated with the concept - as represented in the example 2.10.

2.3.2.4 Concept-based Profiles

Concept-based representation for user profiles is similar to semantic network-based profiles in the way that both use conceptual nodes and establishes relations between them as way to represent profiles, with the difference that, instead of the nodes represent specific or sets of related words, in concept-based profiles the concepts are abstract topics interesting to the user. Besides this similarity, this representation also uses ideas from keyword-based profiles in the way that its used

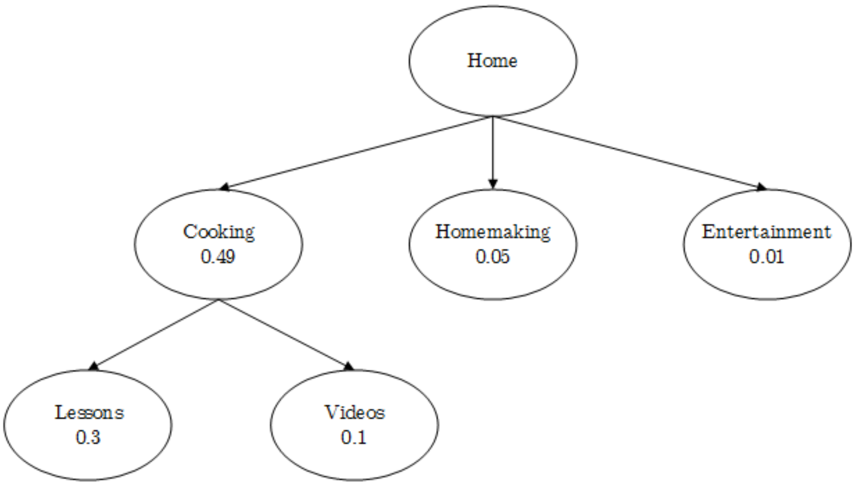


Figure 2.11: An excerpt of a concept-based user profile [GSCM07].

a vector of weight features, but instead of features being used to represent keywords are used to represent concepts [GSCM07].

An approach to represent this kind of profiles is to use a hierarchical view of concepts, as referred by Gauch et al. [GSCM07], since it enables the system to make generalizations, easily adapt levels of the scheme and dynamically change. The most simple concept hierarchy based profiles are built from reference taxonomy or thesaurus. More complex ones are created using reference ontologies, where relationships between concepts are explicit specified, resulting in a profile more richer in information with a wide variety of different relationships. An example on this approach is given on fig. 2.11.

2.3.3 User Profile Construction

User profiles are constructed from information sources using a variety of techniques based on machine learning or information retrieval, trying to build a profile which is the closest reflection of the user interests and characteristics. Depending on the user profile representation picked, different approaches are needed in order to build them. Sometimes this become an even complicated task, since users may not be sure on his own interests, and, on the other side, often the user does not want or even can't do efforts in order to create its profile [Bou13].

The most simplistic approach for building the user profile is to explicit ask the user for keywords representative of his interests. This approach depends totally on the user because if he is not familiar with the system or the vocabulary used, it becomes a difficult task providing the proper keywords representative of his interests. One improvement to this technique is, as the same time that the user is navigating trough pages, suggesting keywords that may match the user interest, being incumbent upon the user the choose on whatever keywords are more interesting to them. Additionally, we got the machine learning approach which is, in the majority of the cases the most appropriated, since that is not reasonable to ask an user a set of keywords describing his preferences, but instead observing the user's behaviour through his interactions with the Web site in order to learn his profile [Bou13].

Depending on the representation used to represent the user profile, there are different necessities for constructing them. In keyword-based profiles, the user profiles are initially created by extracting keywords representative of the Web pages that the user visited and then its taken in consideration the most important keywords in each page and its attributed a corresponding weight. The simplest approach to keyword-based profiles is the construction of a single keyword profile for each user, but generally is better to use multiple keyword profiles for each user, one per interest area, resulting a more accurate picture of the user. Given the example of an user with interest in Sports and Cooking. If a single keyword vector is used it will point to the middle of this two different topics, given an user which is interested in athletes who cook or people who cook for Football games, but if instead is used a pair of vectors as representation, this two interest will be independent, representing the user more accurately [GSCM07].

