FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Machine Learning Algorithms for Visual Pattern Detection on Web Pages

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Information and Knowledge Management

eingereicht von

Iraklis G. KORDOMATIS, BSc BSc BSc
Matrikelnummer 0754001

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Univ.-Prof. Mag. Dr. Reinhard Pichler
Mitwirkung: DI Christoph Herzog

Wien, 19.06.2013

(Unterschrift Verfasser) (Unterschrift Betreuung)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Machine Learning Algorithms for Visual Pattern Detection on Web Pages

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Information and Knowledge Management

by

Iraklis G. KORDOMATIS, BSc BSc BSc
Registration Number 0754001

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Univ.-Prof. Mag. Dr. Reinhard Pichler
Assistance: DI Christoph Herzog

Vienna, 19.06.2013

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Iraklis G. KORDOMATIS, BSc BSc BSc
K-Ebersdorfer-Str. 90/11/63, 1110 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser)

Acknowledgements

First of all, I would like to thank my supervisor Univ.-Prof. Mag. Dr. Reinhard Pichler for his advice and help with this master thesis. In addition, I want to thank Robert Baumgartner for his support, efforts and assistance during the creation of this thesis. I appreciate the time he has sacrified for the TAMCROW project and I am grateful for the valuable input he gave to my team.

My sincere thanks also go to the TAMCROW team, especially Christoph Herzog and Ruslan Fayzrakhmanov for their overall support. What is more important, they make the time we spend together in the weekly meetings and verbose discussions enjoyable by creating a humorous and interesting atmosphere.

Moreover, I thank Peter Filzmoser from the statistic department for taking time to discuss statistical questions in detail with me.

Additionally, I want to thank my parents Martha and Georg who have always supported me for more than 25 years. Last but not least, I thank my girlfriend Carolin for being awesome.

Abstract

In this thesis the question how to robustly identify web objects across different sites is tackled. TAMCROW introduces a novel approach exploiting visually perceivable characteristics of a web object and its surrounding objects. This approach is entirely independent of textual labels, and hence has the noteworthy advantage of being language-agnostic. Another main advantage of the visual detection approach is sample parsimony. Fewer examples are required for the learning process to learn how to find certain web objects on previously unknown pages. Moreover, visual cues are crucial for the human perception and as a consequence also for the usability of a web page. Therefore, web designers create web pages coherent with the human perception in order to yield a high usability.

Supervised machine learning techniques are applied for the object identification process. The knowledge is limited to features representing the visual appearance of the different web objects. An additional question is whether it is possible to predict the role of a web object by its visual appearance which is formally a classification problem. Within the scope of this master thesis, the following machine learning techniques are investigated in detail: logistic regression, k nearest-neighbor, classification trees (in particular, c4.5 of Quinlan) and support vector machines. For support vector machines the following kernels are applied: linear, polynomial, radial and sigmoid. Furthermore, different techniques for data preprocessing/preparation and parameter optimization for some of the classification techniques mentioned above are discussed.

Other scientific papers solve similar problems with either a rule-based approach (see [24, 37, 94]) or like this master thesis, with machine learning techniques (see [56, 74, 75]). In general, it is not possible to compare the results of these scientific papers directly since the web page corpora and aims are differnt.

The evaluation results are illustrated in chapter 7 indicating that the approach developed within the TAMCROW project is very fruitful. The workflow on web object identification is evaluated with different scenarios. These scenarios include searches for train, bus and flight connections as well as for accommodations. K-page cross-validation is used as evaluation technique. The mean precision is chosen as performance measure, since it fits best for the used scenarios. The results are significant for all classification techniques. In particular, the support vector machines with the radial and polynomial kernel functions achieve remarkable results.

Possible reasons for the favorable classification results are the following ones: Firstly, the TAMCROW project uses a vast number of visual features especially compared with other approaches. Therefore, the different classification algorithms seem to learn more easily how to distinguish between the different web objects. Secondly, the methodology of the distance computation (introduced in chapter 4) helps to exponentially increase the number of positively classified observations on the one hand and makes some features numerically comparable on the other hand (e.g. color, text). Thirdly, the postprocessing applied after the classification makes the results very robust against misclassifications.

Kurzfassung

Diese Masterarbeit bearbeitet die Forschungsfrage, wie Webobjekte auf bisher unbekannten Webseiten robust identifiziert werden können. Das TAMCROW-Team stellt einen neuen Ansatz vor, um visuelle Charakteristika von Webobjekten und deren umliegenden Objekten zu verarbeiten. Dieser Ansatz ist unabhängig von Text-Labels und dadurch auch sprachunabhängig. Ein weiterer Vorteil ist, dass er mit einer geringeren Anzahl an Beispielen auskommt. Desweiteren sind visuelle Merkmale essentiel für die menschliche Wahrnehmung. Webusability hängt von dieser Wahrnehmung ab. Da der Erfolg eines Webauftritts von einer hohen Benutzerfreundlichkeit abhängt, sind Webdesigner bemüht, ihre Webseiten für die menschliche Wahrnehmung zu optimieren.

Für den Prozess der Objektidentifizierung werden supervised Machine-Learning-Techniken eingesetzt. Dabei ist das Wissen auf Eigenschaften beschränkt, welche die visuellen Attribute eines Webobjektes beschreiben. Eine zusätzliche Frage ist, ob es möglich ist, dass Webobjekte anhand von ihrer visuellen Erscheinung klassifiziert werden können. Im Rahmen dieser Masterarbeit wurden folgende Machine-Learning-Techniken im Detail für den oben beschriebenen Einsatz untersucht: Logistische Regression, K-Nearest-Neighbor, Klassifizierungsbäume (c.4.5 von Quinlan) und Support-Vector-Machines. Für die letztgenannte Technik wurden folgende Kernelfunktionen verwendet: linear, polynomisch, radial und sigmoid. Darüber hinaus, werden unterschiedliche Techniken für die Datenaufbereitung und die erforderlichen Parameteroptimierungen für einige der oben beschriebenen Techniken erläutert.

Andere wissenschaftliche Ansätze lösen ähnliche Probleme mit einem regelbasierten Ansatz (siehe [24, 37, 94]) oder wie in dieser Masterarbeit mit Machine-Learning-Techniken (siehe [56, 74, 75]). Grundsätzlich ist es nicht möglich die Resultate der anderen Arbeiten direkt zu vergleichen, da die Webpage-Korpora und die Ziele nicht genau übereinstimmen.

Die Resultate dieser Arbeit finden sich im Kaptel 7. Sie zeigen, dass der Ansatz des TAMCROW-Teams äußerst vielversprechend ist. Der Workflow für die Identifizierung von Webobjekte wird in mehreren Szenarien evaluiert. Diese Szenarien beinhalten eine Suche nach Bussen, Flügen und Zügen, sowie eine für Unterkünfte. Im Weiteren wurde eine k-page cross-validation angewendet, um die Ergebnisse zu bewerten. Als Leistungsmerkmal wurde der Mittelwert der Präzision (precision) verwendet. Die Resultate sind für alle Klassifikationstechniken beachtlich. Besonders die support vector machine mit der radialen und polynomischen Kernelfunktion kön-

nen durch exzellente Ergebnisse überzeugen.

Diese beachtlichen Klassifikationsraten basieren auf folgenden Begründungen: Das TAMCROW-Projekt verwendet eine vielfältige Auswahl an visuellen Eigenschaften, besonders im Vergleich mit anderen Arbeiten. Es scheint, als ob die Klassfikationsalgorithmen dadurch leichter zu unterscheiden lernen. Weiters ist die Verwendung einer Distanzberechnung, welche im Kapitel 4 erklärt wird, zum einen äußerst hilfreich, um die Anzahl der positiven Beobachtungen zu erhöhen und zum anderen werden einige Eigenschaften dadurch erst quantifizierbar (z.B. Farbe und Text). Als zusätzlichen Punkt lässt sich anführen, dass durch das Postprocessing die Ergebnisse der einzelnen Klassifikationsalgorithmen robust gegen falsche Klassifizierung werden.

Contents

Abbreviations

1-NN	1-Nearest-Neighbors
BI	Business Intelligence
BGM	block-based gemoetric model
BOM	browser object model
CI	confidence interval
DC	Distance Computation
DM	Data Mining
DOM	Document Object Model
FE	Feature Extraction
IM	interface model
IQR	interquartile range
IR	Information Retrieval
IRLS	iteratively reweighted least squares
kNN	k Nearest-Neighbors
LR	logistic regression
MDR	Mining Data Records
ML	Machine Learning
NP-ratio	negative positive ratio
OLSR	ordinary least square regression

PCA	Principal Component Analysis
QntBGM	quantitative BGM
QltBGM	qualitative BGM
RBF	Radial Basis Function
SBGM	structural BGM
SVM	support vector machines
UOM	unified onotological model
VPM	visual perception model

Publications

Following the good practice at DBAI, some results of this work have already been published and presented to the scientific community by the TAMCROW team and the author of this master thesis:

- **Web object identification for web automation and meta-search.**
  I. Kordomatis, C. Herzog, R.R. Fayzrakhmanov, B. Krüpl-Sypien, W. Holzinger, and R. Baumgartner.
  International Conference on Web Intelligence, Mining and Semantics (WIMS 2013), Madrid, Spain, 12–14 June 2013, Article No. 13. ACM, 2013.
- **Feature-based object identification for web automation.**
  C. Herzog, I. Kordomatis, W. Holzinger, R.R. Fayzrakhmanov, and B. Krüpl-Sypien.
  The 28th Annual ACM Symposium on Applied Computing (SAC'13), Web Technologies Track, Coimbra, Portugal, 18-22 March, 2013, pages 742–749. ACM, 2013.

CHAPTER 1

Introduction

This scientific work was composed within the scope of the TAMCROW [2] project and addresses the problem description of identifying web objects with the help of their visual appearance on a previously unknown web page (details can be found in 1.2).

A real life example shall make the formal problem description more apperceptive. Imagine several web pages where you are able to perform a search over flight connections. A human user who has already booked some flights online, would be rather familiar with such search forms on a web page. Therefore, there is a high probability that such a person is also able to perform a flight connection search on a website which is new for this person. Now, think of that scenario being performed by a computer. Based on this idea the following real-life use cases can be introduced:

- *Meta-search:* The basic idea behind a metasearch is that it forward request to several other databases or search engines without performing a search itself. The results of the inquired databases and search engnines are then merged and presented to the user as one search result. The main advantages are that the user saves searching time, can easier compare the results from different sources and get results from new sources (when he has not known them before).

- *Screen reader:* Their main objective is to transform a screen from a computer in a format that can be read out (also known as text-to-speach) or represented on a Braille output device. This task appear easy for applications like transforming continuous text (e.g. a whole book), but the task becomes rather delicate when it comes to web pages.

In scientific papers, such problems are referred to web form understanding or meta searches and they are quite popular in the scientific community. The following papers are a short selection which illustrate this. This master thesis focuses on Machine Learning (ML) techniques for addressing such problems. An complete list of the techniques used can be found in chapter 5.

This chapter is structured as follows: The first section gives an overview of the structure of the master thesis. Then secondly, the problem description is defined. Thirdly, related work

gives an overview of the scientific work which has been done in similar research areas. The next section gives an overview of the project in which the master thesis is embedded. Then a brief overview of the preceding project is shown. This is limited to its achievements. Finally, the methodology for this scientific work is discussed.

1.1 Structure of the Master Thesis

This section illustrates how the master thesis is structured.

Chapter 1 introduces the master thesis. It defines the problem description and explores related scientific works. In further sections, the integration into the scientific project TAMCROW [2] is explained. In addition, the motivation, the methodology and other topics are discussed.

Chapter 2 illustrates the workflow of addressing the described problem within this master thesis and the TAMCROW project. In addition, some test scenarios are introduced with which the approach is evaluated.

In 3 the visual perceivable features of the web objects are introduced. At first, details about the extraction from the annotated web pages are provided. Then, a complete list of these features is given.

Next, in chapter 4 the transformation from visual features to distances features are discussed. This further processing becomes crucial since some features in its raw representation are not helpful for classification algorithms.

Chapter 5 discusses well-known classification methods and explains their usage in the project, as well as the used libraries. The following ML techniques are discussed in detail: the logistic regression, the c4.5 decision tree, the k Nearest-Neighbors (kNN) and the support vector machines (SVM) with linear, polynomial, radial and sigmoid kernel functions.

Chapter 6 introduces how the workflow of the web object identification is evaluated. As performance measure the mean precision has been choosen together with the corrected resampled t-Test for computing the confidance intervals of the mean precision. The latter is crucial to determine the statistical significance of the results. In addition, more information about the experiments and their settings are discussed.

Chapter 7 shows the benchmarks of the used ML classifiers. The first part deals with the parameter estimation for the different classifiers. In the second part of this chapter, further results ware illustrated which have been performed with the parameter settings from the previous part. The third and last part of this chapater deals with an analysis of the visual perceivable features of the different web objects.

The conclusion of the master thesis is provided in chapter 8.

The appendix is then divided into five parts. The first one gives an overview of the annotated web pages. Secondly, the data set for the illustrations are shown. Thirdly, details for the features of the web objects are presented. The fourth part provides detailed results from chapter 7. The last part illustrates some statistical methods and techniques which were used in the master thesis.

1.2 Problem Description

This master thesis deals with the problem of the identification of web objects from new or unknown web pages by using information from a set of known websites and their elements based on their visual appearance. The main motivation behind this approach is that human have a similar perception. For example, a user has almost no difficulties to detect the navigation menu on a web page in a foreign language, since he or she is used to the visual appearance of the design paradigms which most web designers obey. It seems that ML provides a set of promising techniques, therefore it is worth to implement a solution based on ML techniques to address the problem of web object identification.

1.3 Methodology

The methodology which this master thesis follows is the *design science* approach. There are several scientific papers about this methodology. Hevner did significant work in this field and published some papers which can be suggested as the standard literature in design science. The most recent and relevant ones are [46–48].

The following subsections are going to describe and illustrate the main characteristics of the design science research cycle. For that purpose, the following sections are based on [46].

The design science research cycle

Figure 1.1 illustrates the design science research cycle. Hevner et al. draws the attention to three sub-cycles namely the relevant, design and rigor cycle. All of these three cycles are discussed in subsections below. The design science research cycle is of vital importance for the development of new and innovative IT systems, since research in computer science seldom stick to rigor and inflexible methodologies. Therefore, there is a need to establish a methodology and formalize it so that IT professionals know precisely what stands behind the design science research cycle (and what is not included).

Relevance Cycle

Design science is driven by the desire of improving the current state of a person's or entity's environment. This desirability of improvement can be seen as an abstract problem description. Prototypical improvements generated within an environment is referred to as artifacts by Hevner. The identification of such improvement potentials also known as threads and opportunities in Hevner's model is mostly the starting point of design science. Figure 1.1 shows that the relevance cycle contains field testing. Within this cycle, questions of the measurement for the current state and the improvements are addressed. In addition, the measurement itself is computed. As a result, it is possible to decide if a further iteration is required. It is essential to mention that such iterations are an ongoing process and are never really finished as opposed to some kind of waterfall model. As a consequence it is possible to evaluate different aspects for each iteration, so every iteration is unique.

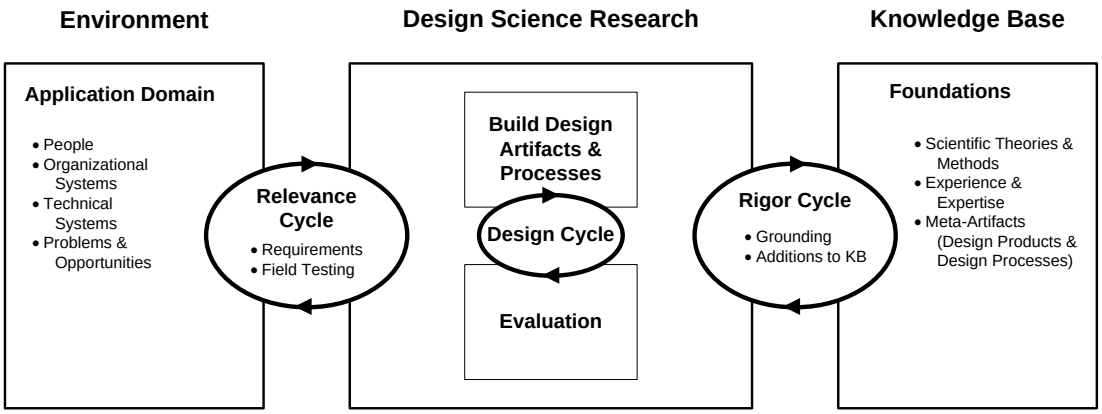


Figure 1.1: Design Science Research Cycles [46]

Rigor Cycle

The rigor cycle is the source of scientific and other knowledge. It is acquired from experiences, past experiments, scientific papers, other experts, state-of-the-art techniques (or best practises), theories and others. Mostly, the scientific knowledge is provided by researchers in scientific papers, which have to be analyzed and evaluated before they can be applied for self-generated solutions. Hevner argued that this can be a constraint, since papers considered being as too new, innovative and using very creative approaches are often rejected by top journals due to their lack of grounding theories.

Design Cycle

The design cycle is the part, where most work is done. The phase of constructing and evaluating is iteratively done until a satisfying state is achieved. For generating considerable performance, it is compulsory that the input from the relevance and rigor cycle are considered. A balance between the constructing and evaluation phase should be maintained over the whole design science research cycle. This is a great challenge and has to be reviewed in constant periods. Finally, it should be mentioned that several iterations are mandatory to overcome the pitfalls of the design science research approach and essential to make significant contribution to science.

1.4 Motivation

There are a series of reasons why the author of this master thesis was motivated to pursue this topic. Firstly, it has to be mentioned that this scientific work is application-oriented in a way, that the results can be used for improving existing implementations or creating new ones. One of that might be to provide blind people with technology for support by filling out web forms, another could be to enhance automated meta-search engines, etc. Secondly, Dr. Robert Baumgartner delivered state-of-the-art problems in his lecture with the name *Applied Web Data Extraction*

and Integration in such an enthusiastic way that the author of this master thesis could not resists choosing one of his suggested topics for a master thesis. Thirdly, it perfectly fits within my interests, education and knowledge about ML techniques and experiment with such techniques.

1.5 Contribution to the scientific Community

The main contribution within this master thesis is to provide a workflow and a clear concept how the problem of web object identification can be addressed with success. Furthermore, it describes the test settings and evaluates methods particularly detailed compared with other scientific works (see [24, 56, 74, 75, 94]). In order to follow good scientific practises all result are replicable and can be used by other scientists.

1.6 Relation to the TAMCROW Project

The TAMCROW-Project [2] is an acronym for task mining and crowed sourcing. Its project leaders are Reinhard Pichler and Robert Baumgartner.

The web is a medium whose development never stops. In this way it is behaving rather like an organism than a standardised medium. Compared with television and radio, it is more flexible and less regulated by public authorities. This makes the web itself more innovative and more difficult to predict. Also the consummation of the web adapted compared to its beginning by the use of social platforms, mobile devices and others.

TAMCROW aims to respond to this changes and manages new challenges on the Web. Moreover, it is believed to be crucial to approach by an web science perspective. Therefore, the TAMCROW initiates and suggests a model which characterizes user behavior of various crowds in the web. This model can be applied to several use cases for user agents in such crowds. Those use cases are addressing web accessibility, mobile browsing and web personalization.

As output several prototypes has been developed to address different needs (blind users, targeting content and state repackaging for mobile devices, personalization trails as well as automatic deep web extraction.)

1.7 State of the Art and Related Work

This section gives an overview of scientific publications describing similar approaches as well as different approaches for a similar problem. Most works deal with web object identification. Some of them implicit others explicit. It is important to mention that most of the supervised and Machine Learning (ML) wrappers pursue the goal of extracting information from similar structured web pages. This is a web page specific approach. However, the TAMCROW approach is to use visual characteristics in a domain specific context. The idea is that visual characteristics are similar for web pages within a domain (e.g. travel and hotels).

Wrappers for Information Extraction

A prominent approach for extracting information from web pages are *wrapper* programs. There are several scientific works discussing this topic. Chang [16] defines wrapper programs as component of an information integration system which connect to an information source via the HTTP protocol.

Wrapper systems can be grouped into manual, automatic and semi-automatic techniques (see [19]). That grouping name describes how the wrapper is generated. Manual approaches have the drawback that the wrapper implementation is rather time-consuming compared to the other two groups. As a result, manually constructed wrappers have lost popularity. Examples are *Jedi* [51] and *PiLLoW* [15].

In contrast to the group mentioned above, fully automatically generated wrappers have gained more and more attention. In the work of Crescenzi at al. [21] it is shown how the likeness and variance of different web pages can be used to automatically build a wrapper. The name of this technology is RoadRunner. Its advantage is that it does not need labelled examples. The Mining Data Records (MDR) [93] uses string-matching and tree alignments to find reocurring information on a web page. Another approach is to use an ontology for several domains in order to extract information from randomly selected web pages [25]. The main benefit of fully automatical techniques is that they can be applied on entirely new documents.

The generation of semi-automatic wrappers strongly depends on the user's input. This group of wrappers can be divided into subgroups *wrapper induction* and *wrapper specification*. The first ones try to create a wrapper with machine learning techniques from training samples provided by the user. Examples of this method are WIEN [61] and Stalker [71]. The latter method (wrapper specification) creates wrappers from interactive user inputs. Examples of this approach are Lixto [9], DEByE [62] and Wargo [82].

Besides the above mentioned approaches for generating wrappers for extracting information from web pages, it has also become important to maintain existing solution in recent years. Outgoing from the desire that a wrapper should be robust against changes on the extracted web page, a new area of wrapper development gained attention, namely *automatic wrapper adaptation*. A fruitful approach was published by Ferrara et al. in [32], where an improved matching method for tree-edit-distances is introduced to compare similarity of trees.

In general, semi-automatic approaches yield a higher precision than the fully automatic approaches. Semi-automatic techniques are sometimes called *supervised* in scientific works. Some authors therefore introduced the term *user-guided* wrapper generation for the wrapper specification in order to distinguish between this generation method and the wrapper induction.

Document Analysis

In comparison to HTML web pages, which are considered as semi-structured documents, information extraction from pdf files needs different approaches since the structure of the document

is harder to extract. It seems that this is a rather young field of research as first recognized publications are from the year 2006 [34]. Fazzinga et al. [31] provide a technique based on disjoint nested fuzzy logic conditions. Hassan published a graph-based approach in [42]. This latter approach is less expressive compared to the one before. However, this fact also reduces the complexity of the wrapper generation significantly.

Analysing the Visual Representation of Web Pages

Basically, the following papers discuss a classification problem, for either special regions of the websites or the website as a whole.

An application of the former can be found in [40]. Here the author used a support vector machines (SVM) technique which tries to the determine the suitability of certain web pages for teenagers based on the text, the visual and hierarchical content.

For the latter case, there are a number of scientific works which try to separate the website into blocks [23], [54], [74] and then extract information [54] or classify them [74]. In [54] a visual block segmentation in combination with a clustering algorithm is used to differentiate between important and redundant information. The work of Chen [17] is similar to the one mentioned before, but in contrast uses a SVM technique. In this case, the problem is reduced to pure topic identification. In contrast, in [23] a mixture of text, meta-data and pictures are used to identify relevant blocks. This was realized by using a SVM. In [74] visual segmentation is additionally used. Like in TAMCROW [2] color, font family, font size and parent level features are extracted from the website. This and other information are used in combination with a SVM to identify the blocks and classify them in a second step.

It is also possible to identify objects with a rule-based approach (see [24, 37, 94]). It might seem that these methods are too rigid and not robust enough. However, Furche et al. in [37] receive impressive results regarding web form understanding with their *ontology based web pattern analysis with logic (OPAL)* technology. It is interessting to see that their approach uses visual, textual and structural features to be independent of labeling as well as the TAMCROW approach[1]. The domain dependend knowledge is added through an ontology in their case. In contrary, the TAMCROW approach is knowledge independent (for details see chapter 7).

Applications

This subsection provides a short overview of some applications which have been developed in recent years.

The software DIADEM published in Furche et al. [36,38] is a meta-search technology which applies the problem of object understanding to search engines. In general, people search for objects on the web and not for words occurring on an HTML web page (e.g. a good dentist near someone's workplace or home). Therefore, if a system can query several databases and classify the objects on the resulting pages, it is able to provide that information to users through a single

[1]The textual features are limited in order to be language independent.

search mask. DIADEM provides a solution for this problem.

The meta-search engine AllInOneNews from Liu et al. published in [63] demonstrates how large-scale news meta-search engines can be developed in order to work efficiently. The paper evaluates their approach with other meta-search engines (Google News and Mamma News) and claims that their approach has advantages considering its effectiveness due to their search engine selection technique, semantic-based matching and high degree of automation. However, Liu et al. admit that Google News is significantly better at providing diversity regarding the news topics as well as at removing/merging redundant results.

In contrast to the applications mentioned above which have been taken from the area of meta-search engines, the work of Bukhari et al. in [14] introduces an application for assisted information extraction designed for visually impaired or blind web users. The published software has the name *Heavyweight Ontology Based Information Extraction for Visually impaired User (HOIEV)*. As the acronym suggests, the approach uses an ontology. In addition to the information extraction, it provides a set of different tools like voice command parsers and domain ontology extractors.

In [65] Mahmud et al. published a work about a context-driven web browser targeted on the visually impaired web community. The main idea is that the browser tries to extract the most important information of a web page which is linked from the current web page. This can significantly save browsing time, since not every link needs to be clicked on and followed by the visually impaired user. The evaluation of the importance is done by a machine learning technique (SVM).

Screen readers are devices for visually impaired users which help to recognize information and text on a monitor. They transfer their output either to a Braille device or to the speakers (also referred to as text-to-speech). The web object identification workflow from TAMCROW can be integrated into screen readers or web browsers. This can be done in such a way that it guides the user through similar web pages (e.g. giving advice for filling out web forms).

In addition, the TAMCROW technique can be used to enhance current approaches for schema-matching in the area of web applications. With our approach it is possible to determine which role a web object has on its web page. This information can then be used to find matching web objects (web objects with the same role) on new web pages within the same domain. Current approaches are based on a semantic matching which is derived from textual information [44]. An introduction about schema-matching and its application in databases and ontologies can be found [85].

CHAPTER 2

Approach

This chapter introduces evaluation scenarios which have been derived from the problem description in chapter 1. The main purpose of these scenario is to evaluate a workflow for web object identification. Chapter A in the appendix will give further information regarding the annotated web pages of each scenario.

In addition, the workflow for the identification of web objects on web pages is introduced. The main parts of this workflow are a general approach for the data extraction, followed by the distance computation to the training and evaluation of the classifiers. A standardised process is necessary to ensure the same test-environment for every test-run. Otherwise, the different runs would be biased and difficult to compare with each other[1].

The main part of this master thesis deals with the training and evaluation of the different Machine Learning (ML) classifiers. These parts are covered in the chapters 5, 6 and 7 in detail.

Chapter 2 is structured as follows: firstly a general approach is shown and described (in section 2.2). Secondly, the scenarios which had been tested in the TAMCROW project [2] are elaborated.

2.1 Scenarios for Evaluation

Scenarios in general are sketches of real world problems. This simulation of real world problems is necessary in order to test the proposed simulation workflow. Therefore, also the TAMCROW team has defined such scenarios. This section will introduce them. The scenarios are from the nature of a meta search. In sum there are five scenarios which can be divided into two groups. The first is a transport connection search, whereas the latter is an accommodation search. The transport connection search itself consists of three scenarios namely, a search for bus, flight and train connections. Figure 2.1 demonstrates the hierarchy of the scenarios graphically.

[1] Technically, it would be possible to compare them, however, the interpretation made from the biased evaluation will be vague and inaccurate.

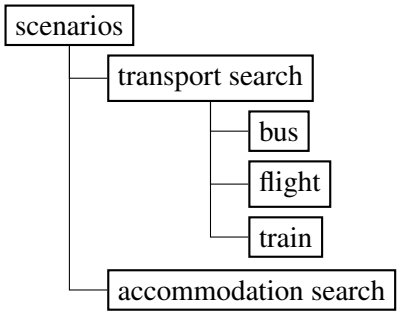


Figure 2.1: The hierarchy of the scenarios

Each scenario group has a selection of special web objects. *Special* within this context mean that they are from certain importance in order to perform the certain tasks within a scenario. In the further work this group of objects is referred to as *functional objects* when speaking of their function within a web page, while these web objects are referred to as tasks in the chapters about classification and evaluation since the aim here is to correctly classify the respective web objects. The following list will illustrate this for the transport connection searches. (See figure 2.2 for a screenshot with the annotated functional objects.)