Semantic network-based profiles are typically built by collecting explicit positive or negative feedback from users. As in the case of keyword-based profiles, the profile is built using keywords that are extracted from the user-rated pages. The principal difference is that every keyword is added to a network of nodes, in which each node can represent an individual word or, in some approaches, a particular concept and its associated words [GSCM07].

In ontologies representation the user profiles are created by automatically and implicitly analysing the user navigation habits. A profile is essentially the reference ontology whose concepts have weights indicating the perceived user interest in each of them. The pages the user visits are automatically classified into the concepts contained in the reference ontology and the results of the classification are accumulated. By this process, the concepts in the reference ontology receive weights based on the amount of related information the user has browsed [GA05].

Concept-based profiles differ from semantic network profiles in the way that they describe the profiles in terms of pre-existing concepts, instead of modelling the concepts as part of the user profile itself. Although, this profiles still depend on some way to determine which concepts fits in a certain user based on their feedback. This information can be collected from user feedback on pre-classified pages or instead by collecting feedback on a wide variety of pages and then apply text classification techniques to find the concepts present in each page [GSCM07].

2.4 Conclusions

From the literature review, several conclusions could be made. E-commerce is a relative new field (as the Internet itself), but is increasing its impact, with ascendant number of transactions and values every year. E-commerce site owners in order to increase profits and number of sales resort to Web personalization for each user, using target marketing and recommendation systems. Here, data mining appears as a crucial component, where techniques has been applied to the Web in order to understand the structure and content of the Web, as well the user behavior, resulting in a sub-area inside the data mining called Web mining which itself splits in three different, but sill connected areas, namely: Web content mining, Web structure mining and Web usage mining.

Different data mining techniques and processes have been applied and modified in order to meet the necessities of web mining, and new tools for data collecting were born, like crawlers. Problems associated with Web mining also show up, and solutions have been discovered and applied, for example the path completion technique applied to incomplete navigational paths. Also, well known process of pattern discovery has been applied like association rule mining and new were born like navigational pattern analysis.

In order to map archetypical Web site users, through profiles, various representation appeared, each one more adapted for specific cases and with different degrees of complexity. Although, when we want to collect and represent all the website data, not only the users but also the pages content and structure, establishing relationships between all this data, there is a lack of design pattern or methodology in order to accomplish this, and this were the methodology described in chapter 3 fits, as a possible approach to mitigate this lack of literature.

Chapter 3

Methodology

3.1 Introduction

This section details the methodology used in this dissertation. In order to retrieve the content of Web pages, relationships between them and knowledge about its archetypical users different approaches for each information source should be used in order to successful retrieve the useful information and mitigate the problems associated with each of this different sources. A proper methodology was designed in order to conciliate this different approaches and techniques in an all-in-one process capable of give as output a representation of a given e-commerce Web site content, structure and usage.

It is also important to consider that the information should be stored in a consistent data model for the data stored be easily retrieved, updated and incremented.

This methodology intends to establish connections between all the information sources trying to complement what is, sometimes, incomplete data, and highlighting relationships between information sources that otherwise could pass unnoticed. This methodology meant to be applied not only over one certain type or specific Web site but be able to operate over any e-commerce Web site, dealing with the different structures, content or formats that it may have.

3.2 Overview of the Proposed Methodology

The methodology will follow the typical flow of a Web mining process, as it showed in the fig. 2.1. As a more detailed overview we can consider the following steps:

1. Choosing and selecting which data sources that are used, in this case will consist on the Web site itself (pages and connection between them) and the Web server records associated with the user interactions with the pages.

2. The proper techniques are used to retrieve the information present on this sources, namely, for the Web site will be used a crawler to navigate between pages an retrieve the content present in them and for the Web server records will be retrieved the server log files.

3. This raw data is preprocessed and cleaned in order to mitigate common data problems and make it usable.

4. In order to retrieve the typical browser habits of the costumers data pattern discovery techniques are applied.

5. User profiles are developed using the data that comes from the patter discovery and analysis process.

6. All the information including user profiles, pages and connections is stored in a database for being easily retrieved, updated and incremented.

3.3 Content Retrieval and Preprocessing

The content retrieval and preprocessing phase of the methodology will consist on aggregate and collect the data from the chosen data sources and make the data usable by cleaning and adjusting the raw data. Dividing this process in two main parts depending on the data source, processes and techniques related to Web page content and structure on one hand, and Web usage on the other.