Figure 2.2: Example of the functional web objects on two web pages for the flight search scenario. The numbers correspond to the list for the bus, flight and train connection search

1. *Departure location:* This is usually a text field where the user can enter the start of his or

her journey.

2. *Arrival location:* Also a text field, which indicates the goal of the users' journey.
3. *Departure date:* The date when the journey starts.
4. *One-way trip:* This is normally a check box which indicates that the user is looking for a connection without return.
5. *Number of adult passengers:* This is an input field where the users enters the number of adult passengers participating in this journey. Here several types are possible (e.g. drop down menu, text box)
6. *Submit button:* This input field is the button which sends the request from the user to the web server. It is mostly from the type of the classical submit button or a picture.

On the other hand, the accommodation scenarios use the following functional web objects (Examples are shown in figure 2.3).

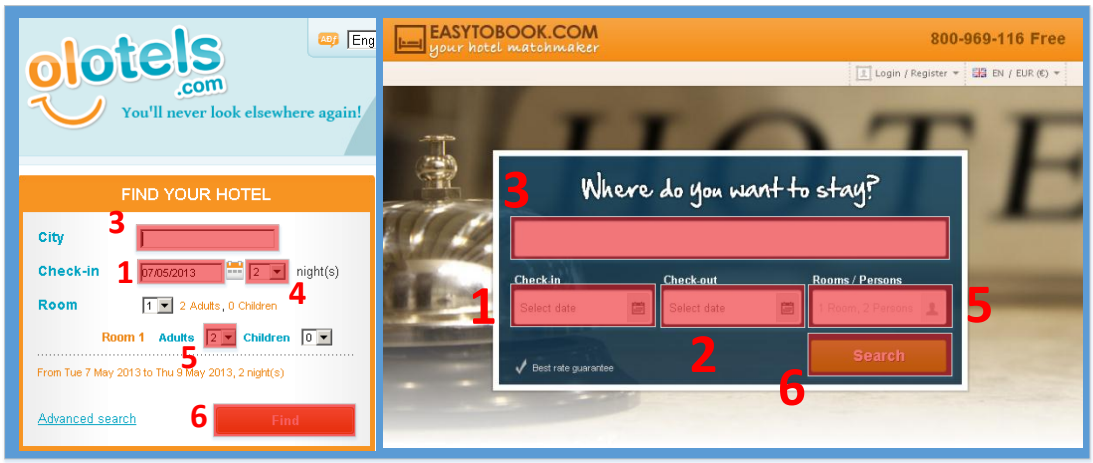


Figure 2.3: Example of the functional web objects on web page for the accommodation scenario. The numbers correspond with the list for the accommodation search

1. *From Date:* The date from which the accommodation should be reserved.
2. *To Date:* The date until which the guest would like to stay. In the most scenarios either this field or the number of nights were possible to enter, but not both. This makes perfect sense for human users, however, these cases have to be divided in order to evaluate ML algorithms.
3. *Where:* This is the preferred location of the accommodation.

4. *Nights:* The nights field indicate how many nights the user wants to stay.

5. *Number of adult passengers:* This field indicates the requested size of the room (or number of beds). Another possibility is to ask for the number of single/multi bed rooms. Since this alternative is very rare, it has not be included into this scenario group.

6. *Submit button:* This has the same function as above.

Any object not given in the two lists above shall, despite of its practical necessity, deemed as input for the booking process and thus will be marked as *other* object[2]. In latter sections and chapters of this work, the search after every functional web object is called task.

A list of scenarios with its web pages can be found in the appendix in chapter A. There are tables which also indicate which *functional* web objects appear on a certain web page.

When taking a closer look at the right screenshot in figure 2.2 it can be observed that visual features as the width and length of the web objects can useful for classifying the different web objects. The left screenshot might suggest that a visual feature like the vertical alignment could be useful as well, since all marked features lay in the same vertical row of the web page. In addition, the right screenshot of figure 2.3 shows that the position relative to the web page can be rewarding as a visual feature as the marked objects are all within a rectangle.

2.2 Web Object Identification Workflow

This section introduces the workflow of the web object identification process in the TAMCROW project. The focus lies on the process itself and not on formal definitions or calculations. Details regarding these calculations can be found in the subsequent chapters.

Figure 2.4 gives an introduction of the whole web object recognition workflow as used in the TAMCROW project. The input of the process[3] are annotated web pages. Subsection 2.2 gives a short introduction how the annotation of web pages is done. Firstly, the features of the annotated and not explicitly annotated web objects are extracted. Details are provided in subsection 2.2 of this chapter, whereas the technical details are presented in chapter 3. Secondly, based on the features of the web objects the distances between two (in following referred to as pair) web objects is calculated. The particularities of this step are provided in subsection 2.2. The technical details are not presented in this chapter, they can be found in chapter 4. The last part deals with the classification process where classifiers are trained based on the distance pair vectors of the web objects. Afterwards, these classifiers are applied on new, previously unknown[4] web pages to classify the web objects on its web page. Details of this process are shown in this section, whereas the theoretical background is described in chapter 5 and the evaluation of the used scenarios from section 2.1 is provided in chapter 7. The full details of the evaluations can be found in the appendix in chapter D.

[2]This mean that objects from type *other* have no function within this scenario.

[3]workflow and process are used as synonym in this section.

[4]In this work, the web pages were technically not unknown or new. Some were excluded from the training set, but included in the test set. This is very similar to a k-Fold cross-validation. The main idea is to simulate a real world scenario, but without really knowing the correct class it is impossible to evaluate its own classifier.

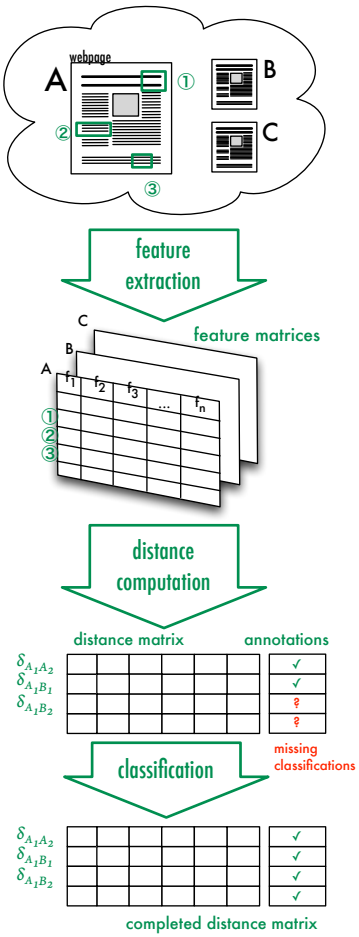


Figure 2.4: The workflow for identifying web objects (This figure is taken from [30])

Web Page Annotation

This subsection gives a brief introduction to the interface of a web page annotation.

Figure 2.5 shows the tool which was created in the TAMCROW project and used for the annotation for the web objects on the different web pages. In that figure, the web page of the Emirates airline is shown. The tool is designed as an eclipse plug-in, therefore similarities in design are not coincidental. In the middle the brightest rectangle (here the departure airport) is the selected web object. The biggest rectangle in purple color surrounding the brightest one, is the neighborhood of this specific web object (departure airport). The other rectangles are web objects within this neighborhood. On the upper left the Document Object Model (DOM) tree of the entire web page is shown. Below that, the reader can find the defined HTML attributes of this web object. On the right side the compiled features are given. This features are extracted for each web object on this web page. Details are provided in chapter 3.

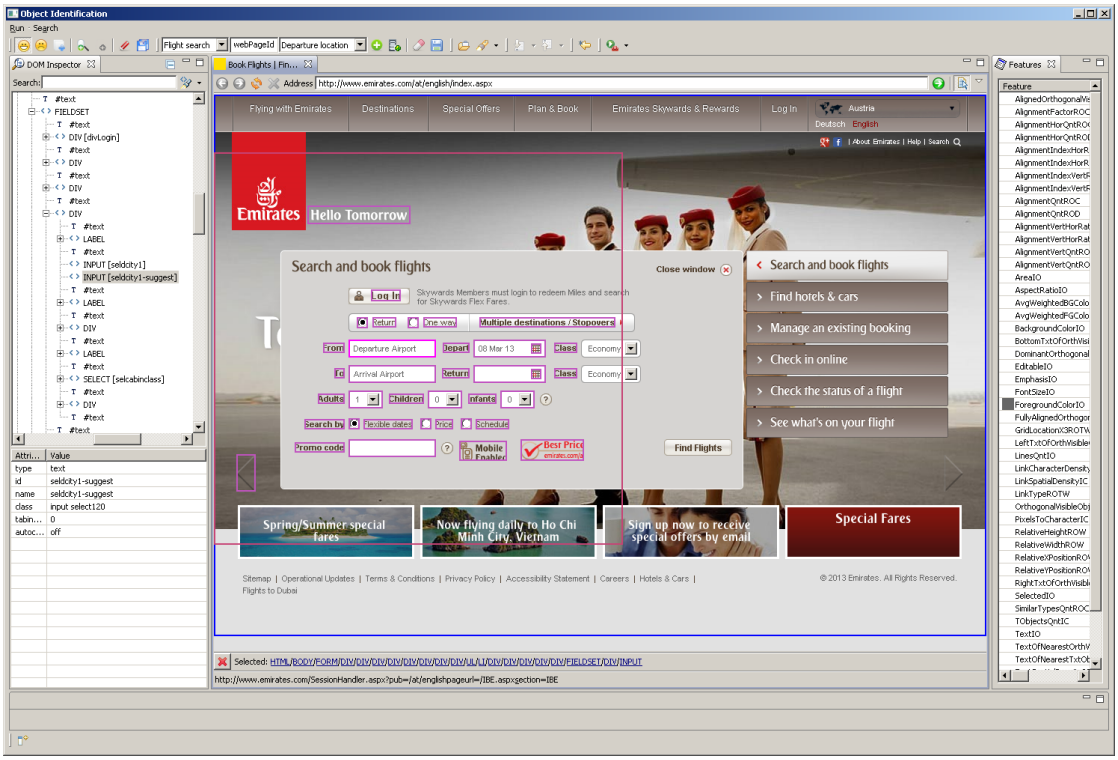


Figure 2.5: Screenshot of a web page annotation example

Feature Extraction Workflow

Figure 2.6 shows the Feature Extraction (FE) in detail. In general, the FE workflow consists of two sub processes. Firstly, the web page is analyzed. The annotated web pages are used as input, while the physical model is the output of the web page analysis. Secondly, the different visual features are computed. The previous output (the physical model) is used as input and a matrix, which provides the features of all web objects, is the resulting output. This matrix is also called feature matrix. The technical details can be found in chapter 3.

Distance Computation Workflow

The Distance Computation (DC) workflow is represented in figure 2.7. The following workflow is executed for every functional object mentioned in 2.1. The exact computation can be found in chapter 4.

One distance measure is computed for every feature, so the features are mapped 1:1 to the distances and the dimension for the data matrix remain constant. However, a distance vector is calculated between two web objects and represents one row in the resulting distance matrix. As a consequence, that increases the number of observations.

The main advantages of the distance computation are the following:

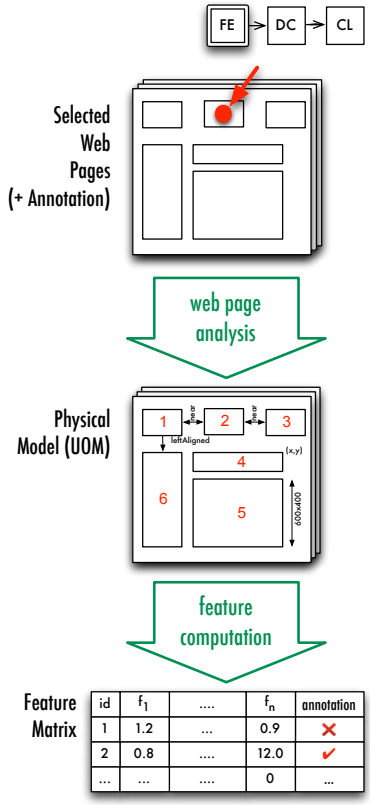


Figure 2.6: The feature extraction (FE) workflow (This figure is taken from [30])

- Due to the nature of the computation the number of observations is increased almost by the power of 2. This has the advantage that more data is available from which the classification algorithms can learn.
- Some features by its raw representation are not useful for classification. One example is the integer representation of the RGBA color code.

Firstly, for every possible combination of observation pairs[5] from the feature matrix, the distances to each other are calculated. Secondly, the rows are stored in a new matrix in the following referred to it as *distance matrix*. The combination of row pairs needs to be defined formally because technically not all possible row pairs are used. In general, every row of the feature matrix has a column indicating the type[6] of this object within a certain scenario. The labels for the functional types depend on the scenario. For every scenario of the *transportation search* this set of labels contains *departure date, departure location, arrival location, number of adult passengers, one-way, submit button, other*. This set can be formally defined as

[5] represented as rows

[6] Sometimes also referred as task.

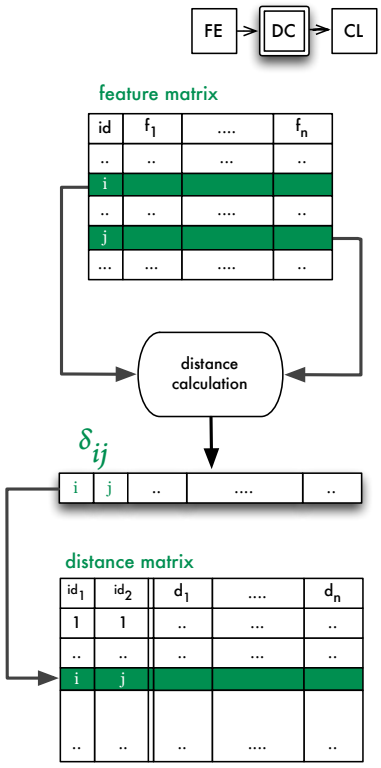


Figure 2.7: The distance computation (DC) workflow (This figure is taken from [30])

$T_t = \{DepDate, DepLocation, ArrLocation, Adults, OneWay, Submit, Other\}$

For the *accommodation search* this set of labels includes the following elements: *arrival date, departure date, nights, arrival location, number of adult passengers, one-way, submit button, other*, or formally written as

$T_a = \{ArrDate, DepDateor\#Nights, ArrLocation, Adults, Submit, Other\}$

Not all combinations of distance pairs of web objects are included in the distance matrix. Only those which fulfil the following formal definition $\forall x \forall y$ where $x, y \in M_F$ and $(x \in type \vee y \in type)$. M_F represents the set of web objects which are contained in the feature matrix.

The distance computation process merges the feature matrices of all web pages within a scenario and creates a distance matrix for every functional web object. Furthermore, a class column is added to each matrix. Details about the class column can be found in chapter 4 in section 4.3.

Technically, the distance matrix also contains information to identify each row. This information is the web page name and the ID of both web objects which are used to form the corresponding line in the matrix.

The exact computation methods and formulas for the distance computation can be found in chapter 4 . All necessary details are explained there.

Classification Workflow

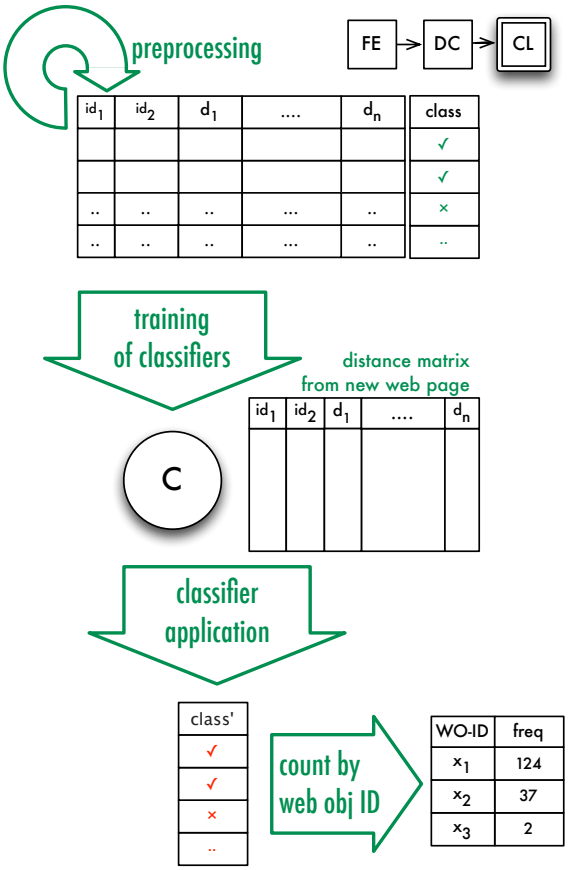


Figure 2.8: Classification (CL) workflow (This figure is taken form [30])

This subsection gives an overview of the classification workflow, which is illustrated in figure 2.8. This workflow starts where the distance computation workflow has ended namely with the distance matrix. At first, the values of the matrix are adjusted by a pre-processing step. This can either be a z-score[7] transformation, a relative rank transformation or no transformation at all. This matrix is then split into a training set and an evaluation[8] set. This is done via a k-page cross-validation[9](A detailed description how the k-page cross-validation exactly works can be found in chapter 6 in section 6.4.) The training data is then used to build the classifier[10]. With the evaluation set it is possible to validate the estimated classifier. The result is a vector indicating the class of each row of the distance matrix. It is important to note that each task within a scenario is treated as its own classification problem. Therefore, only one web object is

[7] also known as standardization

[8] sometimes also referred to as test set

[9] This term is a constructed word, it should illustrate cross-validation which is used for validating the different results.

[10] The singular in this sentence represents an arbitrary classifier. For evaluation several classifier were trained.

the searched object on an unseen web page. As the classification is applied on web object pairs, a technique has to be introduced to link the web object pairs to the searched web object. This method is introduced in chapter 6 in section 6.3.

CHAPTER 3

Visually Perceivable Features

This chapter introduces the technical definitions for the extracted features of the web objects. These extracted features cover aspects of the human perception and are therefore sometimes referred to as visual perceivable features. Some of these concepts are defined in more detail in the following work [30].

A web page consists of several different kind of stereotypical elements like forms, navigation bars, menus and others. Identifying[1] such elements automatically appears as an important task in many areas in computer science. The goal of the TAMCROW project [2] is to address the problem by modelling the visual perception of a human [39, 59, 60]. The main approach is to embrace some spatial, visual and textual characteristics in order to create an abstract model apart from the technical realization.

This chapter is structured as following: Firstly, the unified ontological model is introduced. Secondly, structural elements of web pages are described. Finally, the different visual perceivable features are listed.

3.1 The Unified Ontological Model

In the ABBA[2] [1] and TAMCROW [2] projects the unified onotological model (UOM) [28, 29, 60], a domain specific ontology, was defined. It is used for information extraction on web pages and web page understanding.

Figure 3.1 illustrates the UOM. It consists of two parts, namely a physical and a logical model. The first one is composed of the interface model (IM), block-based gemoetric model (BGM) and the visual perception model (VPM). The latter implements an interpretation to the ideas of the physical model.

[1] Used in this context as synonym of recognizing for linguistic purposes.

[2] ABBA is the predecessor project of TAMCROW.

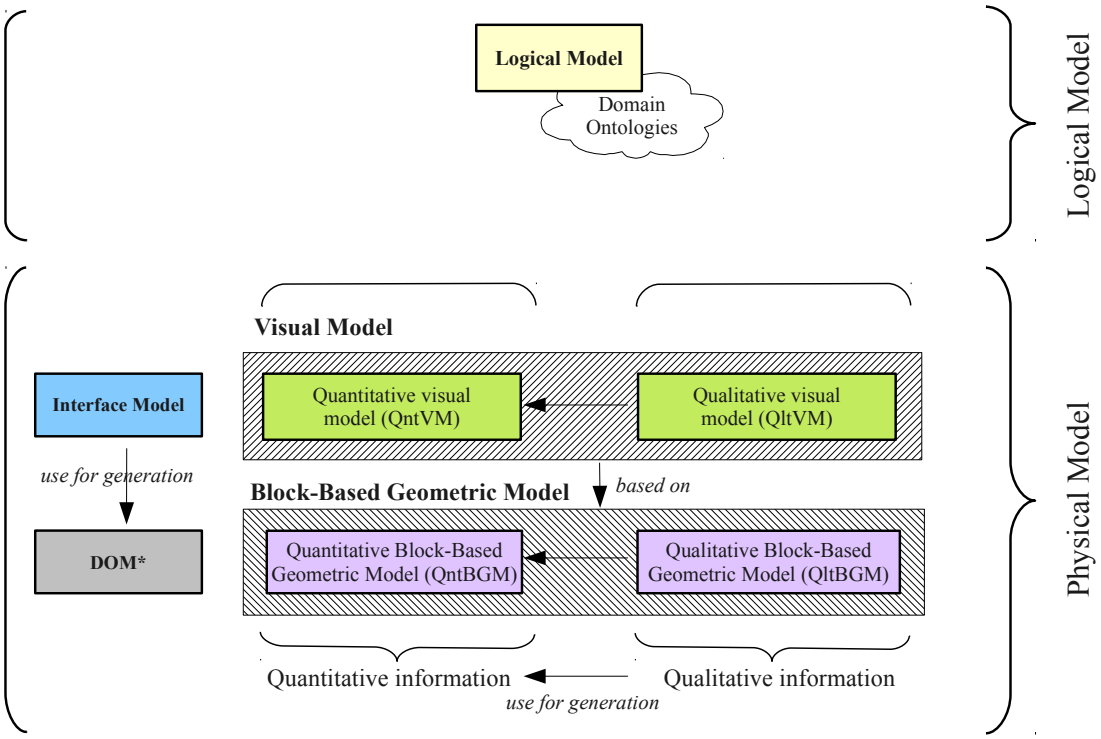


Figure 3.1: The Unified Ontological Model. (This figure is taken from [30])

Interface Model

The IM uses information which is derived from the DOM^*. This includes a list of the following elements:

a) web form elements (text fields, submit buttons, etc.)

b) web forms itself

c) links

d) images

e) elements which have a listener attached, mostly referred to it as interactive elements

f) structural elements like lists and tables as long as the can be derived from *display* (a computed CSS attribute).

Block-based Geometric Model

The BGM is model where web objects are represented in form of blocks or rectangles. The idea is to leverage the spatial features and relationships on a web page. The BGM itself is represented by a structural BGM (SBGM), a quantitative BGM (QntBGM) and a qualitative BGM (QltBGM).

- In the *SBGM* the structure of a web page, which was derived from the investigation of the DOM^*, is portrayed.
- The *QntBGM* provides the quantitative information of a block, distance, width and height in pixel as well as the direction in degrees.
- The *QltBGM* stores the representation about the layout with qualitative[3] values.

Topological relations were used to represent spatial relationships and special features. This concept bases on RCC8 algebra published in [18] as well as alignment, direction and distance. Since most web pages follow a Manhattan layout, direction relationships are represented by rectangular cardinal directions [73] for the TAMCROW project. Some relations like RCC8, alignment (except centering) and direction are revealed with a two-dimensional interval relation (2DIRs) (introduced in [8]) to be consistent with the UOM.

Visual Perception Model

A VPM is composed of a structural, qualitative and quantitative VPM. Furthermore, it contains perceptual features like saliency and the degree of highlighting. Based on the Gestalt concept published in [60], it is possible to group it.

3.2 Structural Elements

After some abstract concepts, the following section will deal with more concrete entities and concepts with which a reader may already have some experience.

Figure 3.2 illustrates the different types of structural elements which are defined in this section. With these entities and the UOM, defined in subsection 3.1, it is possible to compute and extract features of web objects. This technique is published by Fayzrakhmanov in [27].

As stated above, a web object is depicted by a rectangular area which covers a canvas of a web page. In general, we can speak of an Euclidean space when referring to the geometric space of a web page.

The next paragraph shows some definitions which are used in the feature description afterwards.

- A *document* is a web page rendered by the web browser. In general, this is compiled from a file (HTML, xhtml or XML).

[3] Sometimes they are referred to as categorical variables

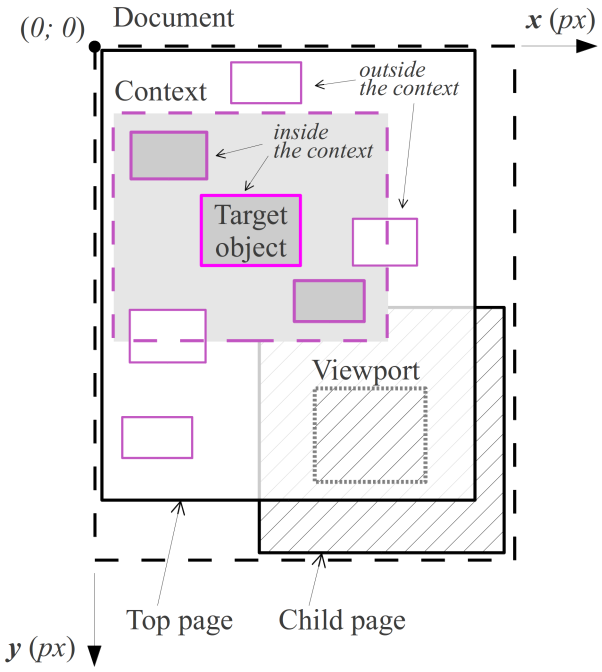


Figure 3.2: Structural objects of a web page. (This figure is taken from [30])

- A *page* is a subpart of a document. It is a DOM tree with the corresponding CSS attributes. The counterpart in the browser object model (BOM) is called window. In general, the origin of ordinates from a window (where $x = 0$ and $y = 0$) is on the top-left corner of the top level page.

- A *(selected) web object* is a minimum rectangle over one or more CSS boxes. In the TAM-CROW project the following types are considered *HtmlButton, HtmlCheckbox, HtmlFile-Upload, HtmlImage, HtmlPasswordInput, HtmlRadiobutton, HtmlSelect, HtmlText, Html-TextArea, HtmlTextInput*. These objects have it counterparts in the CSS model [3] as well as in the IM.

- The *context:* of a selected web object is the neighborhood of this object. It has the shape of a rectangle which spans beyond the object itself. Technically, the context region height is defined by twice the height (h) of the selected object and 1.4 the width (w) of the selected object. The minimum of each is 500 pixel plus width or height. The formal definitions are $h_{context} = max(500 + h_{object}, 2 * h_{object})$ and $w_{context} = max(500 + w_{object}, 1.4 * w_{object})$.

3.3 Features of the Web Objects

After some words about the concepts of the feature extraction, this section introduces the features itself.

The features can be assigned to the following disjoint groups:

- *Interface features:* These features cover web objects which play a functional role in respect to the interface design (e.g. button, image, text). In addition, structural objects like lists and tables are also included. In most cases, the feature calculation of this type is derived from the IM.
- *Spatial features:* Features of this category base on the BGM and include qualitative and quantitative features. Spatial features are *absolute position, size, number of aligned elements within an object's context, etc.*
- *Visual features:* This group focuses on features which reveal visual characteristics of the web objects. They are calculated with the help of the VPM and the IM. *Foreground and background color, emphasis, font size* and a few more are features of this group.
- *Textual features:* As the name suggested, these features are dealing with linguistic attributes. Therefore, they are mostly computed by best practices from natural language processing (see [91] for details). The IM and BGM provide the information needed for the calculations.

In the next four sub-subsections the list of the features and their formal computation is introduced. The groups are the same as the list above. The text written in **bold** indicates the name of the object, whereas the label inside the brackets $(label)$ indicates the technical name used inside the project. For reasons of simplification, only the technical names of the features are given in the illustrated graphs, figures and plots.

List of Interface Features

- **Object Type** (TypeIO): This is the type of the selected object.
- **Editable** (EditableIO): This features indicates if the web object can be modified without alteration of the source code.
- **Selection** (SelectedIO): Indicates if a radio button or checkbox is checked. Therefore, this feature is only valid for those types of web objects.
- **Dominant orthogonally visible object** (DominantOrthogonalVisibleTypeROC): In figure 3.3 it is illustrated what an orthogonally visible object is. However, if an object is completely hidden by another object and thus not visible it is not considered. From the remaining objects within the selected object's context, the object type which appears the most is taken.

- **Similar Types within the context** (SimilarTypesQntROC): This is merely the number of same object types within the object's context.
- **Link Type** (LinkTypeROTW): This value can be obtained by calculating the link type relatively to the URL of the top web page (also referred to as document).
- **Number of Objects** (TObjectsQntIC): This is the number of other objects within the context of the selected object.