Considering the Web pages content and structure the typical approach to collected information is using a crawler that downloads each page of a Web site and search for more pages to crawl. This concept was more detailed in section 2.2.3.3. For obtaining the hyperlinks present in a page regex expressions are used. There exists a list of Web pages to visit and one with the already visited, and for each new hyperlink found there is checked that the page is not present already in any list in order to not visit the same page twice. Each record of a Web page will have associated with relevant data like a list of all the pages that it points to, and possible data extracted from pre-defined places like product category, price and even review score, but, the additional data that need to be collected should be specified as a parameter.

At the Web usage component all the data comes from the Web server logs. This data needs to be separated in all its dimensions (depending on the format of the logs) and users and sessions identified. The are lots of common problem with this logs, and, as presented in detail in section 2.2.3.1, common used solutions. From the point of the preprocessing phase the problems and approached solutions are the following:

- Every hit on the server is recorded, even content that is not a page, like JavaScript or CSS files, are present, and they does not give any useful information about an user. This lines should be ignored in order to clean the data.

- One large part of the server logs is created by the interaction of web crawlers, bots and indexers such as Google Bot[1] and Bing Bot[2] that are associated with the search engines of Google and Bing respectively. This effect is more noticeable on small e-commerce website,

[1] http://www.google.com/bot.html

[2] http://www.bing.com/bingbot.htm

with less visits and the majority of the log entries is noise. For mitigating this problem a list of know bots is present and the entries ignored, and also if its found an user visiting an abnormal quantity of the Web site pages should be classified as a bot too and also ignored.

- Due to the incremental use of cache techniques by the users browsers, sometimes the hits present in the server log does not represent all the interactions that the user had with the website. For mitigating this problem, path completion techniques considering the web site map as reference.

Due to the data in this logs be so disperse, depending on the pattern discovery which we want to apply, some different preprocessing can be needed. This is the case of clustering where there is the need of mapping the user transactions into a multi-dimensional space as vector of pageviews, in order to be possible applying the typical clustering techniques.

In this stage the most essential component is to make the raw data usable, cleaning it from inconsistent and irrelevant data, and mapping the information into the correct dimensions in order to make possible the application of classification and other pattern discovery techniques.

3.4 Pattern Discovery

After the collecting and preprocessing the data from an e-commerce website, follows the process of discovering patterns in this data. This process is essential to get to know the typical user behaviour on the Web site. Techniques for pattern discovery were carefully analysed in section 2.2.4. This techniques are essential in order to understand the user relative data, finding the most usual sequential user flows through the Web site pages, and, also, cluster of users and typical associations between content visited.

Firstly, we proceed to an Exploratory Data Analysis, checking general numbers like the total number of Web site visit and the views for each page (the most or less visited pages), information from the Web usage data, and, on other hand, from analysing the content and structure, we can get information like the total number of pages of the website and the pages with more or less inbound/outbound hyperlinks.

Clustering techniques are applied over the pageview transaction matrix (Web usage data), in order to get a gasp of typical users that visit the Web site in reflection of what pages they visit. The typical approach used is the DBSCAN algorithm due to its simplicity and capacity of ignoring outliers. This process of clustering can fail in relative small e-commerce Web sites, where almost each user can correspond to a different cluster.

For finding the most common contiguous sequential pattern (CSP) a Markov model approach is used in where each page is represented as a state and a transaction probability is given for each transaction, representing the likelihood of an user navigate from a page to another one. Additionally to this process, association rules can be found, reflecting relations between pages. This association rules can be found in the usage data of the Web site, and correlated using information about the Web site structure.

In this stage, resulting from the pattern discovery techniques applied, we get a really deep gasp about the Web site and the users behaviour. An analysis of the pattern found is made to filter inconsistent or irrelevant patterns, and discard this ones.

3.5 Data Representation and Storage

The last stage of the methodology consists on the storage of the Web pages content and possible extracted content, connections between pages and typical users profiles. In order to map the Web page content and structure, a copy of the HTML page will be saved and a unique identifier is attributed, with additional variables for each extracted filed, and references to all of the outbound pages.

For mapping the user interests and common navigational patterns, an ontology approach is used to correctly represent the user interest and respective degree of interest. The references to the pages from the navigational paths are preserved and maintained, resulting in a consistent model, easily finding the references to each object.

After this stage the data is easily retrieved, modified and update, to be able to reflect any change on the website or in the users behaviour.