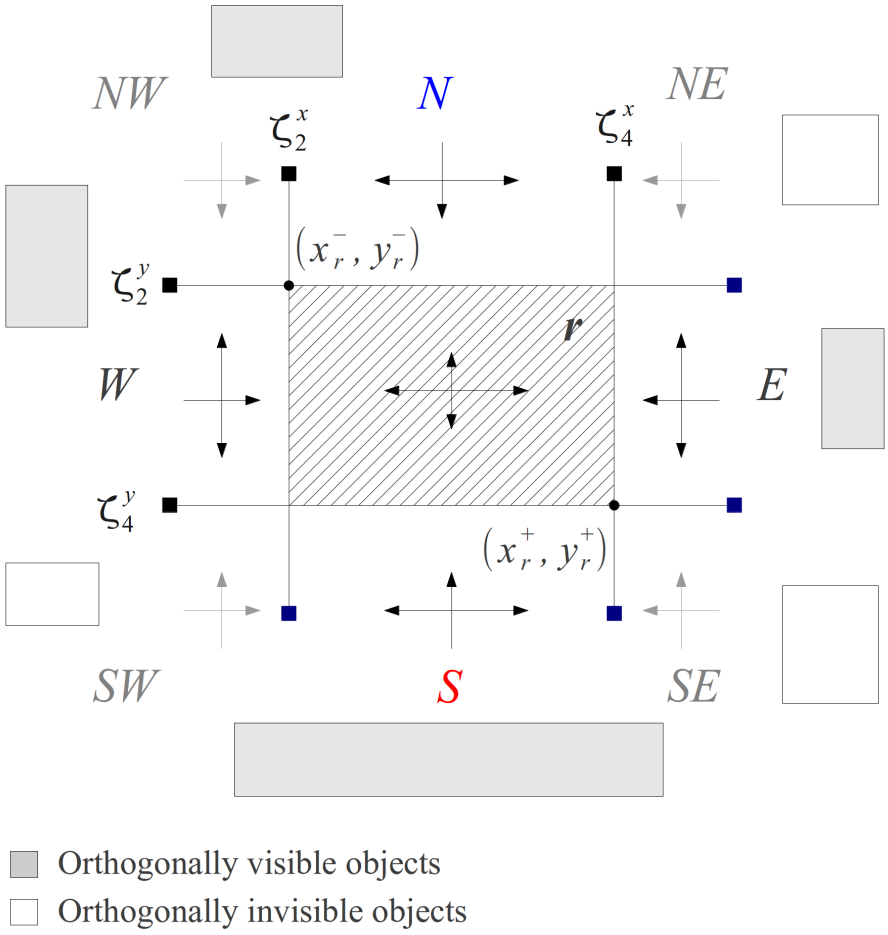


Figure 3.3: Web object's orthogonally visibility (This figure is taken from [30])

List of Spatial Features

- **Area** (AreaIO): This is the region of the minimum spanning rectangle over the selected object. It is computed by $width * height$.

- **Aspect Ratio** (AspectRatioIO): This is the ratio between width and height $\frac{width}{height}$. If the object is higher than wide, the value is below 1, otherwise it is greater than 1.
- **Number of horizontal Alignments** (AlignmentHorQntROC and AlignmentHorQntROD): This item represents two features, one is computed with respect to the context and the other one with respect to the document. In general, it is the number of objects within a horizontal alignment with the selected object.
- **Number of vertical Alignments** (AlignmentVertQntROC and AlignmentVertQntROD): Similar as above. These are also two features (for context and document). However, here these features hold the number of objects which share a vertical alignment relation.
- **Number of Alignments** (AlignmentQntROC and AlignmentQntROD): This number is simply the sum of the two above mentioned features (one for the context version and one for the document one).
- **Horizontal Alignment Index** (AlignmentIndexHorROC and AlignmentIndexHorROD): These features represents the index for the selected object within the same horizontal alignment. Like the three feature pairs before, this feature has two versions one regarding the context and one regarding the page.
- **Vertical Alignment Index** (AlignmentIndexVertROC and AlignmentIndexVertROD): They hold the vertical alignment index.
- **Ratio of vertical to horizontal Alignments** (AlignmentVertHorRatioROC and AlignmentVertHorRatioROD): This is merely the number of vertical alignment relations divided by the number of the horizontal ones. Once again, there are also two versions like above.
- **Alignment Factor** (AlignmentFactorROC): This is a ratio, which is computed by dividing the number of aligned objects within a context by the number of objects within the context.
- **Number of orthogonally visible objects** (OrthogonalVisibleObjQntROC): Figure 3.3 gives a graphical illustration about orthogonally visible objects. It is the number of those objects which are visible.
- **Number of orthogonally visible aligned objects** (AlignedOrthogonalVisibleObjQntROC): Same as above, with the additional condition that the orthogonally visible objects must have the same alignment.
- **Number of orthogonally visible fully aligned objects** (FullyAlignedOrthogonalVisibleObjQntROC): As above, but with the additional attribute of full alignment. This means that they have exactly the same alignment relations (vertical and horizontal) like the selected object has. The following relation holds: FullyAlignedOrthogonalVisibleObjQntROC $\leq$ AlignedOrthogonalVisibleObjQntROC $\leq$ OrthogonalVisibleObjQntROC.

- **Pixel to character ratio** (PixelsToCharacterIC): This is the ratio of the context area divided by the number of characters. It therefore holds the average area in square pixel per character for a certain context.
- **Relative Width** (RelativeWidthROW): It is the relative width of the selected object in respect to the web page's width.
- **Relative Height** (RelativeHeightROW): It is the relative height of the selected object in respect to the web page's height.
- **Relative x-position** (RelativeXPositionROW): This is the x position of the top left corner of the selected object relatively to the websites height. So the value lies between $[0, 1]$.
- **Relative y-position** (RelativeYPositionROW): Similar to above, where it represents the y position of the top left corner of the selected object with relatively to the websites width. Same as above, the value lies between $[0, 1]$.
- **3x3-Grid Locations** (GridLocationX3ROTW): This features represents the location of a web object on the web page divided into a 3-times-3-grid. The value represents an integer value resulting when coding the location on the grid as a 9-bit number.
- **Text Density** (TextSpatialDensityIC): This is the percentage of text in a selected object's context. For this computation, the area with text is divided by the area of the context.
- **Link Density** (LinkSpatialDensityIC): Similar to above, this features represents the percentage of links inside a selected object's context.

List of Visual Features

- **Foreground Color** (ForegroundColorIO): This is the integer value received by coding the color information in 32 bits (8 for each information namely red, green, blue and alpha). This information is only valid for objects which contain information in textual form.
- **Background Color** (BackgroundColorIO): Same color information as above. However, this value is valid for all objects with no multimedia item as background (i.e. for a web object having a picture as background, this value would be *null*).
- **Emphasis** (EmphasisIO): With this feature we try to quantify the level of emphasis of a textual element. The higher the value the stronger the emphasis of the text.
- **Font Size** (FontSizeIO): The font size of an object. This feature is not valid for images, radio buttons and check boxes.
- **Avg. weighted Foreground Color Distance** (AvgWeightedFGColorROC): This is the sum of the weighted foreground color distance (according to the HSV distance). The distances are computed between the selected object and all objects within its context.
- **Avg. weighted Background Color Distance** (AvgWeightedBGColorROC): Same as above for the background color, if it exists.

List of Textual Features

- **Text of Object** (TextIO): This is the text of an object. This feature is only valid for object which can contain textual information.
- **Number of Lines** (LinesQntIO): The number of lines are counted in the minimum spanning rectangle. This information depends on the rendering of the web page.
- **Number of Tokens** (TokensQntIO): This holds the number of tokens of a web object. Mostly, they are words.
- **Text above** (UpperTxtOfOrthVisibleObjsROC): This is the text of the orthogonally visible objects above. Figure 3.3 illustrates orthogonally visible objects
- **Text right** (RightTxtOfOrthVisibleObjsROC): This is the text of the orthogonally visible objects to the right. (Figure 3.3)
- **Text below** (BottomTxtOfOrthVisibleObjsROC): This is the text of the orthogonally visible objects below. (Figure 3.3)
- **Text left** (LeftTxtOfOrthVisibleObjsROC): This is the text of the orthogonally visible objects to the left. (Figure 3.3)
- **Text of the nearest orthogonally visible object** (TextOfNearestOrthVisibleObjsROC): This is the text of the nearest orthogonally visible object, nearest in the sense of the Euclidean distance. (Figure 3.3).
- **Text of the nearest object** (TextOfNearestTxtObjROC): The text of the nearest object regardless of its visibility or its orthogonality to the selected object.
- **Character Density of Links** (LinkCharacterDensityIC): This feature is a ratio of the characters in a link divided by all characters in an object's context. Therefore, the value lies between $[0, 1]$

CHAPTER 4

Computation of Distance Vectors

In this chapter the technical details of the distance computation are illustrated. This chapter completes section 2.2 in chapter 2. In contrast to its complementary section, which aims to demonstrate the procedure and its steps as such, the target of this chapter is to define the computation of the distances technically and formally.

It is scarcely necessary to point out that the distance computation is a crucial method to enhance the computability of the visual perceivable features of the web objects. One example is the integer representation of the RGBA color code. Without the distance computation it would be not possible to compare different colors with each other. Another example is the integer representation of the 3x3-Grid feature (GridLocationX3ROTW). Moreover, the number of observations can be significantly increased with the distance computation. As a consequence, this fosters the training of the classifiers.

This chapter is structured as following: Firstly, the technical definitions how the distance are calculated is introduced. Secondly, the mapping between the visual perceivable features and the distances are illustrated. Finally, the computation of the class attribute for each web object pair is introduced.

4.1 Technical Definitions for the Distance Calculation

This section provides a brief overview of the formulae used for computing the distance of every feature. For each feature exactly one distance is calculated. As a consequence, the dimension of the data is not changed.

- *Relative Distance:* The relative distance is used to compare two numerical absolute values of features between two web objects. This distance always lies between 0 and 1 inclusive. In case that a web object has no value for a certain feature (also known as *NULL*-value), the distance of 1 is assigned which indicates the biggest possible difference. Formula (4.1) shows how the relative distance is computed. The name might seem irritating since the

value is calculated from absolute values. However, the distance maps the absolute values to a relative one.

$$\delta_{rel} = \frac{1}{1 + e^{-max(f_1,f_2)}} - \frac{1}{1 + e^{-min(f_1,f_2)}} \tag{4.1}$$

- *Absolute Distance:* This distance is used for features with a percentage value. This means that the features have already a value between 0.0 and 1.0. The distance is defined in formula (4.2) where f_1 and f_2 hold the value of the desired relative feature from web object 1 and 2. The name of this distance measure might seem confusing, since the values are from relative origin. However, the calculation is absolute and therefore the name.

$$\delta_{absolute} = |f_1 - f_2| \tag{4.2}$$

- *Nominal Distance:* This distance is used for features of nominal or categorical nature. In case that the two features have the same value the distance is 0.0 indicating equality, whereas in any other case the distance is 1.0 indicates inequality. Formula (4.3) shows this formally, where f_1 and f_2 hold the values of the certain feature of web object 1 respectively 2.

$$\delta_{nominal} = \begin{cases} 1 & \text{if } f_1 = f_2 \\ 0 & \text{if } f_1 \neq f_2 \end{cases} \tag{4.3}$$

- *Boolean or Dichotomy Distance:* The dichotomy distance is a special case of the above mentioned distance, because it deals with values which take only the values *true* or *false*. The computation follows equation 4.3.

- *String Edit Distance:* When comparing textual features, the string edit distance is calculated. This measurement is also known as the Damerau–Levenshtein distance. Its definition can be found in [22].

- *Color Distances:* For features which code a color value (RGBA[1] or ARGB[2]) it is necessary to compute a distance measure in order to facilitate further computations. Otherwise, these features would have to be treated as categorical features which is not helpful for machine learning algorithms, since it is difficult to measure their similarity (in case of comparing the color code). For computing the color distance a HSV color space is used. The values in this case lie between 0 and 2 the first indicating equality and the latter inequality. The advantage of the HSV color space is that it closely resembles human perception.

- *Grid Overlapping Distance:* One feature indicates which areas after dividing a web page into a equally proportioned 3-times-3-grid of an web object are touched. The distance of two web objects can be calculated by counting the intersecting areas and dividing them by the sum of areas resulting by the union operator for sets on the two web objects (details on set theory can be found at [20]). Equation (4.4) shows the formal definition where A is a set which contains the IDs of the covered areas of web object α. B is the counterpart for

[1] Red, Green, Blue and Alpha
[2] Alpha, Red, Green and Blue

web object β. With figure 4.1 the following examples illustrate the computation of this distance measure.

$$\delta_{grid} = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{4.4}$$

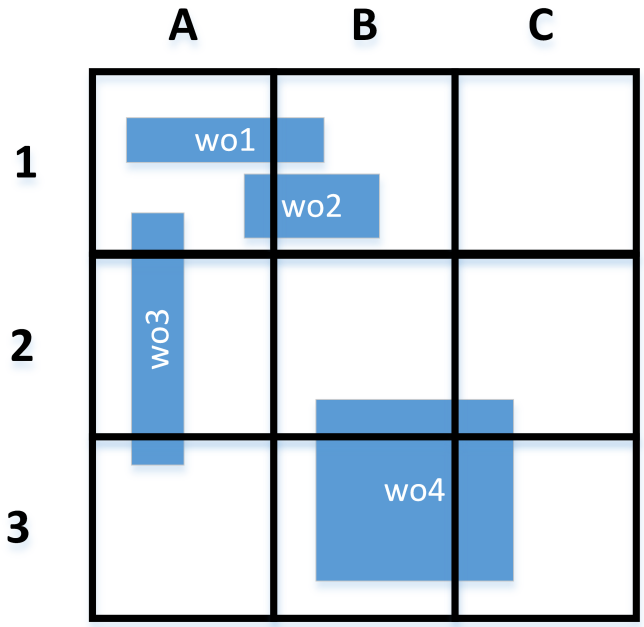


Figure 4.1: Example for the Grid Overlapping Distance. The raster symbolize the segmented web page. woX stands for the different web object on the web page.

$$\delta_{grid}(wo1, wo4) = 1 - \frac{|\{A1, B1\} \cap \{B2, C2, B3, C3\}|}{|\{A1, B1\} \cup \{B2, C2, B3, C3\}|} = 1 - \frac{0}{6} = 1$$

$$\delta_g(wo1, wo3) = 1 - \frac{`|\{A1, B1\} \cap \{A1, A2, A3\}|}{|\{A1, B1\} \cup \{A1, A2, A3\}|} = 1 - \frac{1}{4} = 0.75$$

$$\delta_g(wo1, wo2) = 1 - \frac{|\{A1, B1\} \cap \{A1, B1\}|}{|\{A1, B1\} \cup \{A1, B1\}|} = 1 - \frac{2}{2} = 0.0$$

4.2 Mapping from the Features to the Distances

This section introduces the mapping of the features to the different distance types. The following tables 4.1, 4.2, 4.3 and 4.4 gives an overview of the mapped features grouped in the same way as the features before.

Table 4.1: Interface Feature Distance Measurement

Name	Distance Computation
Object Type	nominal
Editable	dichotomy
Selection	dichotomy
Dominant orthogonal visible object	nominal
Similar Type within the context	relative
Link Type	nominal
Number of Objects	relative

Table 4.2: Spatial Feature Distance Measurement

Name	Distance Computation
Area	relative
Aspect Ratio	relative
Number of Alignments (both)	relative
Number of horizontal Alignments (both)	relative
Number of vertical Alignments (both)	relative
Horizontal Alignment Index (both)	absolute
Vertical Alignment Index (both)	absolute
Alignment Factor	relative
Ratio of vertical to horizontal Alignments (both)	relative
Number of orthogonal visible objects	relative
Number of orthogonal visible aligned objects	relative
Number of orthogonally visible full aligned objects	relative
Pixel to character ratio	relative
Relative Width	absolute
Relative Height	absolute
Relative x-position	absolute
Relative y-position	absolute
3x3-Grid Locations	grid overlapping
Text Density	absolute
Link Density	absolute

Table 4.3: Visual Feature Distance Measurement

Name	Distance Computation
Foreground Color	color
Background Color	color
Emphasis	relative
Font Size	relative
Avg. weighted Foreground Color Distance	absolute
Avg. weighted Background Color Distance	absolute

Table 4.4: Textual Feature Distance Measurement

Name	Distance Computation
Text of Object	string edit
Number of Lines	relative
Number of Tokens	relative
Text Above	string edit
Text Right	string edit
Text Below	string edit
Text Left	string edit
Text of the nearest orthogonal visible object	string edit
Text of the nearest object	string edit
Character Density of Links	absolute

4.3 The Computation of the Class Attribute

Let us recall some definitions from chapter 2 and subsection 2.2, namely T_t and T_a:

$T_t = \{DepDate, DepLocation, ArrLocation, Adults, OneWay, Submit, Other\}$

$T_a = \{ArrDate, DepDate/\#Nights, ArrLocation, Adults, Submit, Other\}$

With these definitions it is possible to formally define the class column which is added to each distance matrix. Formally, a set S is defined as following $\{T \backslash other\}$ where $T \in \{T_a, T_t\}$. That class value depends on the types of the two web objects which a row corresponds to. For every resulting distance matrix, the class column is different. It can be said that each distance matrix corresponds to an element of set S. When calculating a specific distance matrix for one selected element of the set S, the types of the others are treated as they were of the type *other* and as a consequence not functional for this task. When considering this usage, the following formula (4.5) can be used to assign the class value, where i and j indicate the ID of a web object and $type(x)$ is a function which returns the type of web object x.

$$class_{i,j} = \begin{cases} 1 & \text{if } \; type(wo_i) = type(wo_j) \wedge type(wo_i) \neq other \\ 0 & \text{else} \end{cases} \tag{4.5}$$

CHAPTER 5

Used Classification Techniques

5.1 Introduction

The following chapter discusses different classification methods[1]. All of these algorithms were used in the TAMCROW project to classify functional[2] *web objects* on web pages.

This chapter is structured as following: Firstly, it gives a short introduction of the dataset which is used to illustrate the different classification methods. Secondly, the logistic regression is introduced as a possible technique for classification. Thirdly, a popular representative of classification trees, namely Quinlan's c4.5, is presented and further discussed. Fourthly, a k Nearest-Neighbors (kNN) method is shown. This subsection provides an insight to the technique compared with others in this chapter. Finally, the support vector machines (SVM) are introduced. In this subsection different kernel functions are discussed and the process of achieving optimal parameter settings is explained.

5.2 Dataset for Illustrations and Examples

This section introduces the data set which was used to illustrate the methods of the different classification techniques as well as to give some concise and lucid examples.

Unfortunately, the data sets from the *TRAMCROW project* are not well-suited to meet the above claimed attributes. The data sets are rather high dimensional therefore scatterplots representing this data will be crowded. As a result they would be rather confusing than descriptive. Therefore, in the next paragraphs of this section, the selected data set for illustrating purpose is introduced.

For demonstration, the well-known *Fisher's and/or Anderson's* iris[3] data set was used. This data set contains several features for three species of iris namely setosa, versicolor and virginica.

[1] In this chapter classification methods, classification algorithms and classification techniques are used as synonymous.

[2] Functional in this context means that these web objects fulfill a special function within its scenario.

[3] The data set contains information about flowers and not human irises

The collected features are sepal length, sepal width, petal length and petal width in centimeters as well as the species as string. A 4x4 scatterplots in figure 5.1 shows the unscaled data, where the species feature has been excluded, because the color and symbol type already reflect this information. The data set was published in [6] and [33]. The latter provides, besides the data, also some discriminant analysis.

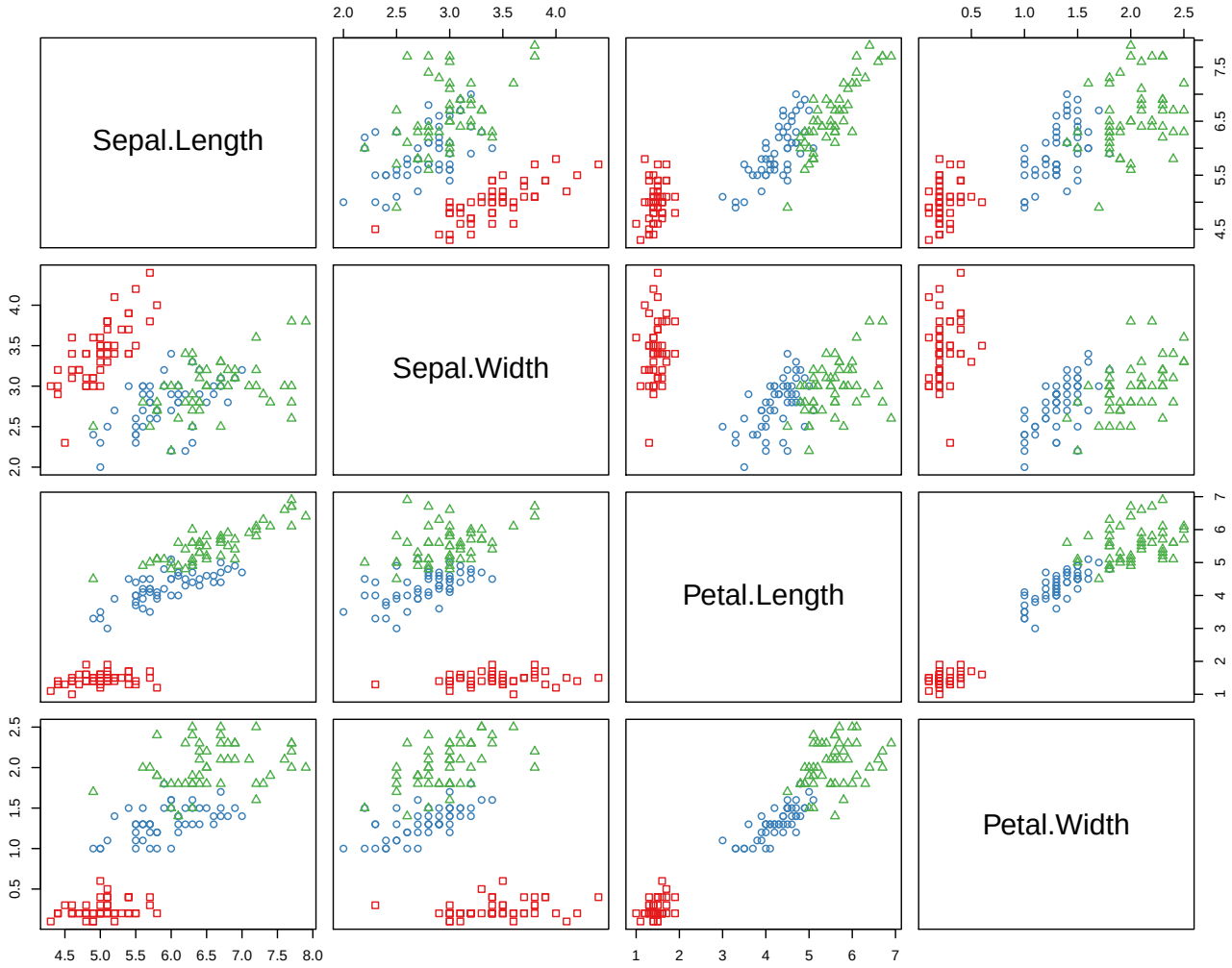


Figure 5.1: scatterplot of the iris data; red squares represent the setosa species, blue circles illustrate the versicolor species and the green triangles stand for the virginica species

What can be easily observed in figure 5.1 is that the red squares (setosa species) are more different from the other two than the other two from each other in means of the sepal length and

width as well as the petal length and width. Therefore, it appears that a classification problem would be easier to manage between setosa and versicolor or virginica than between versicolor and virginica. The entire data set is provided in the appendix at chapter B in section B.1.

5.3 Logistic Regression

This section provides a concise introduction to logistic regressions. If the reader is looking for a comprehensive literature about this topic, he or she might take a look at Agresti [5].

Formal Model Definition

The name of the logistic regression derived from its model definition. In order to understand this, the ordinary least square regression (OLSR) is briefly introduced (see equation 5.1). The vector y is the variable which should be described by the input matrix X. The main assumptions of the model is that y is more or less linearly described by X. As a result, the variance of ε (σ^2) is homoscedastic, the expected value of ε is 0, ε is normally distributed ($\varepsilon \sim N(0, \sigma^2)$. The main task is to find the coefficients for α and the β vector such that the sum of the squared error ($\sum_{i=1}^{N} \varepsilon^2$ or in matrix notation $\varepsilon^\top \varepsilon$) is minimized.

$$y = \alpha + \beta^\top X + \varepsilon \tag{5.1}$$

In the model of the logistic regression the log ratio between the class belonging probabilities are described by a linear regression function. In order to keep things simply this subsection deals with a two class scenario (see equation (5.2)). $P(G = n|x)$ stands for the probability that an observation x is in class n. In case the reader is interested in the N class case, he or she might take a look at subsection 5.3.

$$log\left(\frac{P(G=1|x)}{P(G=2|x)}\right) = \alpha + \beta^\top x \tag{5.2}$$

The probabilities of the class belongings can then be easily calculated as shown in equation (5.3). As a fact all probabilities of belonging to a certain class have to sum up to 1. This means

that in the two class case $P(G = 1|x) = 1 - P(G = 2|x)$.

$$
\begin{aligned}
log\left(\frac{P(G=1|x)}{P(G=2|x)}\right) &= \alpha + \beta^\top x \;\Big|\; e^y \\
\frac{P(G=1|x)}{P(G=2|x)} &= e^{\alpha+\beta^\top x} \;\Big|\; P(G=2|x) = (1 - P(G=1|x)) \\
P(G=1|x) &= (1 - P(G=1|x))e^{\alpha+\beta^\top x} \\
P(G=1|x) &= e^{\alpha+\beta^\top x} - P(G=1|x)e^{\alpha+\beta^\top x} \qquad (5.3) \\
P(G=1|x) + P(G=1|x)e^{\alpha+\beta^\top x} &= e^{\alpha+\beta^\top x} \\
\left(1 + e^{\alpha+\beta^\top x}\right) P(G=1|x) &= e^{\alpha+\beta^\top x} \\
P(G=1|x) &= \frac{e^{\alpha+\beta^\top x}}{1 + e^{\alpha+\beta^\top x}} = \frac{1}{1 + e^{-(\alpha+\beta^\top x)}}
\end{aligned}
$$

Fitting the Parameters

After deriving the formula for the probability that x belongs to class 1, it is obligatory to estimate α and the vector β. The parameter estimation follows the maximum likelihood principle (for details regarding the maximum likelihood principle see [41]). In order to build a log likelihood function with just one parameter we redefine β as shown in equation (5.4). The resulting log likelihood with respect to $\boldsymbol{\beta}$ is shown in equation (5.5) where y_i is an indicator variable depending on the class (see 5.6).

$$
\boldsymbol{\beta} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} \qquad (5.4)
$$

$$
l\left(\boldsymbol{\beta}\right) = \sum_{i=1}^{m} \left(y_i \boldsymbol{\beta}^\top x_i - log\left(1 + e^{\boldsymbol{\beta}^\top x_i}\right)\right) \qquad (5.5)
$$

$$
y_i = \begin{cases} 1 & \text{if } G = 1 \\ 0 & \text{if } G = 2. \end{cases} \qquad (5.6)
$$

In order to maximize the function, $l(\boldsymbol{\beta})$ has to be derived by beta and set to 0, shown in equation (5.8). Be aware that the x_i contains 1 in order to accommodate the intercept (α) which has been moved to $\boldsymbol{\beta}$ (see equation 5.7). In addition, $p(x_i; \boldsymbol{\beta})$ corresponds to $P(G = 1|x)$ and $1 - p(x_i; \boldsymbol{\beta})$ corresponds to $P(G = 2|x)$. The difference of P and p is that the first is the real

probability, while the latter is an estimator function for a certain observation x_i and a vector of the coefficients $\boldsymbol{\beta}$ which are used to model the function $\frac{1}{1+e^{-(\boldsymbol{\beta}^\top x)}}$.

$$x_i = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix} \tag{5.7}$$

$$\frac{\partial l\left(\boldsymbol{\beta}\right)}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{m} \left(y_i - p\left(x_i; \boldsymbol{\beta}\right)\right) x_i = 0 \tag{5.8}$$

The solution of the resulting system of equations can achieved by the Newton-Raphson algorithm resulting in equation 5.9 (explanation regarding this method can be found in [4]).

$$\boldsymbol{\beta}_{new} = \boldsymbol{\beta}_{old} - \left(\frac{\partial^2 l\left(\boldsymbol{\beta}\right)}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top}\right)^{-1} \frac{\partial l(\boldsymbol{\beta})}{\partial \beta} \tag{5.9}$$

In matrix notation the model is easier to understand, as shown in equation (5.11). The $\boldsymbol{y}$ is a $n \times 1$ vector, $\boldsymbol{X}$ is a $n \times (p+1)$ matrix holding the observations, $\boldsymbol{p}$ is an $n \times 1$ vector of the estimated probabilities $p(x_i; \boldsymbol{\beta})$ and $\boldsymbol{W}$ is a $(n \times n)$ diagonal matrix with the weights $P(G = n|x, \boldsymbol{\beta_{old}})$. With these definitions it is possible to derive the following substitutions shown in equation (5.10).

$$\begin{aligned} \frac{\partial l(\boldsymbol{\beta}}{\partial \beta}) &= \boldsymbol{X}^\top \left(\boldsymbol{y} - \boldsymbol{p}\right) \\ \frac{\partial^2 l\left(\boldsymbol{\beta}\right)}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} &= -\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X} \end{aligned} \tag{5.10}$$

$$\boldsymbol{\beta}_{new} = \underbrace{\left(\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{W}}_{\text{weighted least square}} \overbrace{\boldsymbol{X}\boldsymbol{\beta}_{old} + \underbrace{\boldsymbol{W}^{-1}\left(\boldsymbol{y} - \boldsymbol{p}\right)}_{\text{adjustment}}}^{\boldsymbol{z}} \tag{5.11}$$

An other name for the above mention algorithm is iteratively reweighted least squares (IRLS), since for every iteration it solves the following problem:

$$\boldsymbol{\beta}_{new} \leftarrow \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\boldsymbol{z} - \boldsymbol{X}\boldsymbol{\beta}\right)^\top \boldsymbol{W} \left(\boldsymbol{z} - \boldsymbol{X}\boldsymbol{\beta}\right) \tag{5.12}$$

A good start point for $\boldsymbol{\beta} = 0$, however convergence is not guaranteed (see [66] for some problems with non-convergence situations). In case of more than two classes, the Newton-Raphson algorithm might be formulated as an IRLS algorithm where $\boldsymbol{W}$ is no diagonal matrix anymore.

Example with the Iris Data Set

Figure 5.2 shows the a logistic regression example for the petal length and classes of the Iris data set. The left graph illustrates the classes setosa and versicolor. In this example, the classes are separable by a hyperplane. In contrast, in the right graph the classes versicolor and virginica are not separable with an hyperplane since the classes overlap.

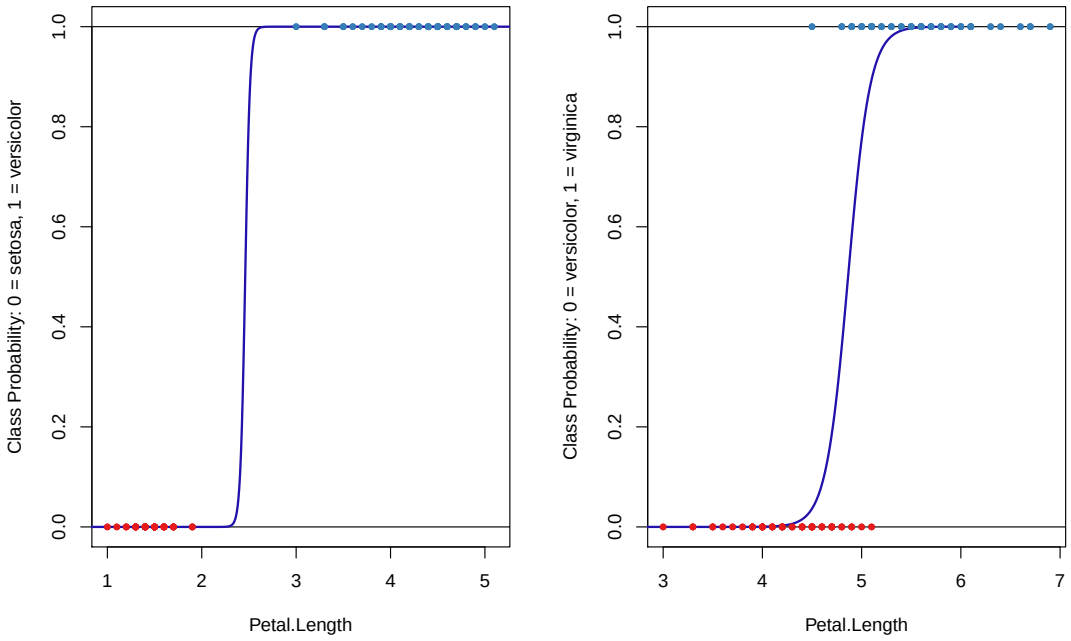


Figure 5.2: left: An example with separating classes (setosa = 0, versicolor = 1); right: An example with not separating classes (versicolor = 0, virginica = 1). Both in respect to the length of the petal.

Logistic Regression Model for N Classes

In the n-class model, an arbitrary class has to be chosen as divisor. In the following model class N was used.

$$\begin{aligned} log\left(\frac{P(G=1|x)}{P(G=N|x)}\right) &= \alpha_1 + \beta_1^\top x \\ log\left(\frac{P(G=2|x)}{P(G=N|x)}\right) &= \alpha_2 + \beta_2^\top x \\ &\vdots \\ log\left(\frac{P(G=N-1|x)}{P(G=N|x)}\right) &= \alpha_{N-1} + \beta_{N-1}^\top x \end{aligned} \tag{5.13}$$

$$
\begin{aligned}
P(G = n|x) &= \frac{e^{\alpha_n + \beta_n^\top x}}{1 + \sum_{i=1}^{N-1} e^{\alpha_i + \beta_i^\top x}} \quad \text{for } 1 \leq n \leq (N-1) \\
P(G = N|x) &= \frac{1}{1 + \sum_{i=1}^{N-1} e^{\alpha_i + \beta_i^\top x}} \qquad\qquad \text{for } n = N
\end{aligned}
\tag{5.14}
$$

Variable Importance

With the logistic regression it is possible to measure the variable importance[4]. Somebody might argues that this could also be done by a decision tree, because the nearer a variable node to the root, the more important this variable is. However, it is not possible to quantify this (e.g. what does it mean if a variable needs more splits and has therefore more nodes? How much importance should a node gain, because it is higher ranked than another?).

In general, the ratio of the explaining variance to the total variance is measured and compared with the case that one variable coefficients is set to 0 (or omitted). If there is a significant drop in explaining power, this variable is considered as significant. An appropriate method is the $\Delta_l \chi^2$ value which the equivalent to $\Delta_l RSS$ in the least squares regression. Details to the computation and its robustness can be found in Pregibon [78].

The inference statistics are very helpful to understand its data set better. However, there are some pitfalls. For example, the significance of one variable or feature is given by the explanatory power of this one variable. When there are interdependences between two variables it might happen that after a non-significant variable was excluded from the model, a variable, which was significant at first becomes non-significant afterwards. Since this information was not used for variable selection in order to build a small logistic regression model, this is negligible.

Advantages

A great advantage of the logistic regression is that it provides the user with an inference statistic. With this information it is possible to build a reduced model for classification. However, since it is technically not an issue to collect the already programmed features, this information is not used to build a simpler model for classification in TAMCROW.

In addition, it is a classical statistcal method that can be used as benchmark for more sophisticated techniques. This might help to address the question if it is worth to spend more time and effort to fit more sophisticated models.

Furthermore, it is easy to use (almost every statistcal software provides this feature), fast with high-dimensional data and does not need many parameter estimations (in general none). However, it seems that his method is widely used within computer scientist.

Disadvantages

It could happened that for highly unbalanced classes the Newton-Raphson algorithm did not convergence, which might be shocking for an ordinary user at the first time. Sometimes, convergence can be achieved, by increasing the maximum number of iterations in the used software

[4] Statisticians would rather say variable importance, while computer scientists might prefer feature importance.

package. If that still does not work, it could be that the problem is some kind of *special*. However, even then, it seems that the prediction is still not a problem, but the coefficients would not be stable for several runs.

Literature and References

Most formulae have been taken from Hastie et al [43]. However, sometimes the author of the master thesis has changed the variable names to be coherent with previous or subsequent notations. In case that no convergence is reached with a used data set, it might be useful to take a look at [66].

5.4 Tree Based Classifiers

In general, there are different types of tree based methods. One is used for regressions and another one for classification problems. The main purpose of this master thesis is to classify web objects and therefore a classification tree method is of interest. This section is based on the well known c4.5 classification tree from Quinlan published in [81]. His solution is still in use and a common technique for classification applications.

Table 5.1: A training set for decision trees. This example is taken from [81]

Outlook	Temp F	Humidity %	Windy?	Class
sunny	75	70	true	Play
sunny	80	90	true	Do not Play
sunny	85	85	false	Do not Play
sunny	72	95	false	Do not Play
sunny	69	70	false	Play
overcast	72	90	true	Play
overcast	83	78	false	Play
overcast	64	65	true	Play
overcast	81	75	false	Play
rain	71	80	true	Do not Play
rain	65	70	true	Do not Play
rain	75	80	false	Play
rain	68	80	false	Play
rain	70	96	false	Play

Construction of a Decision Tree

In [79] Quinlan introduced the id3[5] algorithm and proposed the following basic characteristics of constructing a decision tree. This method originally came from Hunt [52] and is considered as an

[5] A predecessor of the c4.5 classification tree

elegant approach. The input data for the construction is a set of observations T[6]. Each observation belongs to a class, where the set of all classes are referred to as C and $C = \{C_1, C_2, ..., C_k\}$. Hunt's strategy consists of the three parts which are enumerated below. However, before starting with this, it is important to mention that the set of observations T hold only those observations which are valid for the current node of the decision tree.

- When all observations of T are part of one class C_i, then this is a leaf of a decision tree and identified for this class.
- No observations are contained in T. Special treatment is required here. The c4.5 uses the class with the highest probability in the upper or parent node.
- When more than one class are contained in T, a new node is inserted into the decision tree. This node is based on one feature of T, which splits the data according to a certain criteria. This can then be applied recursively until there are no more rules to apply. In Quinlan's [81], he writes about splitting T into several parts (according to the number of splits) and passing on only the applied observations T_i to each branch.

In the next subsection measures for inserting nodes into a decision tree are introduced.

Entropy

This subsection as well as the subsection on decision trees are based on the work of Quinlan [81] and therefore the following equations are composed by Quinlan and not by the author of this master thesis. This holds also for the shown example computations. Furthermore, the table 5.1 is a common example in books about Business Intelligence (BI) and Data Mining (DM). Due to its popularity this example is also used here.

The id3 algorithm (see [79]) used the gain (see equation (5.17)) criteria. It is based on the fact that information transfer can be measured in bits. The mathematical definition can be found in equation (5.15). $P(C_j)$ holds the relative frequency that observations belong to class C_j, therefore the interval is $[0, 1]$. E.g. if there are only two messages possible with the same probability, then $info(S) = -log_2\left(\frac{1}{2}\right)$ or 1 bit. This is also called entropy for the set S.

$$info(S) = -\sum_{j=1}^{k}(P(C_j)) \times log_2(P(C_j)) \tag{5.15}$$

The average of all sub sets of observations is then the expected information requirement. Its mathematical definition can be found in equation (5.16).

$$info_x(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times info(T_i) \tag{5.16}$$

[6]Since Quinlan refers to it as the test cases, the variable T is used in his work. For coherence reasons the author of this work do not change that.

The difference between the proposed splits of a decision tree $info_x(T)$ and a current state is the $gain(X)$. What might be interesting in Quinlan's notation is that the parameter of the $gain$ is X while the parameter used in the the difference is both time T.

$$gain(X) = info(T) - info_x(T) \tag{5.17}$$

The idea then is to find a T in such a way that the $gain$ is maximized.

Example based on Table 5.1

This sub-subsection illustrates the usage of the formula above with the example in the table at the beginning of this section. The following calculations are taken from [81], since the example seems to be very illustrative, the author of the master thesis believes that it would fit well for illustrating classification trees.

When recalling table 5.1, the reader will see that the first column from the right consists of two classes. One indicates that the weather is suited for playing and another which gives the information that the weather is not suitable (so *play* or *do not play*). There are 9 cases of play and 5 of do not play. Equation (5.18) illustrates the result when putting that values into equation (5.15).

$$info(T) = -\frac{9}{14} \times log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.94 \text{ bits} \tag{5.18}$$

Let us assume, as first node we want to introduce a separation by the variable *outlook*. There are three cases sunny, overcast and rain within this variable. Calculation (5.19) shows the result by putting in the above mentioned node into (5.16). The first row in (5.19) represents the case when outlook = sunny (5 out of 14 cases) with 2 plays and 3 do not play. The second row of (5.19) illustrates the case when the outlook = overcast (4 out of 14 times) where all classes are from type play. In the third row, the case where outlook = rain is illustrated. This would results in 2 play and 3 do not play classifications.

$$\begin{aligned} info_x(T) =& \frac{5}{14} \times \left(-\frac{2}{5} \times log_2\left(\frac{2}{5}\right) - \frac{3}{5} \times log_2\left(\frac{3}{5}\right)\right) \\ &+ \frac{4}{14} \times \left(-\frac{4}{4} \times log_2\left(\frac{4}{4}\right) - 0\right) \\ &+ \frac{5}{14} \times \left(-\frac{3}{5} \times log_2\left(\frac{3}{5}\right) - \frac{2}{5} \times log_2\left(\frac{2}{5}\right)\right) \\ &= 0.694 \text{ bits} \end{aligned} \tag{5.19}$$

The resulting gain would be $0.94 - 0.694 = 0.246$ bits. Now let us consider the case when we just use the windy variable. The case windy = true (6 out of 14 cases) results in 3 times play

and 3 times do not play. In contrast, the case windy = false (8 out of 14 cases) results in 6 times play and 2 times do not play.

$$\begin{aligned} info_x(T) =& \frac{6}{14} \times (-\frac{3}{6} \times log_2(\frac{3}{6}) - \frac{3}{6} \times log_2(\frac{3}{6}) \\ &+ \frac{8}{14} \times (-\frac{6}{8} \times log_2(\frac{6}{8}) - \frac{2}{8} \times log_2(\frac{2}{8}) \\ &= 0.892 \text{ bits} \end{aligned} \tag{5.20}$$

The results for this split would be a gain of $0.94 - 0.892 = 0.048$ bits. The more favorable split is with variable outlook, since its gain is much higher than when choosing variable windy.

This above example illustrates the split criteria when using the id3 algorithm of Quinlan (from [79]). Based on the gain criteria, Quinlan used an enhanced version which is described in the next paragraphs and equations. Quinlan comes to the conclusion that decision based only on the gain criteria are strongly biased for tests resulting in a vast number of outcomes. In [81] he suggested to use the $gain\ ratio$ (see equation (5.22)) instead as it corrects the bias with a normalization. The $gain\ ratio$ uses the $split\ info$ which is defined below in equation (5.21).

$$split\ info(T) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times log_2(\frac{|T_i|}{|T|}) \tag{5.21}$$

$$gain\ ratio(X) = \frac{gain(X)}{split\ info(X)} \tag{5.22}$$

For the example above this results in an $split\ info$ of 1.577 (see computation (5.23)). As a result, the $gain\ ratio = \frac{0.246}{1.577} = 0.156$.

$$split\ info(T) = -\frac{5}{14} \times log_2(\frac{5}{14}) - \frac{4}{14} \times log_2(\frac{4}{14}) - \frac{5}{14} \times log_2(\frac{5}{14}) = 1.577 \tag{5.23}$$

Quinlan claims that the $gain\ ratio$ is a robust measure. Mingers [70] has the opinion that this measure would results in smaller trees. By all means, this could only be an advantage, since the pruning procedure becomes less time consuming and overfitting becomes less of a problem.

Pruning

Pruning of a decision tree means to simplify a tree by removing and merging nodes. An advantage for an explanatory purpose is that the pruned tree will be smaller and therefore simpler. In general, simple explanations are preferred over complex ones. For applications in the field of prediction, a pruned tree has the advantage that its overfitting tendency is greatly reduced.

Actually there a several kinds of pruning methods. The simplest one is to prevent that T is divided into smaller than z observations. This prevention will directly work during the building phase. An advantage is that no extra pruning phase is needed, therefore such methods are also considered as *pre-pruning*. However, Quinlan pointed out that this is a rather unreliable technique, since the optimal value of z is difficult to find and often seems kind of arbitrary.

Therefore, Quinlan did not implement this into the c4.5 algorithm and decided to use an error-based pruning approach. Let us assume, it is possible to predict the error rate of a tree and any subtree. Then a certain branch of a node (and with it a subtree) would be removed by a leaf if the error rate of the node where the deleted branch belongs could be decreased. This procedure needs to be applied recursively. However, it is not possible to predict this error rate with all certainty. Therefore, it is obligatory to find a proper heuristic to predict the error rate. Quinlan proposed two types, one named *cost-complexity* pruning from Breiman et al. [12] and his version called *reduced-error* pruning [80]. Quinlan decided to use an improved version of his technique in the c4.5, therefore the following subsection is based on the reduced-error pruning.

His version does not need a segmentation of the observations into test and training data set, compared with the version from Breiman et al. version [12]. The main idea of Quinlan's error pruning is to replace nodes with leaves. The non-trivial part is that, which criteria are to be used in order to decide that the current node should be replaced with a leaf. His idea is to use the sum resulting by the vector of the upper confidence intervals of the error rates (transposed) of the contained leaves multiplied by the vector which holds the number of observations in these leaves. This sum is then compared with the upper confidence interval of just one leaf. If the first is higher, he argues that the replacement with one leaf makes sense, due to the higher expected error rate (with consideration of a confidence interval). However, he fails to demonstrate his approach on a toy example which should be replicable (see figure 4-1. Decision tree before and after pruning on page 38 of [81]). Therefore, the author of this master thesis could not agree more with his statement in his book, which Quinlan confesses after introducing his method:

> Now, this description does violence to statistical notions of sampling and confidence limits, so the reasoning should be taken with a large grain of salt. [81, p. 41]

The following paragraph will shortly describe what Quinlan does technically, but does not describe in detail in his work [81]. His main idea was to model the error rate by a binomial distribution. Binomial because there are only two classes, in our case correctly classified and wrongly classified. The binomial distribution is defined in equation (5.24) (see [77] for details). With this formula it is possible to receive the probability that n out of N elements are correctly or wrongly classified (depending on the definition of p). Let variable p hold the empirical error rate. Then $f(n, N, p)$ would give the probability that exactly n elements out of N are wrongly classified. If the reader is interested in the probability that $n \leq x$, he or she can simply sum up the probabilities for all containing cases (e.g. $x = 2$, then simply use $f(0, N, p) + f(1, N, p) + f(2, N, p)$). The formal definition is shown in equation (5.25).

$$f(n, N, p) = \binom{N}{n} p^n (1-p)^{N-n} = \frac{N!}{n!(N-n)!} p^n (1-p)^{N-n} \tag{5.24}$$

$$F(n, N, p) = \sum_{i=0}^{n} \left(\binom{N}{i} p^i (1-p)^{N-i} \right) \tag{5.25}$$

For the $\hat{p}$ Quinlan adds 0.5 to the the observed or empirical $\hat{p}$ from the test data. Details for the scientific background can be found at [86]. Then Quinlan computed the upper confidence

interval for the estimated probability of each leaf of the node he wants to replace. The author of this thesis is not sure if Quinlan uses the approximation (see equation 5.26) or the exact value, which can be found in Brown et al. [13].

$$CI_{upper} = \hat{p} + z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p}}{n}} \tag{5.26}$$

Then, he computes the upper confidence interval for each leaf and multiplied it with the number of observations resulting in this leaf. Next he sums the resulting values and compares it with the upper confidence interval if the whole node would be replaced by the majority class. If the sum is higher than the latter value, the c4.5 replaces the node with a leaf and the majority class.

In conclusion it should be said that despites the questionable statistical method Quinlan uses, it seems that his approach yields reasonable results. Especially, due to the very optimistic replacement for nodes with many branches, it seems that overfitting can be greatly reduced.

Limitations

Figure 5.3 illustrates the drawbacks of the usage of decision trees. The simplest model to separate the two classes, is shown in the figure by using a hyperplane (dotted line). In contrast the c4.5 decision tree would generate rectangular class borders. This artificial example however, simplifies the reality too much. In general, there are more dimensions available of the input data, where the splits at some point works better then shown above. Before using a decision tree for any applications, it is wise to perform a cross-validation and check the distribution of the resulting classification rate. If the variance of this distribution seems very high, the author would suggest to use another method than a decision tree.

Another Example with the Iris Data Set

In order to show another application of the c4.5 technique and make it more comparable with other methods, the results of an c4.5 algorithm applied on the iris data set is depicted in three different ways below. The iris data has been circumcised such that only the width and length of the petals remain. Then the resulting decision tree is represented graphically in a scatterplot (see figure 5.4), as decision tree (see figure 5.5) and finally as pseudo code with if statements (see algorithm 5.1).

Figure 5.4 gives an very illustrative example of the classification regions[7]

In figure 5.5 the reader can see how the trained c45 classifier looks as a tree.

In algorithm 5.1 the trained decision tree is illustrated as if statements.

Literature and References

Literature about the widely known c4.5 classification tree from Quinlan can be found the following book [81].

[7]Regions which shows the classification result of the trained classifier, if a new observation would be placed randomly at the plot.

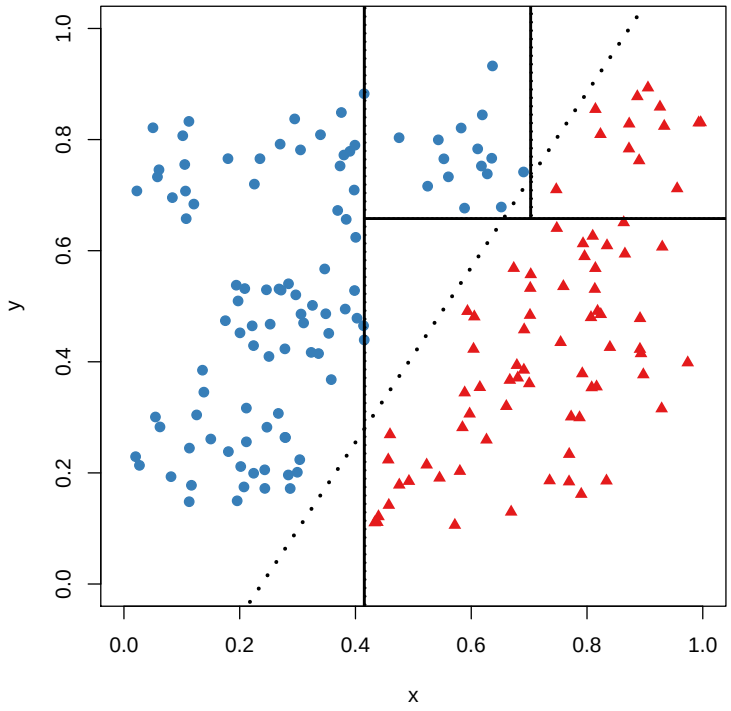


Figure 5.3: The limitations of any decision tree lies in the way it is partitioning the class regions

5.5 k Nearest-Neighbors

Introduction

This section gives an overview of the kNN technique. The kNN is a model-free method for classification. Model-free means that no assumptions have to be fulfilled in order to apply the method correctly. However, the kNN does not evaluate the importance of the different variables. The reader can imagine this classification technique as a kind of black box, where the focus lies on the classification performance rather than on explaining which variables are more important.

The k Nearest-Neighbors as classification technique in detail

Let us suppose that there is a known set (KS) and an unknown set (UnS). Both are distance matrices, where the first one - in comparison to the second one - has an additional class column which indicates the classes of the different observations[8]. The aim of this classification problem is to classify the rows of the UnS with the class information from the KS. For an observation of the UnS the k nearest neighbors from the KS are calculated[9]. Next, the majority class of

[8]The observations are the rows in the distance matrix

[9]Technically, all neighbors have to be calculated and then the k smallest values have to be selected.

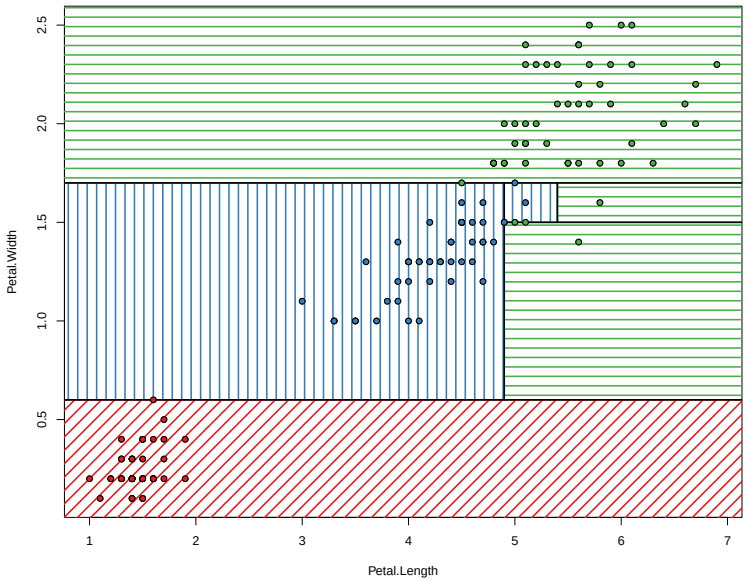


Figure 5.4: C4.5 graphical example of the generated rules

the resulting class vector of length k is assigned to the row of the UnS. These two steps are repeated until all observations/rows of the UnS are classified.