3.6 Validation

For validating the present methodology a *proof of concept* tool will be design implementing this methodology.

The results obtain will be analysed in two different stages. On the first one the resulting data from the Web page content and structure will be matched to the live Web site. The second stage consists on validating the users profile representations, matching the information with the users information from another tool, like, for example, Google Analytics reports.

For a more valid approach a minimum of two Web sites with distinct typical users and products should be used to validate the methodology, in order to mitigate possible bias and other errors typical with data mining projects.

3.7 Conclusion

The methodology here presented is an approach to Web mining over e-commerce related Web sites. This methodology takes as a input all the e-commerce Web site pages and server records, and applying data mining (Web mining) techniques in order to model this information in a consistent and easily retrieved data structure.

This methodology will be validated and tested using real scenarios and a proper *proof of concept* that implements the given methodology.

Chapter 4

Conclusion and Further Work

In this chapter is made a review of the work made until now, namely, the literature review and the proposed methodology. Besides this is also showed a prevision of the further work to be done.

4.1 Final Remarks

During this stage of the project was done a state of the art research crossing all the influencing topics in this dissertation, presented in section 2. Initially was presented the e-commerce context and the importance of e-metrics and recommendation systems. After this was learned about data mining techniques and its applications to web, called Web mining, the taxonomy of it, typical data collection approaches, data preprocessing and cleaning methods and pattern discovery/analysis techniques. This state of art terminates with the presentation of typical user representation approaches and its differences.

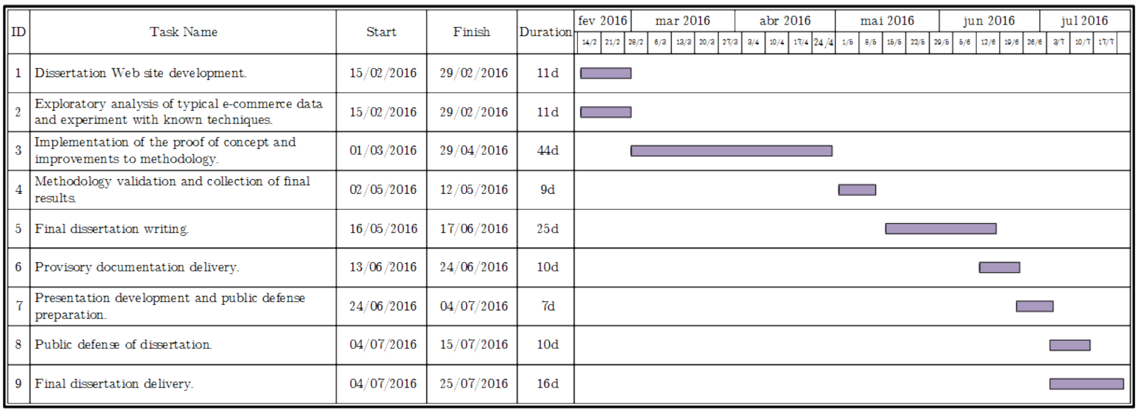
In the section 3 is presented the methodology to accomplish the introduced objectives and goals, and to be implemented as an *all-in-one* approach in Web mining of e-commerce Web sites.

4.2 Further Work

The future work will consist on the development of a *proof of concept* tool capable of demonstrate the applicability of the proposed methodology, and realize possible modifications and improvements to this methodology to make it more complete and consistent.

This *proof of concept* will be done as an iterative process of validation and continuous improvement, in order to mitigate possible problems and impediments that we can come up with. An iteration will act as a milestone, and by the end of each one is expected to have some functional prototype, a list of problems found and a possible list of resolution approaches to the next iteration.

The final work will consist on writing up the dissertation, public presentation and publication.



ID	Task Name	Start	Finish	Duration
1	Dissertation Web site development.	15/02/2016	29/02/2016	11d
2	Exploratory analysis of typical e-commerce data and experiment with known techniques.	15/02/2016	29/02/2016	11d
3	Implementation of the proof of concept and improvements to methodology.	01/03/2016	29/04/2016	44d
4	Methodology validation and collection of final results.	02/05/2016	12/05/2016	9d
5	Final dissertation writing.	16/05/2016	17/06/2016	25d
6	Provisory documentation delivery.	13/06/2016	24/06/2016	10d
7	Presentation development and public defense preparation.	24/06/2016	04/07/2016	7d
8	Public defense of dissertation.	04/07/2016	15/07/2016	10d
9	Final dissertation delivery.	04/07/2016	25/07/2016	16d

Figure 4.1: Gantt chart of the project.