Important questions which might arise at this point are: What does nearest neighbor mean? And when is another observation far from or near to another observation? In order to address these questions some distance metric has to be introduced. In several scientific disciplines (mathematics, statistics, data mining, information retrieval and others), metrics are well known definitions for helping the user to compare the similarity between observations. The most famous one is the Euclidean distance which is shown in equation (5.27). Since the Euclidean distance depends on the unit, scaling and transformation of each column it is wise to transform the whole distance matrix in such a way that the mean of each feature is 0 and the variance is 1. (This is also called *z-transformation* or *standardization*). Of course, there are other distance measures like the Manhattan, Canberra, binary, maximum, Mahalanobis or Minkowski distances. Since the kNN is already a very computationally intensive method, complex[10] distance measures for high dimensional data are almost very unpractical. They might be, from the methodological viewpoint, appreciated but lack practical usage due to long computational times. A good example of such a distance measure is the Mahalanobis distance, whose definition is mentioned below (equation 5.28). During the author's research for this master thesis it became clear that using the Mahalanobis distance for the kNN technique with high dimensional problems is too unpractical in terms of computational time as well as the constraint that the covariance matrix computed by the input matrix must not be singular[11]. Therefore, the author was unable to publish any

[10] in computability

[11] After a dimension reduction with the Principal Component Analysis (PCA) it was possible to compute the covariance matrix, but the results were not as good as with the Euclidean distance and a z-score transformation and

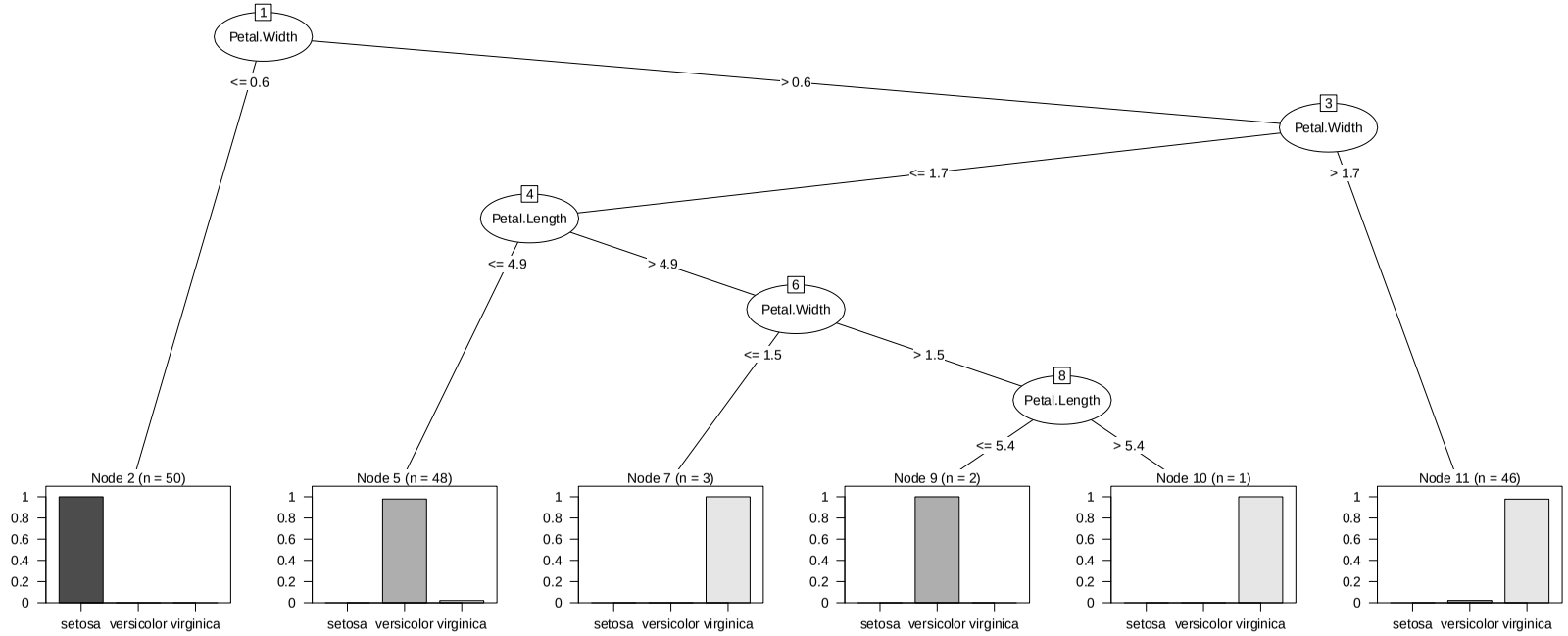


Figure 5.5: The resulting c4.5 decision tree. The class with the highest relative frequency is chosen, whereas other classes in this example are wrongly classified. From left to right, the following classes would have been assigned: *setosa, versicolor, virginica, versicolor, virginica* and *virginica* again. Two notes are wrongly classified, one as versicolor while it is a virginica and a versicolor has been marked as virginica.

```
Data: Vector v which contains the petal length and width
Result: A string for classification
if v.PetalWidth ≤ 0.6 then
    return Setosa;
else
    if v.PetalWidth ≤ 1.7 then
        if v.PetalLength ≤ 4.9 then
            return Versicolor;
        else
            if v.PetalWidth ≤ 1.5 then
                return Virginica;
            else
                if v.PetalWidth ≤ 5.4 then
                    return Virsicolor;
                else
                    return Virginica;
                end
            end
        end
    else
        return Virginica;
    end
end
```

Algorithm 5.1: The c4.5. decision tree, resulting from a transformation of splits.

representative results in appendix D for the Mahalanobis distance.

$$d(i,j) = \|X_{i,.} - X_{j,.}\| = \sqrt{\sum_{n=0}^{N} (X_{i,n} - X_{j,n})^2} \tag{5.27}$$

$$d(i,j) = \sqrt{(X_{i,.} - X_{j,.})^\top \Sigma^{-1} (X_{i,.} - X_{j,.})} \tag{5.28}$$

In these equations (5.27) and (5.28), $d(i,j)$ is the distance between the observation i and j. These observations are rows from the input matrix X of the dimension $M \times N$ where M represents the number of rows such that $0 \leq i,j \leq M$ and N represents the number of columns. $X_{a,.}$ is the vector which contains all columns from row a. $X_{a,b}$ is an element of the input matrix at row a and column b. Σ^{-1} is the inverted covariance matrix of the input matrix X.

therefore dismissed.

Parameter to Optimize

The kNN itself has only one parameter, which can optimize the classification outcome, namely the k value. It represents how many neighbors should be taken into consideration for the class evaluation. However, besides the k value, it can also be interesting to alternatively try different distance measures as well as data transformation as some kind of preprocessing.

In the next sub-subsection the k value are described. Since the other parameters are not unique to the kNN algorithm, they are not further discussed in this sub section.

The k-Value

Choosing the right k-value is a challenging task. A brute force attempt would solve the problem if it was not such a computational intensive task for high dimensional data. Therefore, it is important to carefully select the different instances of k for evaluating the different k values. (e.g. If I want to say that a k value of y is better than a k value of z.)

The following paragraph outlines some possible thoughts in order to select some representative k values for the defined scenarios and tasks from chapter 2. Obviously, the k value must not be too large, since at some point more observations are included than the closest neighborhood, which is not wanted. In addition, the difference between the varying k values has to be high enough, or otherwise the results would be the same. (E.g. consider a k value of x and one of $x + 1$, then the result of this two classifiers would only differ by 1 observation, which makes the probability rather small that they classification will be different. Moreover, if x is already huge, one observation will have almost no influence.) As a result, the differences of the k value may have to increase by a factor and not by a fixed number. Furthermore, since the underlying problem deals only with two different classes, it is important that the k value is an odd number, so there will always be a majority class (in a two class scenario). The following recursive function (5.29) illustrates the chosen k values, which tries to address the outlined problems. The maximum k value should always be smaller than the number of the searched class elements, or otherwise even the best classifier will include some observations, which are not of the desired class. In the considered scenarios for this master thesis, the maximum of k is reached with 63.

$$k(x) = 2k(x-1) + 1 \text{ where } k(0) = 1 \tag{5.29}$$

Computational Considerations

A major drawback of the kNN technique is its time complexity for classification. Usually, the usage of Machine Learning (ML) techniques consists of two phases. First, model estimation and then the application phase. The latter can be used to classify new observations. In general, the first phase can become time consuming when the data set for model estimation is huge. However, with the estimated model, the classification phase can be finished considerably fast and in most cases it is not an issue. The kNN does not have such a first phase as no model is estimated. The similarity calculation has to be done for every classification phase, for each observation in the whole stored data set. After the distance calculation the results have to be sorted. A possibility is the merge sort algorithm. Its time complexity is $n_{\delta,known} log(n_{\delta,known})$ (see [57]),

where $n_{\delta,known}$ is the distance pairs of the known web pages. This merge has to be performed for every distance pair of all web page objects from the new website ($n_{\delta,new}$). For the distance calculation it is required to iterate over the distance pairs of all web page objects from the new website ($n_{\delta,new}$) times the distance pairs of the known web pages ($n_{\delta,known}$) times a constant c_f for the total number of features. Equation (5.30) shows the complexity with respect to the distance pairs. Since it might be more natural to calculate the time complexity in web objects equation (5.31) shows this by putting in the following equations $n_{\delta,new} = n_{f,new} \times n_{f,known,pos}$ and $n_{\delta,known} = n_{f,known} \times n_{f_known,pos}$.

$$\underbrace{n_{\delta,new} \times n_{\delta,known} \times c_f}_{\text{Distance Calculation}} + \underbrace{n_{\delta,new} \times n_{\delta,known} \times log(n_{\delta,known})}_{n_{\delta,new} \times \text{ merge sort}} \tag{5.30}$$

$$\underbrace{n_{f,new} \times n_{f,known,pos}^2 \times n_{f,known} \times c_f}_{\text{Distance Calculation}} + \underbrace{n_{f,new} \times n_{f,known,pos}^2 \times n_{f,known} \times log(n_{f,known} \times n_{f_known,pos})}_{n_{f,new} \times n_{f,known,pos} \times \text{ merge sort}} \tag{5.31}$$

Illustration

The following subsection provides an illustration (see figure 5.6) after applying a kNN on the Iris data set with a k value of 1 and 31. A higher k value smoothes the border between two classes. This can be seen in figure 5.6 between the upper right green and the middle blue class. In general it seems that a small k tends to overfit, while a high k value smoothens too massively. The target for calibrating this method is to find a suitable k.

Advantages

A main advantage is that the kNN classifier does not depend on any assumptions of the feature distribution. In addition, the key concept can be easily understood. Furthermore, there are other applications where the kNN technique could be applied (apart from classification problems) (see section 5.5).

Disadvantages

The k value is computationally intensive to optimize for high dimensional data. In addition, the k value is domain dependent, which means that the k-value has to be examined for each application. Furthermore, the selection of a proper distance measure could be rather challenging for high dimensional data.

Literature and References

Introduction and further information can be found in Ripley [83] and by a more practical book [90]. There are of course also some pages about kNN in Hastie, Tibshirani and Friedman [43] and Bishop [11]).

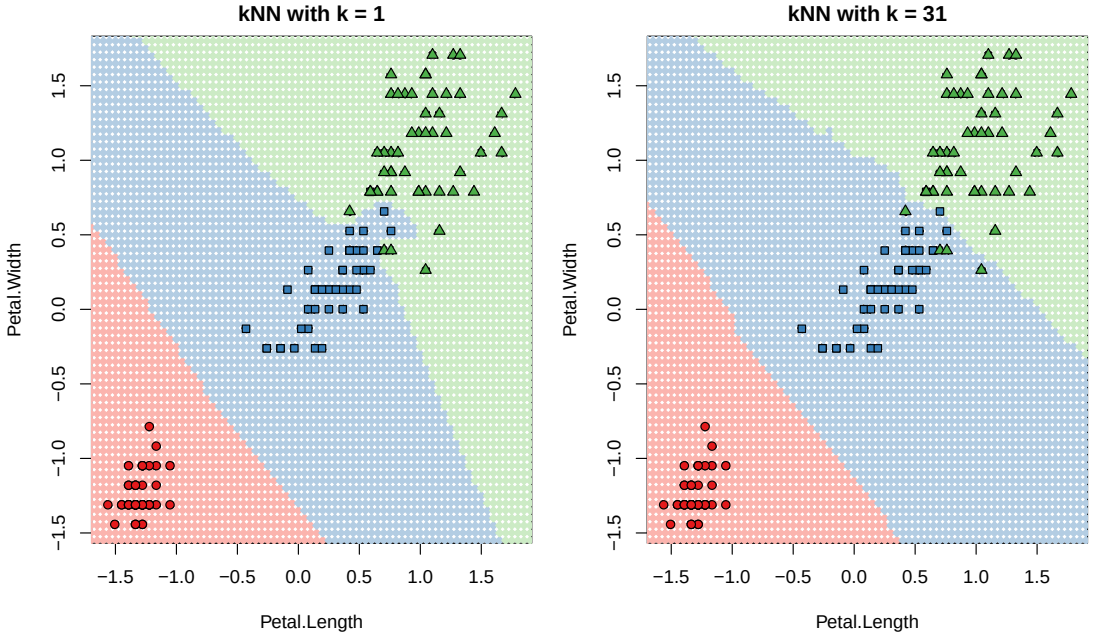


Figure 5.6: 1-nearest neighbor left, 31-nearest neighbor right. The kNN technique was applied on a circumcised Iris data set which only contains the petal length and width (transformed by its z-score). The setosa species is illustrated in red circles, the versicolor in blue squares and the virginica in green triangles. The background color indicates the area on which a new element with a certain petal length and width would be classified.

In practice kNN is not only used for classification problems. In [88] and [76] applications for data imputation are described. Another application for feature selection was published by Xing in [92].

5.6 Support Vector Machines

The machine learning technique of SVM was introduced by Vapnik in [89]. Compared with the other methods which have been introduced in this master thesis the SVM is a rather new one. However, there are numerous scientific publications about this topic. The basic idea of SVM is to find a linear separation between different classes, which are referred as *hyperplanes*. This alone would not have been new, since the logistic regression, linear discriminant analysis and others work very similar. The innovative part of SVM is to search for this linear separation in a high dimensional space, without computing in such a high dimension in practice. Sometimes this is referred to as the kernel trick.

Applications of SVM range from regression, classification and novelty detection problems. For all these versions the mathematical objective function varies. In this master thesis, the focus lies on classification and therefore the used objective function is used for classification problems.

To be precise, it focuses on the so called C-classification. Alternatively, the ν-classification could be used (see Schoelkopf et al. [84]). The differences between C and ν is that in the first the penalizing factor for misclassified observations are defined, whereas for the second (ν) defines the upper bound of the testing error and an lower bound on the fraction of support vectors. The latter is more sophisticated. However, it is only advised to users who know their data quite well and its interpretation is not so intuitive as the C value. In general, the higher the C value, the narrower the hyperplanes margin. In contrast a low C value will result in a larger margin. Here, it has to be said that the high C value, tends to overfit and might be not a good choice.

Formal Model Definitions

The classification problem of SVM could be defined as seen in equation 5.32 in matrix notation respectively 5.33 for non-matrix notation. Karatzoglou, Meyer and Hornik [55] have one side condition different. In the first side condition of 5.32 they stated $0 \leq \alpha_i \leq \frac{C}{m}$, as you see α_i is bounded by 0 and the constant C divided by the number of features n. Since Meyer omits n in a latter publication [68], the author of this master thesis has also omitted it. However, it has only minor effects, since C has to be estimated for different applications and n is a constant for all scenarios.

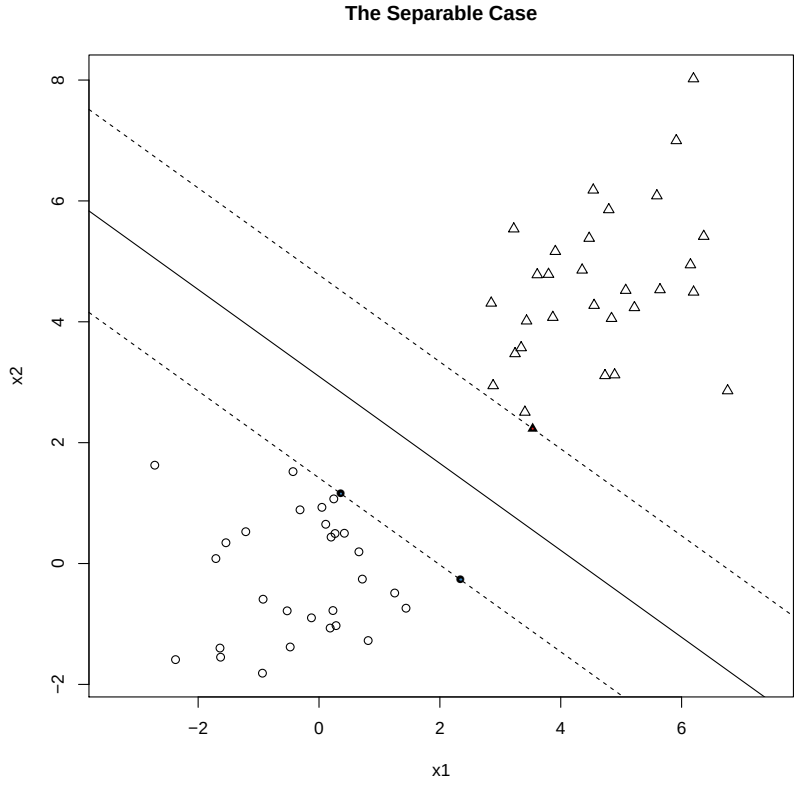
$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & W(\alpha) = \frac{1}{2}\alpha^\top Q\alpha - e^\top\alpha \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \; i = 1, ..., n \\ & y^\top\alpha = 0 \end{aligned} \tag{5.32}$$

$$\begin{aligned} \underset{\alpha}{\text{minimize}} \quad & W(\alpha) = \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i\alpha_j y_i y_j K(x_i, x_j)) - \sum_{i=1}^{n}\alpha_i \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \; i = 1, ..., n \\ & \sum_{i=1}^{n}(\alpha_i y_i) = 0 \end{aligned} \tag{5.33}$$

Hyperplanes

The very basic idea of SVM is to find a linear separation between two (or more) classes. If in a two class scenario all observations are easily linear separable[12], then there is usually an infinite amount of possibilities how to draw such a line. Therefore, the most desired line is that one which maximizes the margin between the two classes. The upper graph in figure 5.7 illustrates the case of separable classes. The hyperplanes are the solid line whereas the both margins (once towards the first class and the other towards the second one) are illustrated in form of dotted lines. Mathematically, the following function has to be maximized with respect to the vector α_i, which is an $(n \times 1)$ vector, where n represents the number of observations. The vector y_i represents the classes which are either -1 or 1 and y_i corresponds to the line of the feature matrix x_i which is a vector with the size of the stored features.

[12] separable by just a linear line.



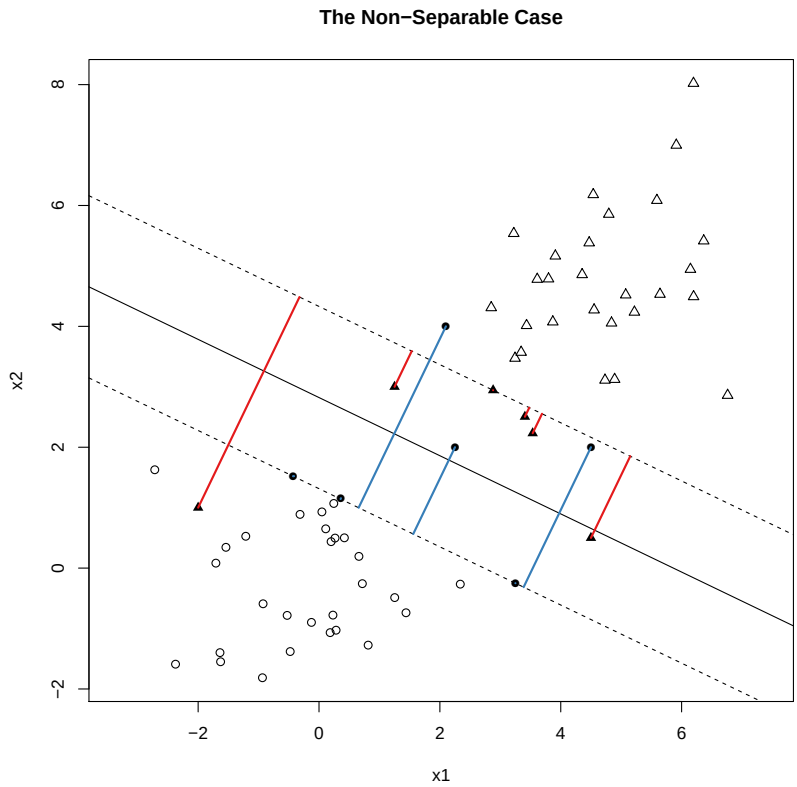


Figure 5.7: illustration of the two possible cases for hyperplanes. The upper graphic shows the case, where the classes do not overlap. In the lower graph the two classes overlap.

$$\begin{aligned} &\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j) \\ &\text{subject to } \ \alpha_i \geq 0 \ \forall i \ 1 \leq i \leq n \end{aligned} \tag{5.34}$$

The constraints shown in formulae (5.35) have to be fulfilled in order to receive a valid solution. Since all α_i are positive, the last equation results in $y_i(\boldsymbol{x}_i^\top \boldsymbol{\beta} + \beta_0) \leq 1$. When $\alpha_i \geq 1$, then the inequation becomes an equation with $= 1$ and $\boldsymbol{x}_i$ lies on the margin. Otherwise it stays > 1 and the observations does not lie on the margin line ($a_i = 0$ in this latter case).

$$\begin{aligned} \sum_{i=1}^{n} (\alpha_i y_i \boldsymbol{\alpha_i}) &= \boldsymbol{\beta} \\ \sum_{i=1}^{n} (\alpha_i y_i) &= 0 \\ \alpha_i (y_i (\boldsymbol{x}_i^\top \boldsymbol{\beta} + \beta_0) - 1) &= 0 \quad \forall i \ 1 \leq i \leq n \end{aligned} \tag{5.35}$$

Non-separable case

In contrast to the previously mentioned case further constraints are included in order to find the desired hyperplane. The lower graph in figure 5.7 shows an illustration about this case. The colored lines which are on the wrong side of the margin are often describe by the letter ξ_i where i shows the index of the observation. The variable holds the value of 0 if the observation i is not on the wrong side of the margin. As a result the total violations are defined as follows: $\sum_{i=1}^{n} \xi_i$. In general, this value should be within a border. In the classical C-classification SVM, this is achieved by introducing a cost variable C. In addition, a new constraint is introduced $\sum_{i=1}^{n} \xi_i \leq C$. As a consequence, each value of α_i has to be lower than or equal to C for the maximization problem from 5.34 above (formally s.t $0 \leq \alpha_i \leq C \ \ \forall i \ 1 \leq i \leq n$). In addition, some other constraints are also added in equation (5.35), see (5.36) for their definitions.

$$\begin{aligned} C &= \alpha_i + \lambda_i \\ \lambda_i \xi_i &= 0 \\ \alpha_i (y_i (\boldsymbol{x}_i^\top + \beta_0) - (1 - \xi_i)) &= 0 \\ y_i (\boldsymbol{x}_i^\top + \beta_0) - (1 - \xi_i) &\geq 0 \end{aligned} \tag{5.36}$$

The variables λ_i are the Lagrange multipliers resulting from the Lagrange optimization function. This technique is used to find local optima which have several constraints. Further details regarding the Lagrange multiplier can be found in [7].

Kernel Functions

This subsection introduces four different kind of kernel functions. The mathematical definition, a short introduction and the possible parameters which have to be estimated are presented. In order to find the optimal parameter for each kernel function, it is general useful to grow them

exponentially. Some authors apply an exponential growth with respect to the base of 2 [26]. However, the author of this thesis has decided to grow it by the bases 10 (with some exceptions), since it appears more natural for humans.

Linear Kernel

The linear kernel is simply the dot product of two vectors. While it seems to be only useful if the data can be linearly separated, is not entirely true. Karatzoglou et al. reported that the linear kernel is a solid choice for applications with vast sparse data (e.g. in text categorization). In addition, it can be also used as benchmark for other kernel functions and does not need much parameter estimation since only the cost C has to be specified.

The following set of values where tested for C: $C_{test} = \{0.1, 1, 10, 100\}$.

$$K(u, v) = \langle u, v \rangle = u^\top v \tag{5.37}$$

Polynomial Kernel

Polynomial kernel functions are widely used for image processing. For the parameter optimization table 5.2 shows which levels where examined. The coefficients c_0 is set to 0, since a constant makes a smaller difference for the classification results than a change in the other parmeters. Therefore, this value is not alterated for the parameter optimization. Fan et al. conduct a similar parameter estimation [26]. They used 1 for the parameter estimations instead of 0 and left it fixed.

Table 5.2: Used parameters for the SVM with the polynomial kernel

Cost C	$\{0.01, 0.1, 1, 10, 1000, 10000\}$
gamma γ	$\{0.05, 0.125, 0.25, 0.5\}$
dimension d	$\{2, 3, 4\}$
Coefficient c_0	$\{0\}$

$$K(u, v) = (c_0 + \gamma \langle u, v \rangle)^d \tag{5.38}$$

Radial Kernel

The radial kernel is also known as the Gaussian Radial Basis Function (RBF) kernel. This kernel is used when the user has no prior knowledge of his or her data. Furthermore, it is the classical classification kernel and it is strongly advised to use it for such applications since it has only few parameters to estimate. In table 5.3, an overview of the parameters which are used for the grid search optimization are shown.

$$K(u, v) = e^{-\gamma \|u-v\|^2} \tag{5.39}$$

Table 5.3: Used parameters for the SVM with the radial kernel

Cost C	$\{0.1, 0.1, 1, 10, 1000, 10000\}$
Gamma γ	$\{0.0005, 0.005, 0.05, 0.125, 0.5, 0.1, 10\}$

Sigmoid Kernel

Mostly referred to as a proxy for neural networks, since sigmoid functions[13] are used within neural networks. Parameter optimization was performed with the following values (see table 5.4).

Table 5.4: Used parameters for the SVM with the sigmoid kernel

Cost C	$\{0.01, 0.1, 1, 10, 1000, 10000\}$
Gamma γ	$\{0.00025, 0.025, 0.125, 1, 10\}$
Coefficient c_0	$\{-2.5, -1.25, -0.5, 0, 0.5, 1.25, 2.5\}$

$$K(u,v) = tanh(c_0 + \gamma \langle u, v \rangle) = \frac{1 - e^{-2(c_0 + \gamma \langle u,v \rangle)}}{1 + e^{-2(c_0 + \gamma \langle u,v \rangle)}} \tag{5.40}$$

Robust Scaling

Due to the fact that the raw data of the distance matrices can contain outliers, which would then dominate the classification since the difference of the observations would be driven by outliers, the data has to be transformed. For that reason, most scientists consider scaling as a crucial tool in order to obtain good results. However, an ordinary z-score transformation might not be enough, since the outliers would still be there in the resulting data. The author of this master thesis has decided to replace the raw data with its relative rank. With this approach it is possible to ultimately eliminate every single outlier.

Parameter Estimation

Parameter estimation is the most important issue which has to be addressed before using a SVM for classification purpose. Some authors [26], suggest first to find a suitable C value and then go on with the remaining parameters. However, the most reliable way is to perform a grid search. Unfortunately, the higher the number of parameters to estimate, the higher the resulting complexity for all possibilities of the grid search is. Some authors have suggested to perform a rough search at the beginning and then continue with greater detail in the interesting regions[14]. For further information regarding the methods for the parameter estimation please consider the following scientific papers [10, 50]. For further details regarding good starting values for the

[13] Sigmoid functions are function which colloquially spoken mapping a input variable x to a sort of s-shape function. Examples are $f(x) = \frac{1}{1+e^{-x}}$, $f(x) = arctan(x)$, $f(x) = tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$, $f(x) = \frac{1}{1+|x|}$ and $f(x) = \frac{1}{\sqrt{1+x^2}}$.

[14] Regions with very high classification results

grid search refer to the work of Fan et al. [26] (as well as the two mentioned in the sentence before) are of interest. Software solutions regarding the parameter estimation in R are presented in [55, 68].

Example Illustrations

In this subsection, two illustrations regarding SVMs are shown. The first illustrates two different kernel functions on the same data sets. The second one shows an overfitting example.

Different Kernel Functions

Figure 5.8 shows the same data classified with a SVM one with a linear kernel function and one with a Gaussian RBF kernel. The data set consists of a random sample of 120 data points (x and y variables, as a consequence 240 random variables) which are normally distributed around the points $(0, 0)$ for class 1 and $(3, 3)$ for class 2. Each x and y value has a standard deviation of 1. In figure 5.8 the boundaries for the two classes are shown in both plots. With the linear kernel function the boundaries are only linear, whereas with the radial kernel function it is possible to form more complex boundaries between the two classes.

Overfitted C Value

Figure 5.9 illustrates an overfitting example for the SVM with a Gaussian RBF kernel function. The data set is the same as above. In order to illustrate the overfitting of the SVM, the cost variable C was changed. On the top, C has the optimal value of 10 (this value was obtained by a 10-fold cross-validation) whereas it is set to $C = 1000$ for the bottom graphic.

Advantages

Via the many possibilities (number of kernel functions, parameter calibration) the user of the SVM technique receives great flexibility. Despite the cumbersome parameter estimation, the SVM technique provides the user with enough flexibility to achieve superior classification results.

Disadvantages

The main disadvantage resulting from the usage of the SVM method is that a lot of parameter analyses have to be done prior to proper classification. These analyses have to address the following questions: How should the data be scaled? Which kernels should be used and with which parameters? The first question might be straight forward, while the second is not so easy to be answered without further analysis.

Literature and References

It seems that there is a lot of scientific literature about SVMs. Especially, besides comprehensive books about statistics and machine learning (see Hastie, Tibshirani and Friedman [43] and

Bishop [11]) receive a lot of attention in journals and other papers. A good introduction to the theory of SVM with elucidatory examples in R[15] is given in [55, 68] (an introduction without R code can be found in [10]). For parameter estimations of the different kernels as well as for the cost constant C Fan, Chen and Lin [26] published a scientific work.

[15] the statistical programming language

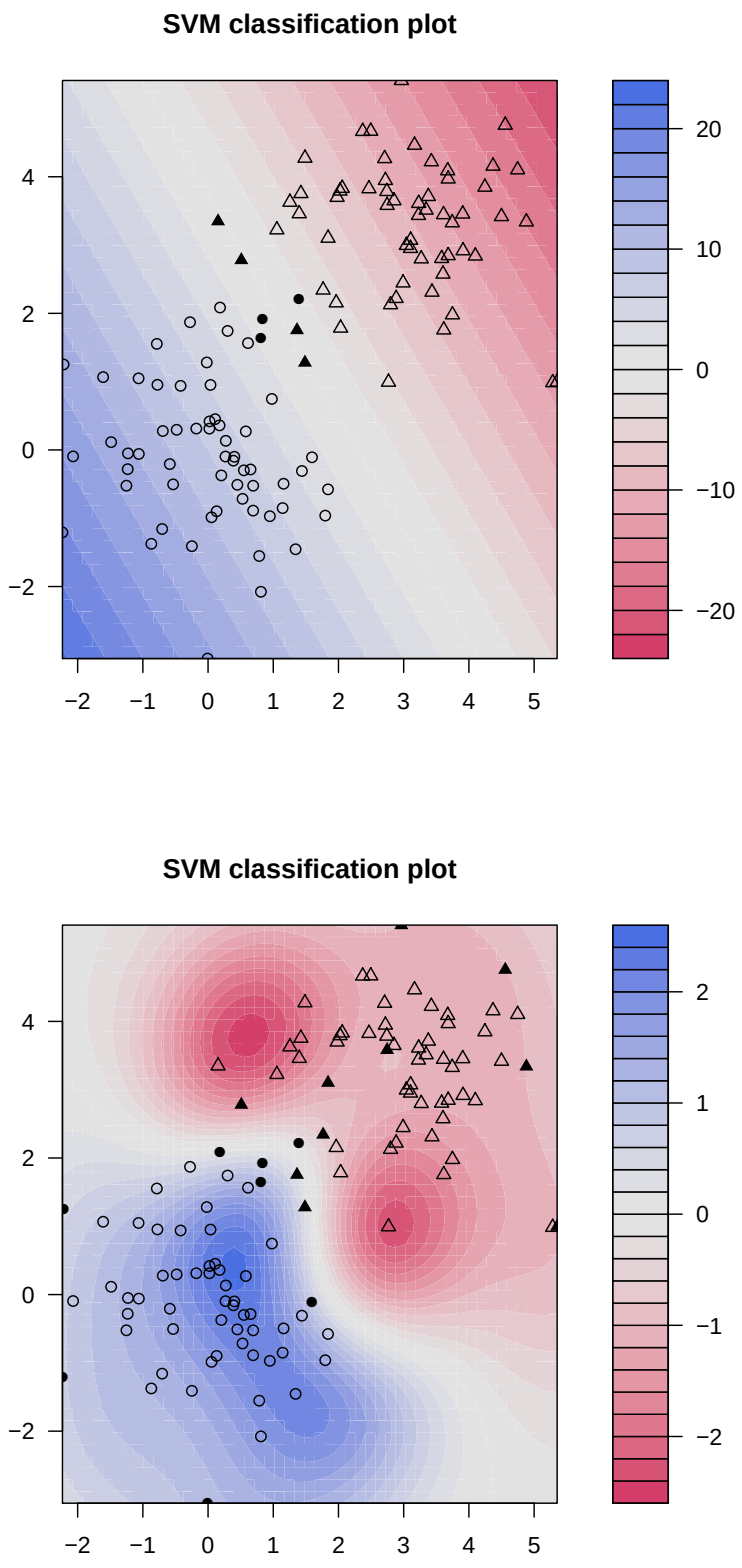


Figure 5.8: different kernel functions. Above with a linear kernel and below with a radial kernel function. The colors indicate the boundaries of the two classes.

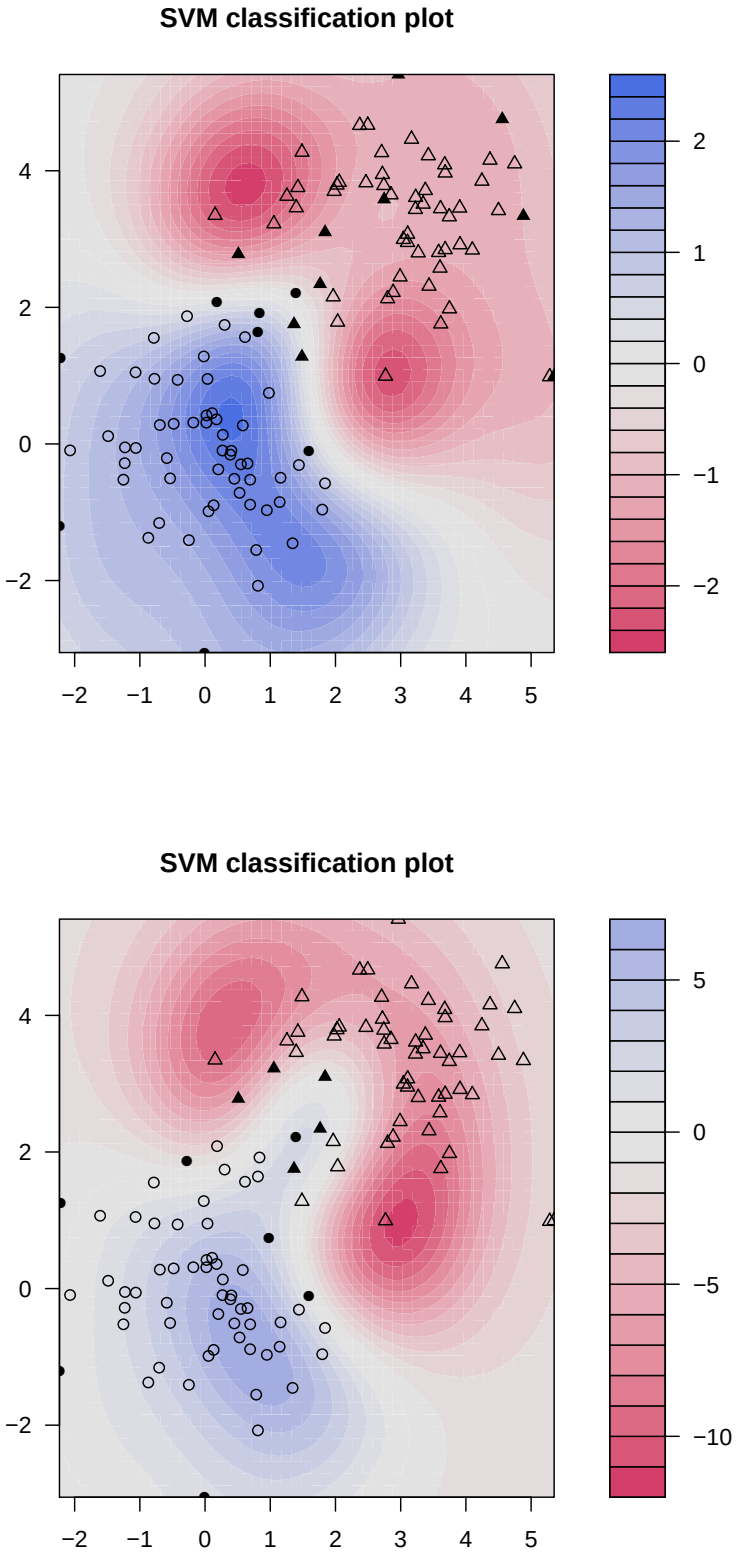


Figure 5.9: illustration of an overfitting example due to the cost variable C. Top $C = 10$ with smoother borders and on the bottom a $C = 1000$ with rather tight borders (the colors represent different values!). Both figures show the SVM method with the Gaussian RBF kernel for a $\gamma = 1$.

CHAPTER 6

Evaluation Methods

The following chapter defines the methods which are used to evaluate the workflow of web object identification which is used in TAMCROW. Besides the actual evaluation itself, the experiments settings are discuessed. Finally, this section shows how the data for the Machine Learning (ML) techniques are prepared (or preprocessed) and postprocessed in order to extract relevant information.

This chapter is structured as follows: Firstly, the tested scenarios are shortly reintroduced. Secondly, different types of preprocessing for the input data are shown. Thirdly, postprocessing procedures are presented. Fourthly, the used cross-validation is defined. Fifthly, the hits count is discussed. Sixthly, some performance measures are defined. Finally, it is discussed how to make different classifiers comparable.

In total, 1,122,805 classifiers were estimated and evaluated in the course of this master thesis.

6.1 Scenarios

The evaluation is based on the scenarios which were introduced in section 2.1 of chapter 2. There are in total four disjoint and one merged scenarios. The first four are searches for bus, train and flight connections as well as for accommodations. The merged one is a combination of the first three scenarios which is referred to as the connection search scenario since it combines all connection searches (bus, flight and train connections).

6.2 Preprocessing

This section provides a short introduction about the preprocessing steps which were used in this master thesis. The main purpose of this is to clean the input data. Therefore, four methods are introduced namely imputation, z-score transformation, relative rank transformation and a dimension reduction with a Principal Component Analysis (PCA).

Imputation

Imputation is the process which eliminates missing values in observations. A lot of effort has been undertaken in science to find procedures and techniques which have particularly good statistical properties. For example median/mean substitutions are widely considered as impractical since they alter the variance structure of a feature. A simple but efficient method is to take a random sample of the *n* most similar observations.

However, in the *TAMCROW project* none of these particular methods are used. Missing values are simply interpreted as the biggest possible distance for its distance feature. This handling of missing values is applied due to their nature. Missing values in the distance matrices are generated for web object pairs, for which one of the web object is not able to possess a certain feature since it does not exist for that type. One may argues that two missing values in a feature pair should be handled differently. However, the TAMCROW team decided to assign the maximum distance, as it is not possible to make a point about this relationship.

Z-Scores

The z-score is a transformation of a vector to map a certain value of this vector to a standard normal distribution[1]. The main advantage of this transformation is that the values of the different features[2] can be compared regardless their different units in a data matrix. A general definition of the z-score can be found in the appendix E in section E.9.

Relative Rank Transformation

This method transforms the values of the matrix in such a way they are in the interval of $[0, 1]$. This can be done with the following approach: Firstly, a rank transformation on the vector is performed. This changes the real values into natural numbers from 1 to the length of the vector. Secondly, every value of the vector is divided by the length of the vector. The resulting vector contains values from 0.0 to 1.0.

Dimension Reduction with Principal Component Analysis

The PCA is a common method in statistics to decrease the number of dimensions[3] of a data set. A reduced data set is more practical to use since, most ML algorithms work faster on it as they need to estimate less coefficients. However, when working with a reduced number of principal components, it is difficult to tell which features help to classify the web object and which are rather useless.

[1] A standard normal distribution has a mean of 0 and a variance of 1

[2] or columns

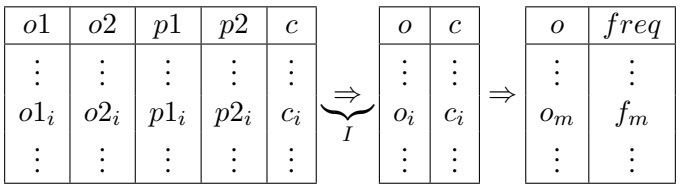
[3] Dimension in this case means the number of features or column of a data matrix.

6.3 Postprocessing

The following section deals with the postprocessing of the classifier's output. The classifier takes a matrix with feature distances for pairs of web objects as input. This matrix contains a class column which is trained by the classifier. The classifier can then estimate the class belongings for the distance pairs of the web objects. From these pairs of web objects, the object which belongs to the unseen web page has to be extracted. This transformation is called postprocessing in this master thesis, because it is a necessary processing step directly after the classification task. The next paragraphs deal with the concrete implementation. An illustration of this implementation can be found in figure 6.1.

At first, a function evaluates which web object id (column o_1 and o_2) belongs to the unseen web page. This function needs the information from which web page web objects 1 and 2 (p_1 and p_2) are and which web page id the unseen web page (new) has. The return value for this function is the web object id of the web object which belongs to the unseen web page. The name of this function is the identification function I.

Then, with the resulting vector of non-unique web object ids it is possible to count them by id and sort them by the resulting count. With this procedure, it is possible to receive a sort of ranking for each id. This quantity represents the likelihood that a certain web object is the searched one. The higher the number, the higher the probability that this web object is the desired one. Therefore, the web object with the highest frequency is chosen.



$$I(o1_i, o1_i, p1_i, p2_i, new) = \begin{cases} o1 & \text{if } p1_i = new \\ o2 & \text{if } p2_i = new \end{cases}$$

Figure 6.1: From classifiers output to object identification. (This figure was taken from [58])

6.4 Cross-Validation Type

For evaluation of the classification precision, a cross-validation is used. This is a common method in statistics or data mining to evaluate a classification algorithm. The standard approach is to divide a data set into a training and an evaluation set. The training set is used for the creation of an instance of a classifier while the evaluation set is used to evaluate the trained classifier. For the evaluation, the estimation of the classifier is compared with the correct classification.

Different kinds of cross-validation methods exist. One of the most common is the k-fold cross-validation. This method divides all available data into k equally sized chunks of data.

Then, the classifier is k times estimated where the evaluation set is changing each run. For every estimation the classifier is evaluated by the corresponding evaluation set.

Figure 6.2 illustrates the difference between an k-fold cross-validation and a k-page cross-validation. Since the web objects originate from different web pages and the real world scenario is to find a *functional* web objects on unseen web pages. It does not make sense to separate the evaluation set into equally sized chunks as the number of web objects of each web page is different. Therefore, the chunks are separated into chunks which depend on their web page belonging. So the results which were obtained for each run, consider every web page once as unseen or new.

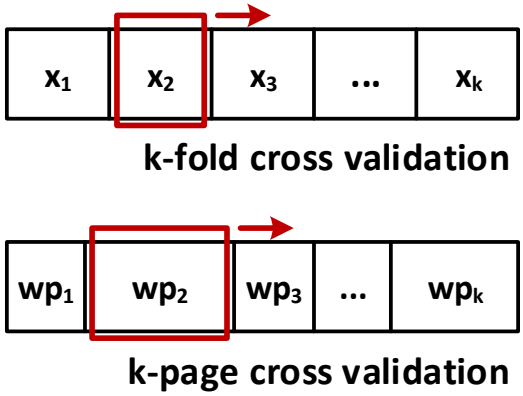


Figure 6.2: top: k-fold cross-validation; bottom: k-page cross-validation

6.5 Performance Measure

This section provides information about the performance measures used for evaluating the classifiers in this master thesis. In data mining and Information Retrieval (IR) there are several ratios to evaluate classifiers. Some of the most commonly used are accuracy (6.1), positive recall (6.2), positive precision (6.3) and f-measure (6.4). These measures derive from comparing the actual class with the estimated class of the classifier. A confusion matrix (see table 6.1) shows how such a comparison could look like.

$$\text{accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{false positive} + \text{false negative} + \text{true negative}} \tag{6.1}$$

$$\text{positive recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{6.2}$$

$$\text{positive precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{6.3}$$

Table 6.1: Confusion Matrix

		actual classification	
		positive	negative
estimated classification	positive	true positive	false positive
	negative	false negative	true negative

$$\text{f-measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6.4}$$

For evaluating the machine learning algorithm from chapter 5 only the *positive precision* (6.3) has been used for each task. This is due to the nature of the search task. The aim in the different search scenarios is to find only one functional web object. As it is unimportant for this task how many negative web objects are correctly classified, a lot of ratios become uninteresting. A short example will illustrate this mode of thought. Imagine a task where you have to find the one positive example from 100 web objects. Here some would interprete this task as a classification problem with 1 positive and 99 negative web objects. However, this is actually wrong. When one is not able to find the one positive it does not make any difference if one correctly classifies 97, 96 or none. An other approach might be, to classify only one web object as positive and the rest as negative. Therefore, someone would has a classification rate of 100% if someone finds the positive web object or 98% if someone does not find it. Equation (6.6) illustrates this example. Here, the reader immediately sees that the 98% produce the illusion of an extraordinary classification result, but in reality it is most certainly not.

$$accuracy = \begin{cases} \frac{1+99}{100} & \text{if the positive element is found} \\ \frac{0+98}{100} & \text{if the positive element is not found} \end{cases} \tag{6.5}$$

In compliance with the example above, the author therefore chooses the positive precision as a performance measure which addresses the problem illustrated above. Equation (6.6) shows the definition of the positive precision. If the positive web object is found the positive precision is 100% and 0% otherwise.

$$\text{positive precision} = \begin{cases} \frac{1}{1+0} = 1 & \text{if the positive element is found} \\ \frac{0}{1+0} = 0 & \text{if the positive element is not found} \end{cases} \tag{6.6}$$

Result Aggregation

The different classes of the web object distance pairs are not uniformly distributed. In fact, their class sizes are in a great discrepancy to each other. As a result, the classifier would be biased towards the majority class[4]. Therefore, an under-sampling strategy has been used to overcome this problem. In an under-sampling strategy, only a fraction of the observations from the majority class are used. It is not sufficient to limit the sampling to a percentage of the number

[4]Which is in every scenario every time the negative class

of elements of the majority class, since the ratio between the positive and negative class is not fixed for all scenarios. Therefore, a ratio has been used telling the portions of negative and positive observations. In the further work it is referred to as the negative positive ratio and is defined as seen in equation (6.7), where $|x_.|$ is the length of vector $x_.$, x_{neg} is a vector holding all elements from class negative and x_{pos} is a vector holding all elements from the positive class. In subsection 7.2 results about the right size of this class are discussed.

$$\text{negative positive ratio} = \frac{|x_{neg}|}{|x_{pos}|} \tag{6.7}$$

6.6 Counting Hits

This section provides information how the discovered web objects on the new or previously unseen web page are counted. In general, it sounds trivial, but if the searched task is modelled by one web object within this web sites, then the classification result is 100 % when the classifier finds this web objects or otherwise it is 0%. Figure 6.3 shows this example with the green rectangle. In contrast, sometimes web developers use more than one web object to model a certain functionality or task. Figure 6.3 shows this for the arrival date on the booking.com web page with two red rectangles. In this case, the TAMCROW team has decided that if the classifier finds only one of them, this is counted as a classification rate of 100%. One might now argue that this is unjustified, since it does only half of the job. However, there are two reasons why this approach was chosen. The first one is simple, because the technical settings and the problem description for the experiment are less complex and therefore easier to formulate and program. The second one, which is illustrated in figure 6.4 and might be more convincing, is that mostly web objects which fulfill one task together are encapsulated in a division html element. This knowledge can then be used in a further post-postprocessing step to identify all web objects which have to be used to perform a certain scenario correctly.

6.7 Comparing Results from different Classifiers

This section is mostly based on the work of Nadeau and Bengio in [72]. In IR, Data Mining (DM) and other scientific disciplines which are applying statistical learning methods, it is crucial to compare different classification algorithms. This is mostly done with statistical tests, which tell if the better outcome of a classifier is just arbitrary or significantly better over another. Significantly in this sense means that it is not for sure better, but with very high probability.

A very common and basic method is a ordinary *t-test*, which tests if the means of the different classifier outcomes are equal or not due to the cross-validation (grouped by the different classification methods). In statistical tests there is always a basic assumption namely the *null hypothesis* or short H_0. In our case, the $H_0 : \mu_{c1} = \mu_{c2}$, where μ_{c1} stands for the mean of the performance ratio[5] of classifier 1 ($c2$ symbolizes classifier 2). In case that the probability drops

[5] As introduced in section 6.5, the performance measure for the scenarios in this master thesis is the positive precision. (See equation 6.3)

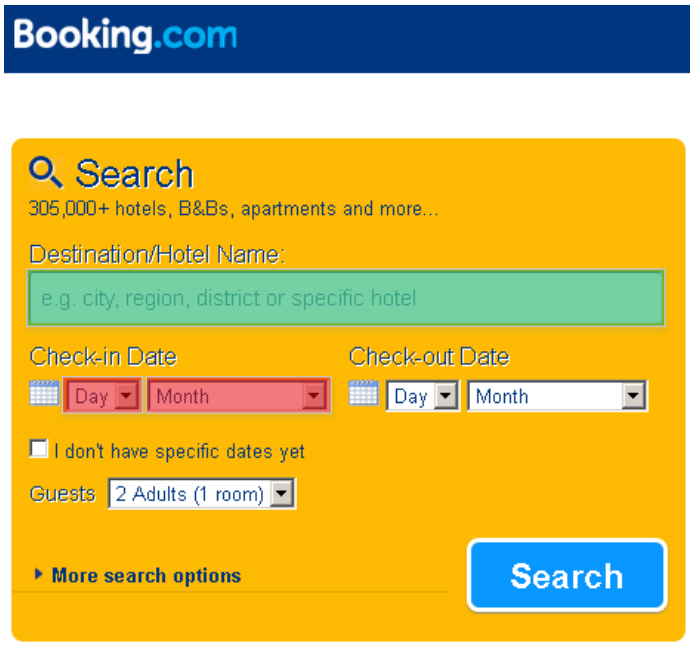


Figure 6.3: Counting example. The green box illustrates that only one web object is used to model this task, whereas red illustrates the case where more than one web object has been used.

below a predefined α value (given in %), H_0 is rejected and H_1 is assumed. H_1 is the *alternative hypothesis*, which is just the opposite of the *null hypothesis*, in this case $H_1 : \mu_{c1} \neq \mu_{c2}$.

Most of the above mentioned knowledge are provided by an introductory book about statistics (e.g. Freedman, Pisani and Purves [35]). However, when going into further details, estimating the probability that the means are equal or not is not so trivial. Actually, there is a simple approach namely the ordinary *t-test*, but it often produces too liberal decisions. The main problem is that it underestimates the variance of the performance measure, which leads to a too narrow confidence interval[6] and this leads to an increase in the probability of rejecting H_0. In [72] Nadeau & Bengio evaluate different estimators for the test variance. As a result, the authors suggest to use the *corrected resampled t-test*.

Formal Definition of the corrected resampled t-Test

This section provides the necessary mathematical definitions for calculating the confidence intervals (equation 6.8). $\hat{\mu}_{cl}$ is an estimator for the the arithmetic mean (equation 6.9) of the computed performance ratios for classifier cl. c is a critical value (equation 6.11). $\hat{\sigma}^2$ is the esti-

[6]The outcome of a statistical test corresponds to the length of the confidence interval. If two confidence intervals overlap, then H_0 in a statistical test would not be rejected. If two confidence intervals do not overlap, then H_0 is rejected and H_1 is assumed.

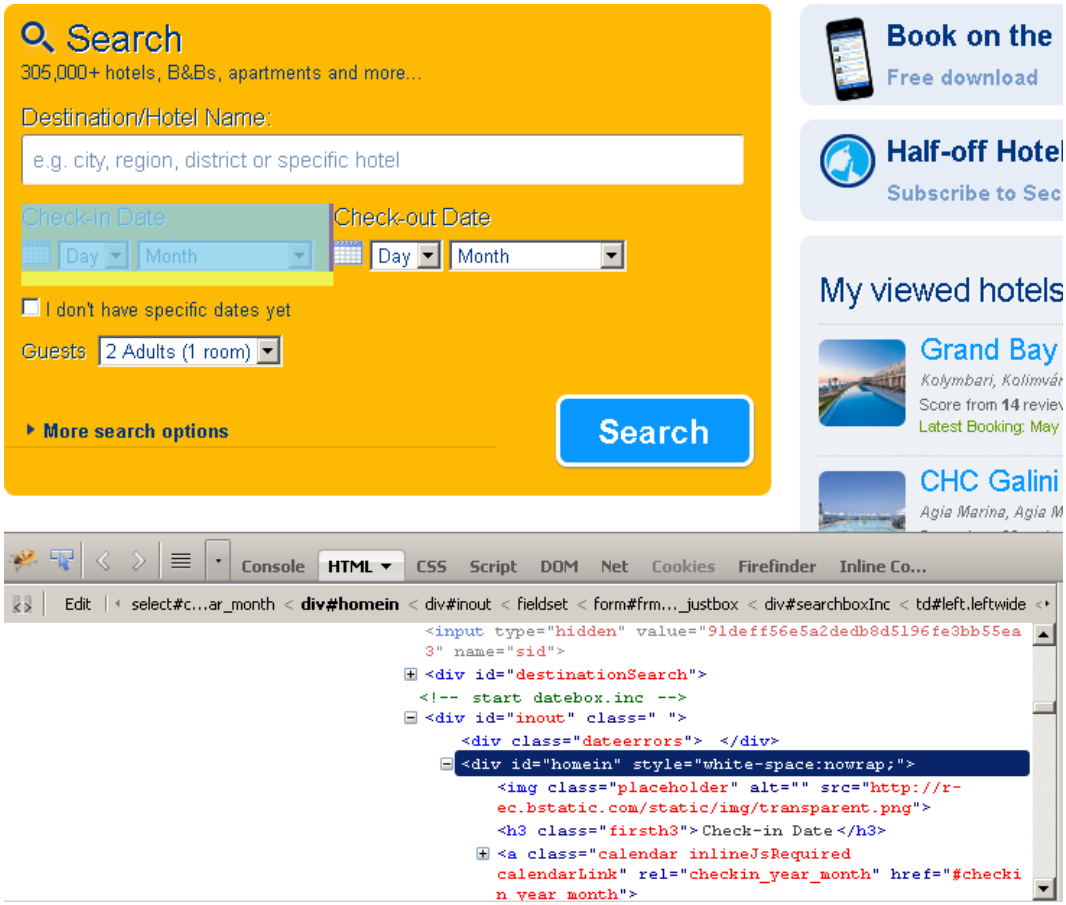


Figure 6.4: Screen shot of booking.com and the DOM tree with firebug. Web objects which fulfill a task together are often encapsulated in a div-html tag as shown on the screen shot.

mator of the variance of $\hat{\mu}$ (equation 6.10). J is the number of runs. $x_{cl,j}$ is the j-th element of the performance ratio from classifier cl. $S^2_{x_{cl}}$ is the sample variance of the obtained performance ratios (equation 6.12). n_1 is the number of observations used for the training set, whereas n_2 is the number of observations used for the testing or evaluation set. The ratio between n_1 and n_2 is fixed by equation 6.13 during the whole cross-validation procedure. $t_{J-1,1-\frac{\alpha}{2}}$ is the quantile of a t-distribution with $J-1$ degrees of freedom for the probability $1-\frac{\alpha}{2}$.

$$\text{confidence interval}(CI) = \hat{\mu}_{cl} \pm c * \sqrt{\hat{\sigma}^2} \tag{6.8}$$

$$\hat{\mu_{cl}} = \frac{1}{J}\sum_{j=1}^{J} x_{cl} \tag{6.9}$$

$$\hat{\sigma}^2 = \left(\frac{1}{J} + \frac{n_2}{n_1}\right) * S^2_{x_{cl,j}} \tag{6.10}$$

$$c = t_{J-1,1-\frac{\alpha}{2}} \tag{6.11}$$

$$S^2_{x_{cl}} = \frac{1}{J-1}\sum_{j=1}^{J}(x_{cl,j} - \mu_{cl})^2 \tag{6.12}$$

$$n_1 \approx \frac{N_{pages}-1}{N_{pages}} \text{ and } n_2 \approx \frac{1}{N_{pages}} \tag{6.13}$$

$$\frac{n_2}{n_1} = \frac{\frac{1}{N_{pages}}}{\frac{N_{pages}-1}{N_{pages}}} = \frac{N_{pages}}{N_{pages} * (N_{pages}-1)} = \frac{1}{N_{pages}-1} \tag{6.14}$$

Since the probability receiving a value that is higher than 1 or lower than 0 is 0, the confidence interval (CI) is restricted within the interval $[0, 1]$. Formally, this is defined in equation (6.15).

$$\begin{aligned} CI_{upper} &= min(1, \hat{\mu}_{cl} + c * \sqrt{\hat{\sigma}^2}) \\ CI_{lower} &= max(0, \hat{\mu}_{cl} - c * \sqrt{\hat{\sigma}^2}) \end{aligned} \tag{6.15}$$

CHAPTER 7

Evaluation Results

This chapter will provide the results of the classification algorithms. The different machine learning algorithms, introduced in chapter 5, were used to find certain functional objects on *unseen* web pages.

Firstly, the evaluation workflow in general is introduced. Secondly, the results of the parameter estimation are provided. Thirdly, the extended results are graphically illustrated. Finally, further analysis of the different visual features is shown.

7.1 Evaluation Workflow

This section provides a short overview of the order of the generated results (see figure 7.1).

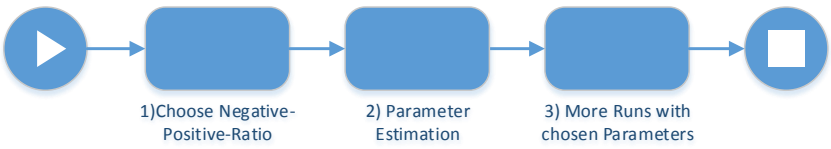


Figure 7.1: The evaluation workflow

Firstly, the logistic regression and the c4.5 decision tree are used to estimate a negative positive ratio (NP-ratio) for further evaluations (see section 7.2). The logistic regression and the c4.5 decision tree has been used because they do not require any complex and sophisticated parameter tuning compared to the support vector machines (SVM) and k Nearest-Neighbors (kNN). However, the results are valid for the SVM as well, since the model is similar to the logistic regression. This might not be the case for the kNN, but since it is infeasible to provide a parameter optimization for the kNN, the results from the logistic regression and the c4.5 decision tree are used for the kNN as well. At this stage 61,560 classifiers were trained.