4.2.1 Gantt Chart

The preview work time dedicated to each component and also the total duration of the dissertation work is presented in figure 4.1. The project is scheduled to start on February 15, 2016, spanning months and finishing in July 1.

References

[B$^+$98] Jeff A Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[BC00] Kristin P Bennett and Colin Campbell. Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter*, 2(2):1–13, 2000.

[Bou13] Djallel Bouneffouf. Towards User Profile Modelling in Recommender System. pages 1–5, 2013.

[CG05] Jr. Da Costa, M.G. and Zhiguo Gong Zhiguo Gong. Web structure mining: an introduction. *2005 IEEE International Conference on Information Acquisition*, pages 590–595, 2005.

[CHM$^+$00] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Visualization of Navigation Patterns on a Web Site Using Model Based Clustering. pages 1–29, 2000.

[CKK02] Yoon Ho Cho, Jae Kyeong Kim, and Soung Hie Kim. A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, 23(3):329–342, 2002.

[CMG$^+$99] Mark Claypool, Tim Miranda, Anuja Gokhale, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. *Proceedings of Recommender Systems Workshop at ACM SIGIR*, pages 40–48, 1999.

[CMS97] R Cooley, B Mobasher, and J Srivastava. Web mining: information and pattern discovery on the World Wide Web. In *IEEE International Conference on Tools with Artificial Intelligence*, pages 558–567, 1997.

[Cut10] Justin Cutroni. *Google Analytics*, volume 1. O'Reilly Media, Inc., first edition, 2010.

[DJ10] Easley David and Kleinberg Jon. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press, New York, NY, USA, 2010.

[EK08] Andreas Eisingerich and Tobias Kretschmer. In e-commerce, more is more. *Havard Business Review*, 86(3), 2008.

[EKSX96] Martin Ester, Hans P Kriegel, Jorg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[Exp09] Explorable.com. Statistical correlation, 2009. Available at https://explorable.com/statistical-correlation, accessed last time on January 2016.

[Fou08] The Apache Software Foundation. Apache http server documentation, 2008. Available at http://httpd.apache.org/docs/1.3/logs.html#accesslog, accessed last time on January 2016.

[Für02] Johannes Fürnkranz. Web structure mining: Exploiting the graph structure of the world-wide web. *OGAI Journal (Oesterreichische Gesellschaft fuer Artificial Intelligence)*, 21(2):17–26, 2002.

[GA05] Daniela Godoy and Analia Amandi. User profiling in personal information agents: a survey. *The Knowledge Engineering Review*, 20(04):329, 2005.

[Gef02] David Gefen. Customer loyalty in e-commerce. *Journal of the association for information systems*, 3(1):2, 2002.

[Get03] Lise Getoor. Link mining: a new data mining challenge. *ACM SIGKDD Explorations Newsletter*, 5(1):84–89, 2003.

[GHP] Robert Gentleman, Kurt Hornik, and Giovanni Parmigiani. *Use R!* Springer.

[GSCM07] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User Profiles for Personalized Information Access. *The Adaptive Web*, 4321:54–89, 2007.

[HF08] Siping He and Meiqi Fang. Ontological User Profiling on Personalized Recommendation in e-Commerce. *2008 IEEE International Conference on e-Business Engineering*, pages 585–589, 2008.

[HKTR04] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[Hyd05] Jim Hydzik. The revolution is just beginning. *Total Telecom*, page 32, 2005.

[JK12] Faustina Johnson and Santosh Kumar Gupta. Web Content Mining Techniques: A Survey. *International Journal of Computer Applications*, 47(11):44–50, 2012.

[KB00] Raynd Kosala and Hendrik Blockeel. Web mining research: A survey. *ACM Sigkdd Explorations Newsletter*, 2(1):1–15, 2000.

[Kle99] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[KS10] P Ravi Kumar and Ashutosh K Singh. Web structure mining: Exploring hyperlinks and algorithms for information retrieval. *American Journal of applied sciences*, 7(6):840, 2010.

[Leu07] K Ming Leung. Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 2007.

[Liu07] Bing Liu. *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media, 2007.