Secondly, parameters of the different classifications are estimated with the resulting NP-ratio. This is especially important in order to receive good results for the SVM with its different kernel functions. For the logistic regression, different types of data transformation were performed and evaluated. In this part 755,345 classifiers were estimated.

Thirdly, after the most usefull parameters for the different classifiers were evaluated and selected, more runs were performed. This is important because of the subsampling approach. In this part 305,900 classifier were estimated (43,700 for each classifier).

In total 1,122,805 classifier have been estimated and evaluated for this master thesis.

7.2 Results of the Parameter Estimations

Negative Positive Ratio

This subsection discusses which value of the NP-ratio should be used for classification. The logistic regression and the c4.5 algorithm have been examined, since both have no other parameters to estimate and their training phase is computationally cheap compared with the others Machine Learning (ML) techniques mentioned in chapter 5.

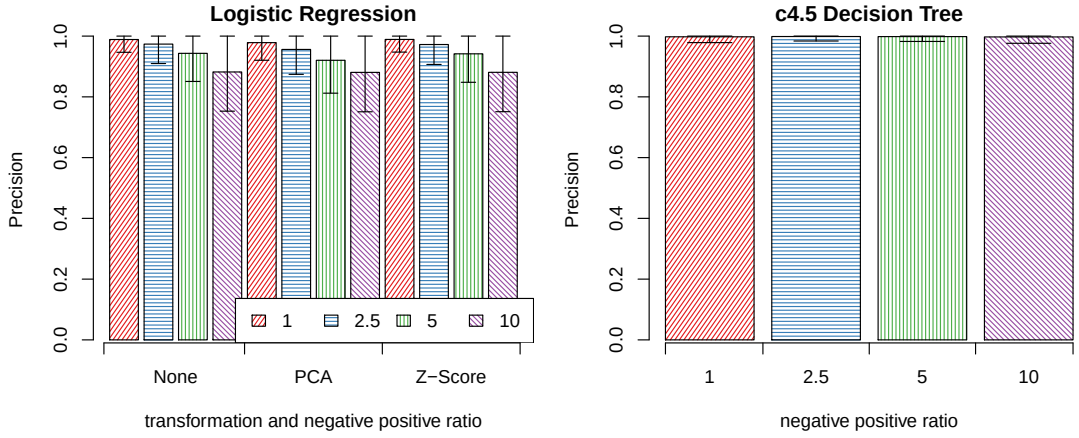


Figure 7.2: left: the results for the logistic regression with different data input transformations (none,pca and z-score); right: the results of the c4.5 decision tree. In both examples the mean precision is computed for 4 values of the NP-ratio 1(red),2.5(blue),5(green) and 10(purple) (from left to right). The whiskers illustrate the 90% confidence interval for a corrected resampled t-test (see subsection 6.7). The data is available in table D.1 in appendix D

Figure 7.2 shows the average mean precision for the c4.5 and the logistic regression, where for the latter also different data transformation methods have been used, namely none, a z-score transformation as well as a Principal Component Analysis (PCA) dimension reduction. The right plot shows that for the c4.5 the number of classes makes no difference, due to the nature of the c4.5 classifier. On the left side, the logistic regression seems very sensitive towards the NP-ratio.

Even if it might not seem to be statistically significant for an $\alpha = 0.1$, there is a decreasing trend which can not be denied, especially when taking the test setting into consideration. The tests with a higher NP-ratio contain all negative and positive observations from the test with the direct lower NP-ratio plus some additional negative observations to get the desired ratio. So all tests undertaken for the 2.5 NP-ratio contain all observations from the 1.0 NP-ratio plus 1.5 parts negative observations to receive a 2.5 NP-ratio (see figure 7.3). This means that adding negative examples to the learning phase[1] of a logistic regression decreases the mean precision of the classifier for an unseen web page. It would be natural to assume that the opposite be the case, namely the more information is added the better the classification results become. However, this is the phenomenon the author mentioned in section 6.5. A logical reason could be that the classifier is overfitted towards the negative examples and performs therefore less accurately in the cross-validation. Since, the SVM is mathematically more similar to the logistic regression than to the decision tree, a NP-ratio of 1 is chosen for estimating the different parameters of the SVM. Furthermore, a low NP-ratio does not have a negative influence on the c4.5. For the kNN also a NP-ratio of 1.0 was chosen.

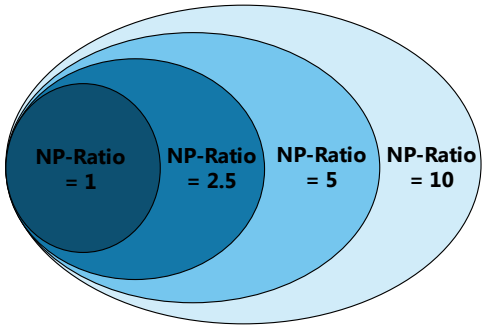


Figure 7.3: Graphical illustration of the NP-ratio test settings

Details to the Data Preprocessing for the Logistic Regression

This subsection provides results for the different methods of data preprocessing used in this master thesis, namely none, z-score transformation and a PCA dimension reduction within the logistic regression. Figure 7.4 illustrate the results per scenario and data preprocessing. For the PCA dimension reduction, the number of principal components whose cumulative variance sum up to at least 90% has been choosen. As the reader might see, there are no significant differences resulting from the type of data preprocessing. Furthermore, it seems that the 90% PCA reduction yields to a slightly worse mean precision. (Especially, for the flight connection search, there is a decrease of almost three percentage points. This might seem to be only a minor difference. However, when considering a mean precision with an average of 99.49% for none and the z-score transformation, the number of wrongly classified web objects is more than six time higher

[1] The phase when the coefficients of the classifier are estimated.

for the PCA version ($\frac{1-0.9692}{1-0.9949} = 6.0392$). Since no transformation is almost everytime as good as a transformation by the z-scores (exception is the all connection search scenario -0.05%), the further results are computed without any transformation, because the classifcation workflow is shorter. Particularly, the z-scores need to be calculated including the data of the unseen web page. Therefore, it is not possible to compute a classifier in advance and store its coefficients. This is major drawback, since a fast just-in-time classification would become impossible with a z-score transformation.

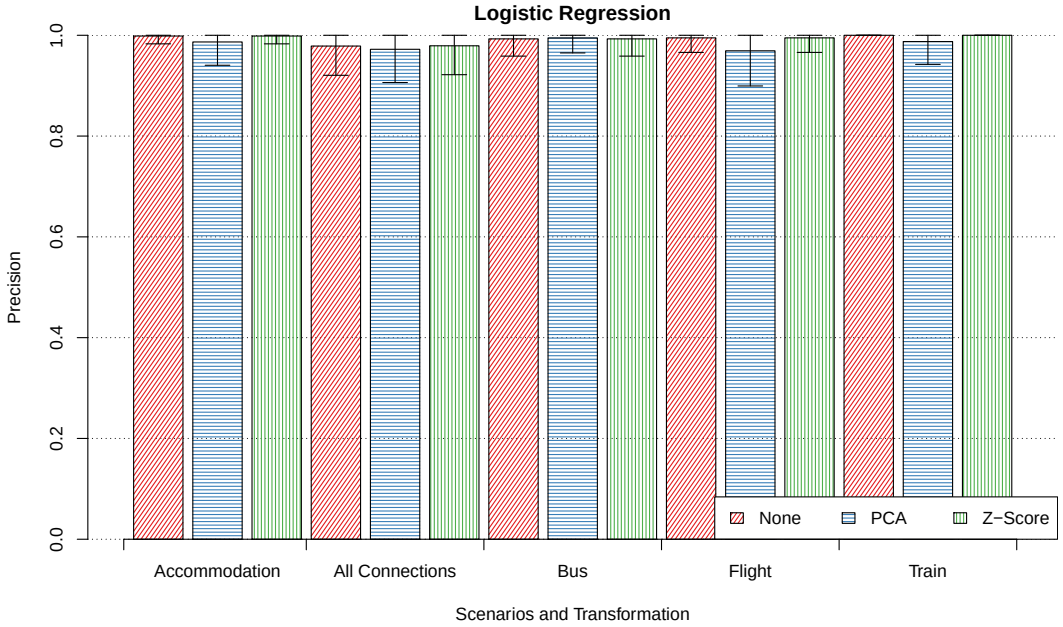


Figure 7.4: shows the results of the logistic regression per scenario. For details see table D.3 in appendix D.

kNN

The classification results of the kNN are shown with a z-score transformation as well as PCA dimension reduction where at least 90% of the total variance are described. Figure 7.5 are divided into the five scenarios. For the bus, train and flight connection it does not make any difference which k-value or which transformation is used. However, in the accommodation scenario it becomes obvious that the mean precision descreases with a higher k value. The PCA transformation has a slightly unfavourable influence on the mean precision. Therefore, for further computations k is set to 1 and only a z-score transformation is applied to the input data of this classification algorithm.

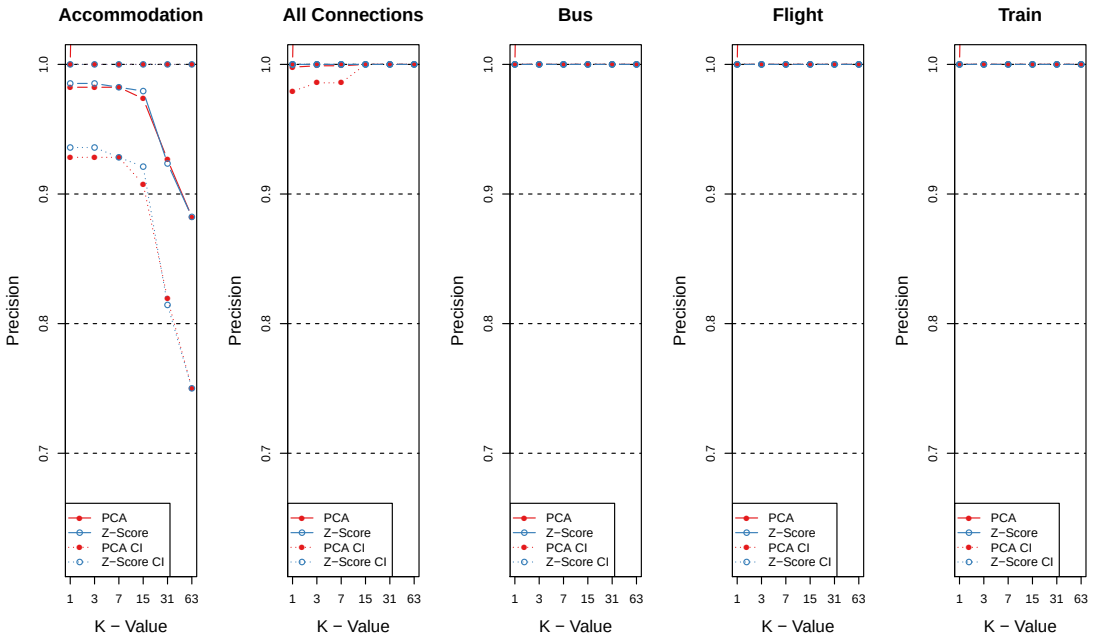


Figure 7.5: Graphical illustration of the precission for the kNN classifier. The exact results can be found at D.8 in appendix D. Every plot shows a different scenario (from left to right: the accomodation search, the all connections search, the bus connection search, the flight connection search and the train connection search).

Support Vector Machines - Linear Kernel

Figure 7.6 gives an overview of the SVM with the linear kernel. It seems that the cost C does not make any difference for the results. When taking a look at table D.5, it becomes clear that there is a very small difference. The best results are achieved when setting C to 0.1. Therefore, this setting is used to compute further results.

Support Vector Machines - Polynomial Kernel

In figure 7.7 the reader finds three 3d scatterplots in the upper row and three 2d scatterplots in the bottom row. The upper scatterplots show the relation between the mean precision on the one side and the cost C and γ on the other for the dimensions 2,3 and 4 (from left to right). From the 3d scatterplots it is hard to say which combination of parameters will cause the best performance for the mean precision value. The scatterplots in the bottom row reveal that chosing a dimension value of 4 is better than 3 and 2 (not statistically significant). However, these results are within 99.4 and 100 % which is a rather small interval. In order to receive a better understanding of the top results, it might be better to take a look at table D.5 in appendix D. There, the reader can see that among the best results (last column) most of them have the following values: a cost C of

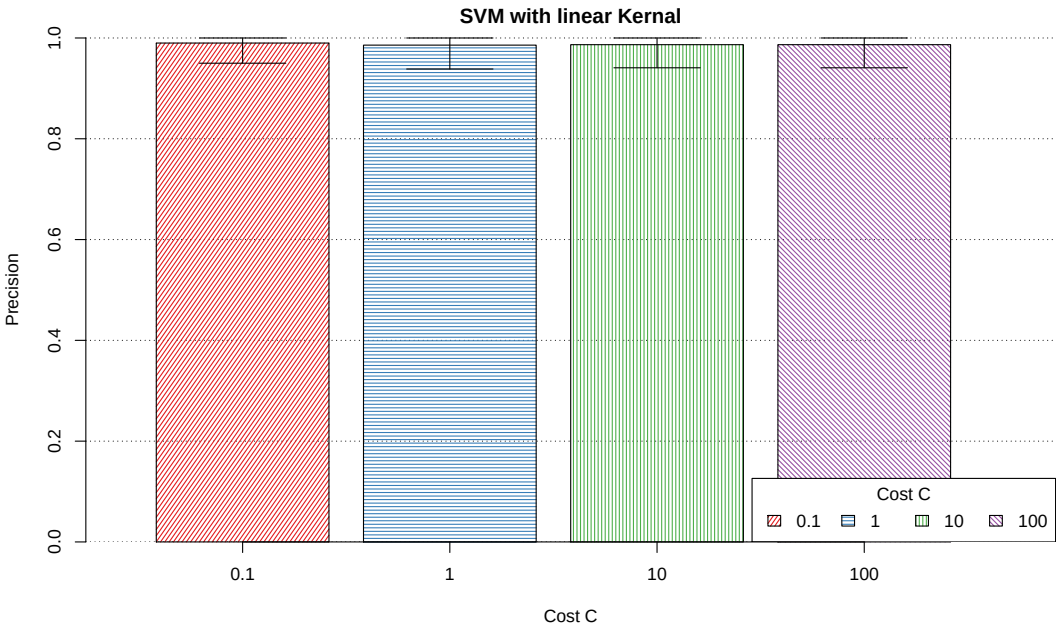


Figure 7.6: The results of the SVM with the linear kernel for different cost C. For detail results see table D.4 in appendix D.

0.01 and a γ value of 0.05. For the dimensions there is a tie between 2 and 4. However, when taking into consideration the bottom right plot, the spread between the results for dimension 4 is much smaller than for dimension 2. Therefore, a polynomial kernel with the parameters $C = 0.01$, $\gamma = 0.05$ and $dimension = 4$ is chosen for further performance measures.

Support Vector Machines - Radial Kernel

Figure 7.8 shows the results for the parameter estimation for the SVM with the radial kernel function. What can be directly seen from the 3d scatterplot on the upper left side is that the mean precision is higher for bars in the front than bars in the back (by means of the y-axis). Furthermore, γ has a higher influence towards the mean precision than the cost C (seen at the upper right and the bottom left plot). However, it seems that mean precision could not further be increased by chosing a much lower γ value. The best performance for the mean precision (100%) is reached when setting the cost C to the value 0.1 and γ to 0.0005. This result is significantly[2] better than every result with an upper confidence interval below 1 or 100% (that are 20 out of 35 rows in table D.6).

[2]by means of a statistical test (in this case the corrected resampled t-Test).

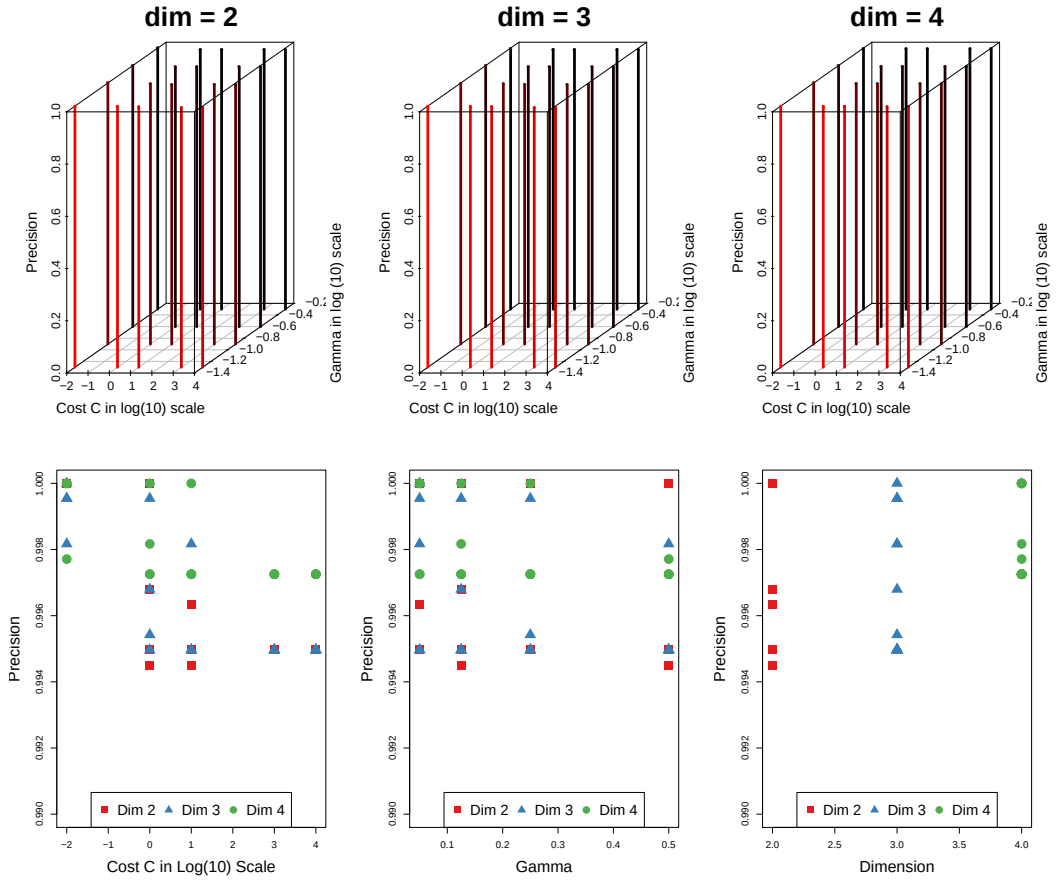


Figure 7.7: This figure shows the result from the SVM with the polynomial kernel. The upper row illustrates three 3d scatterplots for cost C, γ and mean precision (with dimension 2, 3 and 4 from left to right). The bottom row shows three 2d scatterplots which illustrate the relation for the mean precision on the one side and to the C, γ and the dimension (from left to right) on the other. C and γ are in log_{10} scale (except for the bottom middle 2d scatterplot). For detail results see table D.5 in appendix D. The results contain the estimation of 131,100 classifiers.

Support Vector Machines - Sigmoid Kernel

The results from the parameter optimization of the SVM with a sigmoid kernel function can be found in figure 7.9. At the first glance, the illustration seems not to be informative. However, when considering that the top results converge to a certain point, the best results are achieved with those parameters that cause the smallest spread. In the left plot of figure 7.9 the smallest spread for the mean precision is received when setting the cost C to $10^0 = 1$. In the middle figure, setting γ to 1 results in the smallest spread. For the right figure, it is more difficult to identify where the smallest spread is. However, when eliminating the result with the lowest mean precision the coefficient $c_0 = 2.5$ yields the smallest spread. So the best setting is achieved when setting $C = 1$, $\gamma = 1$ and $c_0 = 2.5$. Another approach of selecting the parameters is to count the

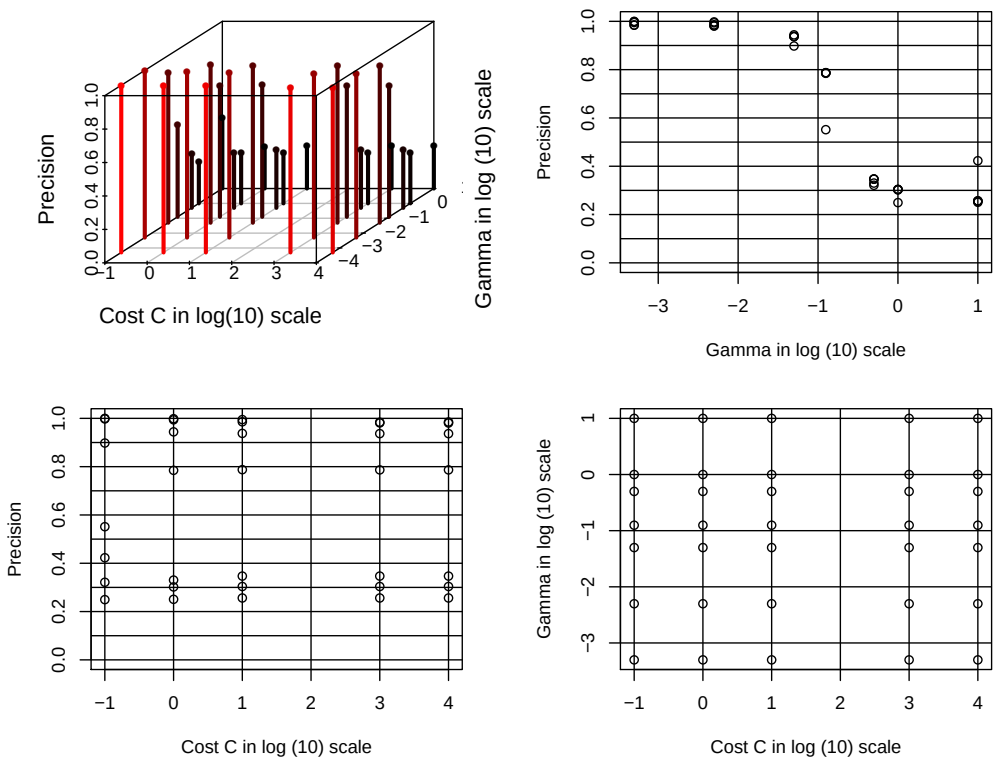


Figure 7.8: Results from the parameter estimation for the SVM with the radial kernel. For details see table D.6 in appendix D. The upper left plot shows a 3d scatterplot for the cost C in a log_{10} scale (x-axis), γ (y-axis) in a log_{10} scale and the mean precision (z-axis). The upper right plot shows a 2d scatterplot between γ and mean precision. The bottom left plot shows a 2d scatterplot between the cost C and the mean precision. The bottom right plot shows a 2d scatterplot between the cost C and γ.

number of appearances of the different parameters in the detailed result table D.7 in appendix D where the column *Best* is true. (This lead to the same result as above.)

7.3 Extended and Aggregated Classification Results

This section provides the results with optimized parameters for each ML technique. In total 305,900 classifiers have been trained, 43,700 for each technique. The first part shows the results aggregated for all scenarios. Then the results are shown for each scenario separately.

Overview of the Aggregated Results

Figure 7.10 shows the aggregated results for the ML algorithms after the parameter optimization. Herefore, 305,900 runs have been executed (43,700 for each classifier). The black lines indicate

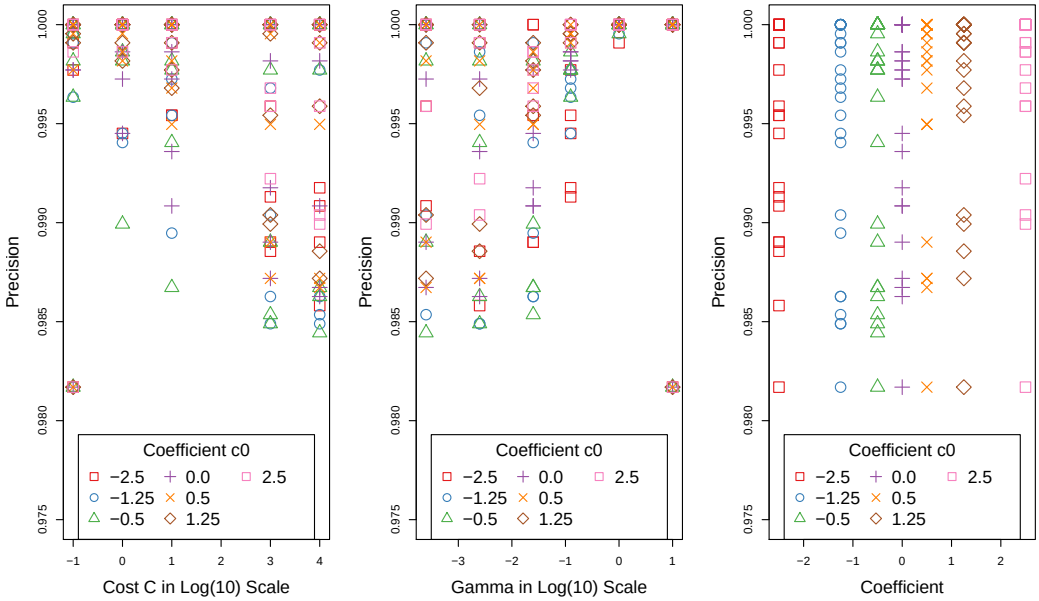


Figure 7.9: Results from the parameter estimation for the SVM with the sigmoid kernel. For details see table D.7 in appendix D.

the averages of the mean precision, where the colored rectangles indicate the confidence intervals for each technique. It clearly shows that the SVM with the radial and the polynomial kernel outperforms all other techniques and finds all elements. These are remarkable results. However, the results of the other classifiers also perform well. All classify the correct element in 98 of 100 cases, and 6 out of 7 classify in 99 of 100 cases correct.

Details of the Aggregated Results per Scenario and Task

The Accommodation Search Scenario

Figure 7.11 gives a detailed insight into the results for the *accommodation search* scenario. The figure includes plots for different tasks and shows the mean precision as well as the confidence for each algorithm. Firstly, the *adult passengers* object occurs in 11 web pages and were mostly addressed very well. Only the logistic regression and the SVM with the linear kernel did not find all web objects, but with 99.91% they still obtain very good results. Secondly, the *from date* object seems more challenging compared to the task above. What can be easily seen is the fact that the c4.5 decision tree performs rather poorly. Thirdly, the *departure date* object was well solved by all techniques except the SVM with the linear kernel. Fourthly, the task for *nights* object was indentified by all techniques without any problems. Fifthly, for the *submit button* task the SVM with the linear kernel function did not perform as well as the other kernel functions. Sixthly, the task of finding the *where location* object seems to be very difficult for the kNN as well as the SVM with the linear kernel.

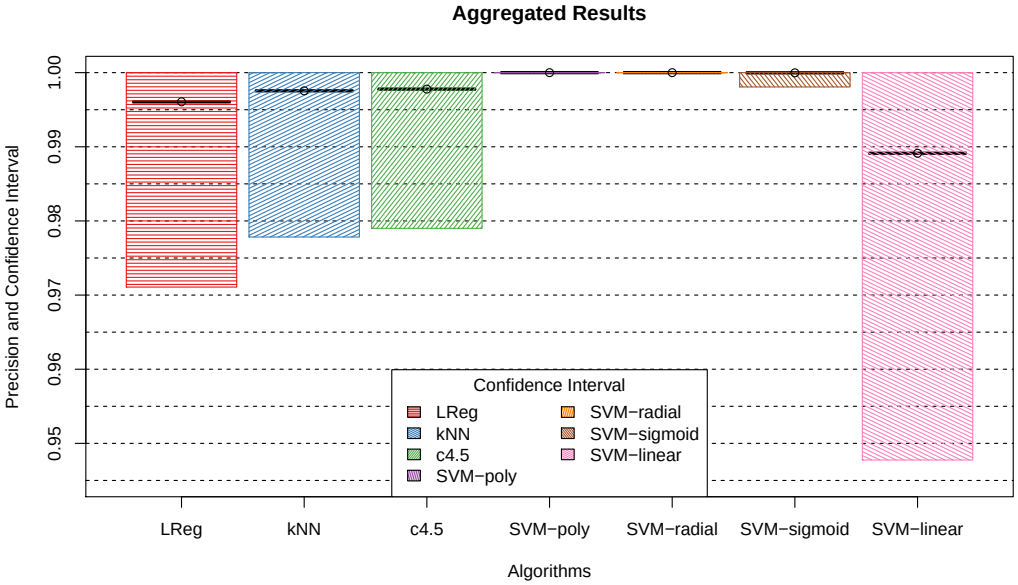


Figure 7.10: Overview of the results with optimized parameters for each ML technique. From left to right the reader sees the logistic regression, kNN, c4.5 decision tree, SVM with polynomial, radial, sigmoid and linear kernel. For details see table D.9 in appendix D.