[Mat12] Kate Matsudaira. Data mining the web via crawling, 2012. Available on http://cacm.acm.org/blogs/blog-cacm/153780-data-mining-the-web-via-crawling/fulltext#, accessed last time at 28 of January 2016.

[MC10] Li Mei and Feng Cheng. Overview of Web mining technology and its application in e-commerce. *2010 2nd International Conference on Computer Engineering and Technology*, 7:V7–277–V7–280, 2010.

[MCS00] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on Web usage mining. *Communications of the ACM*, 43(8):142 – 151, 2000.

[MMN01] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-Boosted Collaborative Filtering. *Proceedings of the 2001 SIGIR Workshop on Recommender Systems*, page 9, 2001.

[Mob06] Bamshad Mobasher. Web usage mining. *Web data mining: Exploring hyperlinks, contents and usage data*, 12, 2006.

[Moh12] Sanjay Mohapatra. *E-commerce strategy: text and cases*. Springer Science & Business Media, 2012.

[MS04] Alessandro Micarelli and Filippo Sciarrone. Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. *User Modeling and UserAdapted Interaction*, 14(2-3):159–200, 2004.

[MSD04] Stuart E. Middleton, N.R. Shadbolt, and D.C. De Roure. Ontological User Profiling in Recommender Systems. 22(1):54–88, 2004.

[Naj09] M Najork. Web crawler architecture. *Encyclopedia of Database Systems*, pages 3–5, 2009.

[PB07] Michael J. Pazzani and Daniel Billsus. Content-based recommendation systems. *The adaptive web*, pages 325–341, 2007.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[PCP11] Ketul B Patel, Jignesh A Chauhan, and Jigar D Patel. Web Mining in E-Commerce: Pattern Discovery, Issues and Applications. *International Journal of P2P Network Trends and Technology*, 1:40–45, 2011.

[Pet09] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[PSM04] Gautam Pant, Padmini Srinivasan, and Filippo Menczer. Crawling the Web. *Web Dynamics*, pages 153—-177, 2004.

[PUPL01] Alexandrin Popescul, Lyle H Ungar, David M Pennock, and Steve Lawrence. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. *Artificial Intelligence*, 2001:437–444, 2001.

[PV12] Ladislav Peska and Peter Vojtas. Evaluating various implicit factors in e-commerce. *CEUR Workshop Proceedings*, 910(Rue):51–55, 2012.

[Rav10] Ashutosh Kumar Ravi, Kumar and Singh. Web Structure Mining: Exploring Hyperlinks and Algorithms for Information Retrieval. *American Journal of Applied Sciences*, 7(6):840–845, 2010.

[SA13] Ahmad Siddiqui and Sultan Aljahdali. Web Mining Techniques in E-Commerce Applications. *International Journal of Computer Applications*, 69(8):39–43, may 2013.

[SC00] Jim Sterne and Matt Cutler. E-Metrics: Business Metrics for the New Economy. page 61, 2000.

[SF98] Myra Spiliopoulou and Lukas C Faulstich. Wum: A web utilization miner. In *International Workshop on the Web and Databases, Valencia, Spain*. Citeseer, 1998.

[Sta16] Statista. Number of e-commerce transactions worldwide 2011-2015, 2016. Available at http://www.statista.com/statistics/369333/number-ecommerce-transactions-worldwide/, accessed last time in February 2016.

[TK11] Efraim Turban and David King. *Overview of electronic commerce*. Springer International Publishing, 2011.

[TPSA07] Sandeep Tata, Jignesh M Patel, Computer Science, and Ann Arbor. Estimating the Selectivity of tf-idf based Cosine Similarity Predicates. 36(4):75–80, 2007.

[TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[WHF07] Kangning Wei, Jinghua Huang, and Shaohong Fu. A survey of e-commerce recommender systems. In *Service Systems and Service Management, 2007 International Conference on*, pages 1–5. IEEE, 2007.

[ZDH07] Huang Z., Zeng D., and Chen H. A comparison of collaborative-filtering algorithms for ecommerce. *IEEE Intelligent Systems*, 22(5):68–78, 2007. cited By 89.

[ZS08] Qingyu Zhang and Richard Segall. Web Mining: a Survey of Current Research, Techniques, and Software. *International Journal of Information Technology & Decision Making (2008)*, 07(January), 2008.