All in all, it seems that the SVM with the linear kernel has its deficits in almost every task (exception nights). In contrast, the SVM with the radial, polynomial and sigmoid kernel finds all web objects. The kNN's and the c4.5 decision tree's mean precision fluctuate for the different tasks. The logistic regression finds all web objects in 5 out of 6 tasks and in the other task it scores with 99.91% which makes it the fourth best classifier for the accommodation search scenario.

The all Connections Search Scenario

Figure 7.12 illustrates the results of the *all connection searches* scenario. It includes the plots for each task of this scenario. The kNN, c4.5 decision tree as well as the SVM with the polynomial, radial and sigmoid kernel find all web objects for all tasks. These are exceptional results. The SVM with the linear kernel have some problems with the departure date. However, the logistic regression seems to overfit for this scenario. In every task, where more than 18 web pages are involved, it does not find all web objects, but the mean precision is still quite high with above 96% in every task.

The Bus Connection Search Scenario

In figure 7.13, the results for the *bus connection scenario* are illustrated. In the *arrival location* and the *one-way* tasks the mean precision of every ML technique is 100%. However, in the

submit button task, the logistic regression and the SVM with the linear kernel do not find all web objects, while the other techniques do. The *departure date* is found, in every run by all techniques except the SVM with the linear kernel. The task of finding the *departure location* results in a perfect mean precision rate for 6 out of 7 techniques. An exception is the logistic regression with 99.92%.

The Flight Connection Search Scenario

Figure 7.14 graphically illustrates the details for the *flight connection search* scenario. With techniques it is possible to find all web objects for all scenarios, the kNN, the c4.5 decision tree, as well as the SVM with the polynomial, radial and sigmoid kernel. With the logistic regression and the SVM with the linear kernel, the tasks of finding the *arrival location*, *departure date* and the *submit button* leads to a mean precision slightly lower than 100%. Furthermore, the SVM with the linear kernel did not find all web objects for the task of *adult passengers*.

The Train Connection Search Scenario

A detailed illustration for each task for the *train connection search* scenario can be found in figure 7.15. The following ML algorithm managed to discover all searched web objects: the logistic regression, the kNN, the c4.5 decision tree as well as the SVM with the polynomial and radial kernel. The SVM with the sigmoid kernel reaches a mean precision of 99.92% for the *departure location*. However, the SVM with the linear kernel has considerable problems with finding all web objects in all tasks for the train scenario.

7.4 Evaluation Bias

This section discusses the bias which stems from the evaluation setting. The root of the bias is, that not all web pages contain all functional web objects for its scenario. As a consequence, only web objects which exist on the page can be found. The evaluation shown above in this chapter describes the mean precision which results when a present web object is found or not. However, if the classifier would deal with a new unknown web page, how should it know if a certain web object is on this web page or not? To address this problem, further work need to be done which lies beyond the scope of this master thesis. In Kordomatis et al [58] the results for the connection searches assign a mean precision of 0 to those web pages which have a certain web object not on. The real value lies between the results of [58] and the results in this work.

7.5 Feature Discussion

Feature Importance for Classification

This section provides some insights into the importance of the visual features of web objects. The details can be found at table D.2 in appendix D. The importance of different variables or features are derived from the inference statistics of the logistic regression. A skilled data analyst

would suggest that there are a great variety of the different feature importance. This might be true for most applications, however, for this application it is not.

In order to examine the different feature importances, the level of significance (p-value[3]) of each features coefficient were collected for the 43,700 classifier trained for the extended classification benchmarks. This p-value lies between 0 and 1 and the lower the value the more statistically significant this feature is. In most applications, a p-value which is below 5% is in general considered as statistically significant. Since within this master thesis a vast number of classifiers have been trained, the results are aggregated in the following form to give a good overview of the general tendencies. For different values of α (5%, 1% and 0.1%) the numbers how often each feature is significant are counted and then divided by the number of trained classifiers. This resulting value could be interpreted as a percentage of importance. 100% means that the corresponding feature is significant for all trained classifiers, 0% means that it is never significant. The results are very good for every feature. No feature was in less than one of two cases significant. The highest alpha value $\alpha = 0.05$ is 0.7185 or 71.85%. In conclusion it can be said that all features yield a high importance.

However, one point should be added. These results represents the significance of the logistic regression for the distance matrices and not obligatory postprocessing. The results therefore might not be exactly methodically correct, but they do give a good overview of the tendencies. The exact approach would have been rather time consuming and would go widely beyond the scope of this thesis.

Exploratory Data Analysis of the Web Object's Features

In order to get a better feeling of the data the TAMCROW project is dealing with, it is good practice to investigate the different features exploratively. This section provides some plots and figures which will illustrate the features in a very intuitive way. The illustrations are based on the scenarios as introduced in chapter 2 in section 2.1. A complete list of annotated web pages can be found in the appendix A. In total, there were 53 web pages annotated which can be divided into 5 scenarios (4 unique plus 1 which is made up of 3 of the 4). All web pages together contain 9033 web objects whose visual features were computed and stored.

Visualization of the Feature Distributions

This section provides some plots for illustrating the distributions of the web object's features. Depending on the scale it is possible to use different types of plots. For factorial features, histograms are used to illustrate the relative frequencies of their different levels. In contrast, for ratio and interval scaled features, boxplots are used to illustrate their distributions. In addition, these features are standardized[4] and plotted together to make them comparable.

[3]This p-value is the result of a statistical test, where the 0-hypothesis is that no better classification result can be reached when setting the respective coefficient to 0. Therefore, a p-value below a certain alpha (α) means that this hypothesis is refused and it could be assumed that this feature/variable has a significant influence for predicting the class of an observation.

[4]Transformed so they have mean = 0 and a standard deviation = 1

In figure 7.16 the nominal features are graphically illustrated. It consists of six sub graphs displaying the scenarios, object type, editability, select-ability, link type and the dominant orthogonally visible object. The y-axis indicates the different levels whereas the x-axis shows the relative frequency. If the value *[NULL]* appears in the plot, it means that for this element a value is not defined in general or can not be assigned, since it would have no meaning for this type of web object. Detailed results can be found in appendix C in section C.2.

The bar plot for the scenarios shows the distribution of the web objects for each scenario. Almost 40% of the web objects came from the accommodation search scenario, whereas the rest is almost equally divided between the bus, flight and train connection search scenario with around 20%. (Details in table C.5).

The next bar chart shows the relative frequency of all object types. Most of the web objects are textual elements (almost four out of five objects), followed by 13% of objects which were images. Objects of the types *select* and *text input* occur between 2.2% and 2.7%. *Radio buttons* and *submit buttons* are represented with 1.2% and 1.59% in the collected data. (Details in table C.6)

The third graph illustrates the editability of the web objects. It seems natural that most (93%) of the web object are not editable by the user (Details in table C.7).

The fourth plot indicates the possibility that a web object can be selected. This feature is only valid for *radio buttons* and *check boxes*. This is the reason why the number of *[NULL]* values lies above 98%. When the web object is either a radio button or a check box, then it is more likely that it is not selected with a probability of about 65% compared to 35% that it is selected. (Details in table C.8)

The fifth plot shows which is the most dominant orthogonally visible feature of a web object. Since most web objects are of type text, it seems logical that this number has to be high. The bars are in general quite similar to the ones in plot two. However, the *[NULL]* value is new resulting from objects without orthogonally visible objects in their context. (Details in table C.9)

The last plot illustrates the *link type* of the web objects. This mainly depends on the relation to the top web page. (Details in table C.10). The value *[NULL]* appears for web objects that are not a link.

Unfortunately, some feature names are quite long. As a consequence the names have been replaced by the following IDs to make the figures 7.17 and 7.18 more readable: 1) AreaIO 2) AspectRatioIO 3) EmphasisIO 4) FontSizeIO 5) LinesQntIO 6) TokensQntIO 7) SimilarTypesQntROC 8) AlignmentQntROC 9) AlignmentHorQntROC 10) AlignmentIndexHorROC 11) AlignmentVertQntROC 12) AlignmentIndexVertROC 13) AlignmentFactorROC 14) AlignmentVertHorRatioROC 15) OrthogonalVisibleObjQntROC 16) AlignedOrthogonalVisibleObjQntROC 17) FullyAlignedOrthogonalVisibleObjQntROC 18) PixelsToCharacterIC 19) AvgWeightedFGColorROC 20) AvgWeightedBGColorROC 21) RelativeWidthROW 22) RelativeHeightROW 23) RelativeXPositionROW 24) RelativeYPositionROW 25) GridLocationX3ROTW 26) AlignmentQntROD 27) AlignmentHorQntROD 28) AlignmentIndexHorROD 29) AlignmentVertQntROD 30) AlignmentIndexVertROD 31) AlignmentVertHorRatioROD 32) TObjectsQntIC 33) TextSpatialDensityIC 34) LinkSpatialDensityIC 35) LinkCharacterDensityIC .

Figure 7.17 gives a first impression to the distributions of the web object's features, which are ratio or interval scaled. The features have been standardized in order to be comparable. However, as some features have a vast number of outliers, the main part of the data are in a too narrow interval to be further investigated. Therefore, the same data has been replotted in figure 7.18. Here the interval on the y axis from -3 to 3 has been chosen, since in case of normally distributed features 99.73% of the data points would lie in this interval. In the plots, the reader will find 35 boxplots and three lines (two dashed and one solid). There are also 34 black dotted vertical lines, which are just inserted to make the columns more readable. One of this line, the solid one, illustrates the mean of all features (Due to the transformation it is 0 for every feature). The other two red dashed lines illustrate the bandwidth where 50% of the data would lie if the data would be normally distributed. In the next paragraph the different boxplots are discussed. In case the reader is not familiar with boxplots, it might be useful to take a look at the appendix E at section E.1.

The bar of the boxplot represents 50% of the data for its feature (this is also called interquartile range (IQR)). It seems that for almost every feature, the distribution is narrower than it would be in case of normally distributed data. In general, it could be stated that if the median lies below the mean then the data is right skewed. (The opposite is the case, if the median lies above the mean.) However, it has to be mentioned that due to explorative data analysis this estimations are rather fuzzy and not exact. Examples are feature 3 and 24. The first is most certainly right skewed, whereas it is harder to tell if this also applies to the feature with ID 24 (due to the large number of outliers probably yes). What might be interesting to point out, is that the feature with ID 1, 5, 18, 22 and 24 have a very thin IQR. In contrast, the feature with ID 3 and 35 are rather broad. The feature with ID 35 seems to be possibly normally distributed. After further investigations with a histogram it is clear that the data is similar to a uniform distribution. Regarding the outliers, only the features with ID 3 and 35 do not have any outliers in their data. For the other features, this can might happen due to the fact that the IQR is rather small.

Analysis of the linear Correlation between features

This sub-subsection discusses the linear correlation between features. Therefore, two commonly used correlation coefficients have been evaluated for every combination of feature pairs, namely the Pearson and Spearman correlation coefficient. Pearson's version is probably the most common correlation measure. In appendix E in section E.2 the difference and the formulars for computing the coefficients are given.

Figure 7.19 and 7.20 are heat maps where the correlation coefficient is illustrated in color grades. The more the color tends to be white the lower is the correlation of the two features. If the color appears to be red, then the features are negatively correlated. On the contrary, the features are positively correlated, if the heat map shows a blue field. It can be observed that both figures are quite similar. However, the robust version (Spearman) is more interesting since it is more robust against outliers and therefore, it shows information which holds for a greater amount of observations. In the next paragraph the interesting feature pairs are mentioned.

Before talking about the correlations, it has to be mentioned that the heat map is symmetric

in respect to its first diagonal[5]. Therefore, the author is describing the features from the above right point of view. It is not surprising that the *area* of a web object positively correlates with the *relative height* and *relative width*. However, it is true that *area* has a higher positive correlation with the *width* than with the *height*, which lead to the assumption that the *area* depends more on *width* or in other words, wider objects also have a greater area. After discovering this, it is not surprising that the *aspect ratio*[6] has a strong positive correlation with the *area*. It might seems logical that if more objects have the same horizontal alignment[7] the probability that a wide objects appear in this row is rather small. This is probably the reason why the object's *area* negatively correlates with both versions of the feature which hold the *number of horizontal alignments*. What might be expected is that the *area* is positively correlated with the *number of tokens*, since the more words a text has the higher the probability is that it is rather large. It is also intuitively clear that the *number of tokens* is positively correlated to the *number of lines* and to the *relative width* of an web object.

What is rather surprising is that the *number of objects* positively correlates with the *similar types within the context*. This would lead to the interpretation that the more objects there are within a context, the higher the probability that these objects are of the same type. In addition, it might seem unexpected that the *similar types within the context* is negatively correlated with the *pixel to emphasis ratio*.

For the different kind of alignment measures there are numerous linear dependencies which are the logical consequence of their definition. There is for example a strong positively correlation between the different alignments within the context and the corresponding alignment within the document. Therefore, correlations within this group are not discussed in detail.

The *relative y position* is negatively correlated with the *pixel to character ratio*, the *average weighted foreground* and *background color*, which seems rather arbitrary. The *pixel to character ratio* has a strong negative correlation to *text density*, because of the nature of its definition. The *pixel to character ratio* has the area of the context in the numerator, whereas the *text density* has the area of the context in the denominator.

Feature Discussion based on statistical Ratios

The following discussion is based on table C.1 in appendix C. The set of rations (mean, median, quartiles, standard deviation, skewness, kurtosis, minimum and maximum) are mostly referred to as descriptive statistics.

In general, standard normally distributed features have a mean of 0, a standard deviation of 1, a skewness of 0 and a kurtosis of 3. However, a feature is not per se standard normally distributed if these ratios for its sample are received. For a sample, the ratios are close to the corresponding values. A simple normally distributed feature has a different mean and standard deviation (the kurtosis is $3\sigma^4$), but can be transformed by its z-score to be standard normally distributed.

[5]The first diagonal goes from top left to bottom right.

[6]The aspect ratio is above 1 for objects which are wider than high

[7]Horizontal alignment means the objects are in the same horizontal row

A kurtosis of 0 means that the distribution is symmetric, a value below 0 means that the distribution is right skewed and a value above 0 means that it is left skewed. The kurtosis describes the peakedness of a distribution. It depends on the standard deviation σ as the kurtosis $\kappa = 3\sigma^4$. In most cases, when the median is much greater than the mean, then the distribution is left skewed; on the contrary, when the median significantly exceeds the mean, then the distribution is right skewed.

Table C.1 shows, that all features are not normally distributed. This might happens due to the nature of web objects. Almost every feature is right skewed and therefore not symmetric. Since it is difficult to gain an overview of the single features by looking at the descriptive statistics, the author of the master thesis strongly suggests to take look at the sub-section regarding the explorative feature discussion above.

7.6 Conclusion to the detailed Results

The SVM with the polynomial and the radial kernel find all web objects in all tasks of all scenarios. These results are impressive since each algorithm has to find 43,700 web objects. The SVM with the sigmoid kernel founds 43,699 out of 43,700 web objects. The other ML algorithm perform slightly worse, but seem to perform considerably accurate with a mean precision not less than 0.8783 or 87.83% for each scenario and task.

The results of the SVM with the radial kernel function (and the polynomial kernel function[8]) for the different connection search scenarios (bus, flight, train and all together) demonstrate that the knowledge of other subdomains are not necessarily needed, since the results do not differ for the domain specific search connections and the merged scenario.

In addition, it is shown that the visually perceivable representation of the different web objects in the TAMCROW project provides a profound base to achieve extraordinary results. The main advantages of this approach is that it is independent of the source and the language on the web page.

The visual perceivable features themselves do not depend on a single dominant feature to achieve high classification rates. It is the set of features that enables those exceptional results.

However, it is needed to enhance the current approach in order to address the question whether a certain web object is present on a new web page or not. This can be done by sophisticated methods or by introduction of a threshold in the postprocessing step.

[8]The radial kernel function should be prefered over the polynomial kernel function, since it yields the same mean precision with less parameters to estimate.

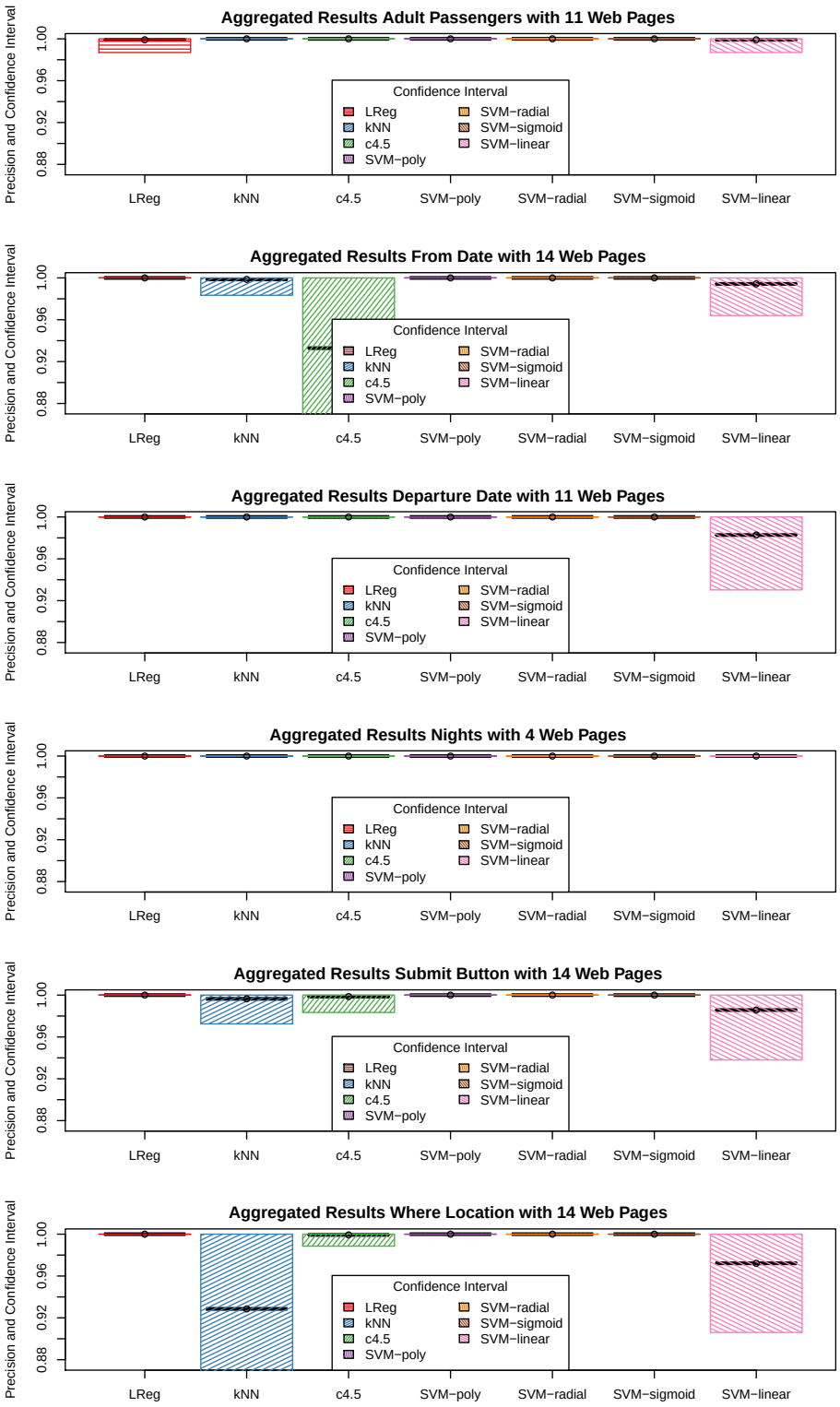


Figure 7.11: Results details after the parameter optimization of each ML technique for the *accommodation search*. From left to right the logistic regression, kNN, c4.5 decision tree, SVM with polynomial, radial, sigmoid and linear kernel. Each plot shows a different task, from top to bottom adult passengers, from date, departure date, nights, submit button and where location. For details see table D.10 in appendix D.

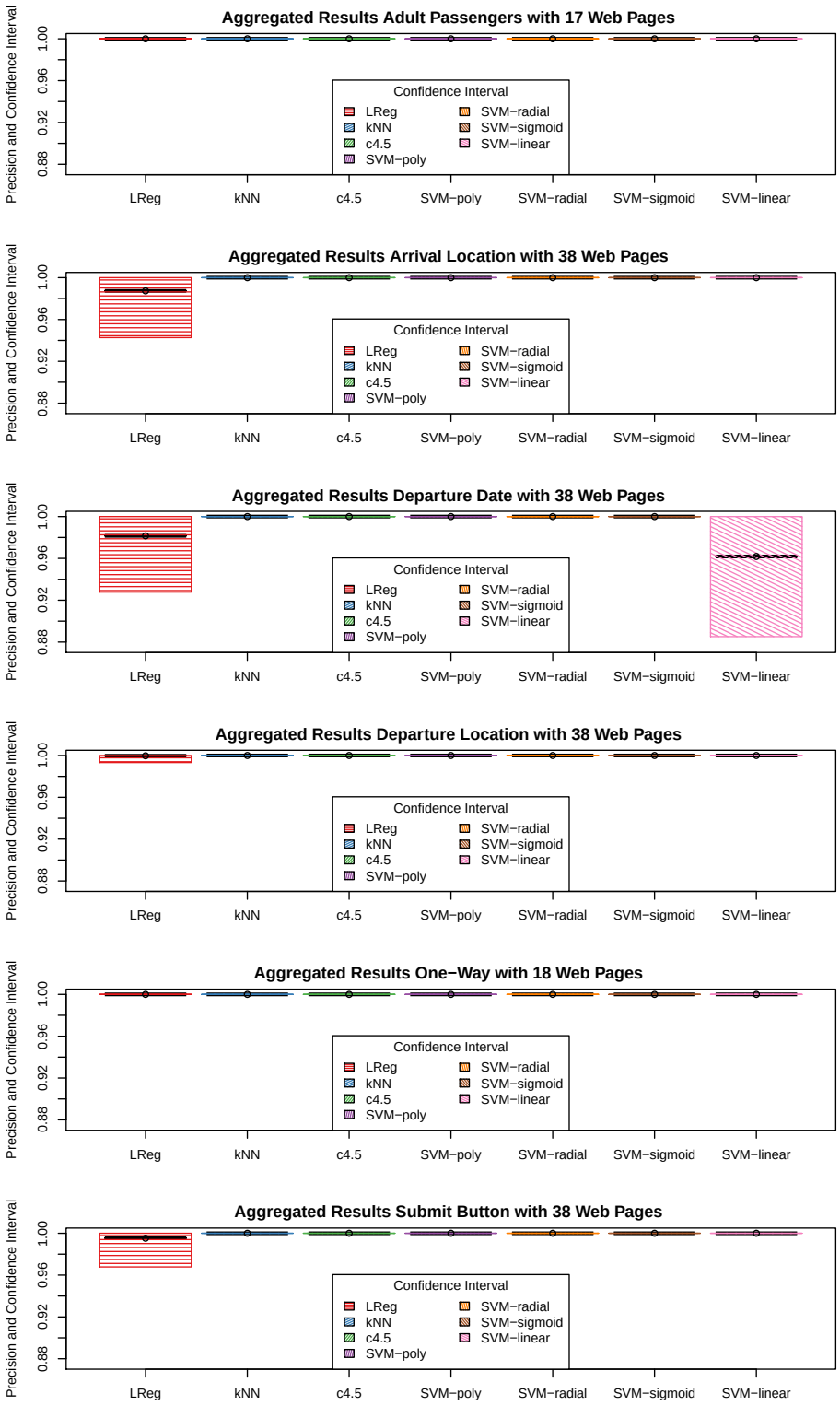


Figure 7.12: Details of the results with optimized parameters for each ML technique for the *all connection searches*. From left to right the reader sees the logistic regression, kNN, c4.5 decision tree, SVM with polynomial, radial, sigmoid and linear kernel. Each plot shows a different task, from top to bottom adult passengers, arrival location, departure date, departure location, one-way, and submit button. For details see table D.10 in appendix D.

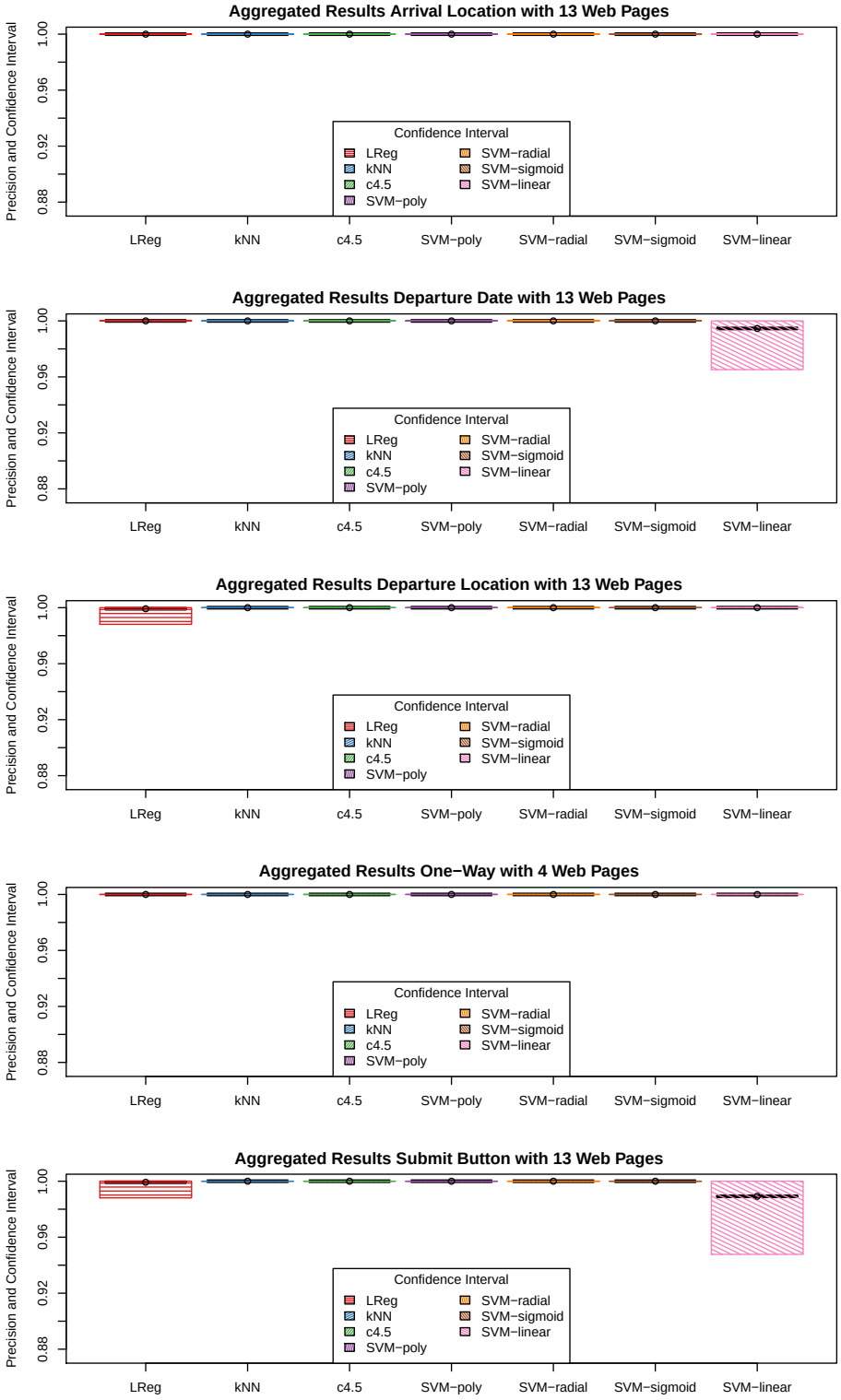


Figure 7.13: Details of the results with optimized parameters for each ML technique for the *bus connection search*. From left to right the reader sees the logistic regression, kNN, c4.5 decision tree, SVM with polynomial, radial, sigmoid and linear kernel. Each plot shows a different task, from top to bottom arrival location, departure date, departure location, one-way, and submit button. For details see table D.10 in appendix D.

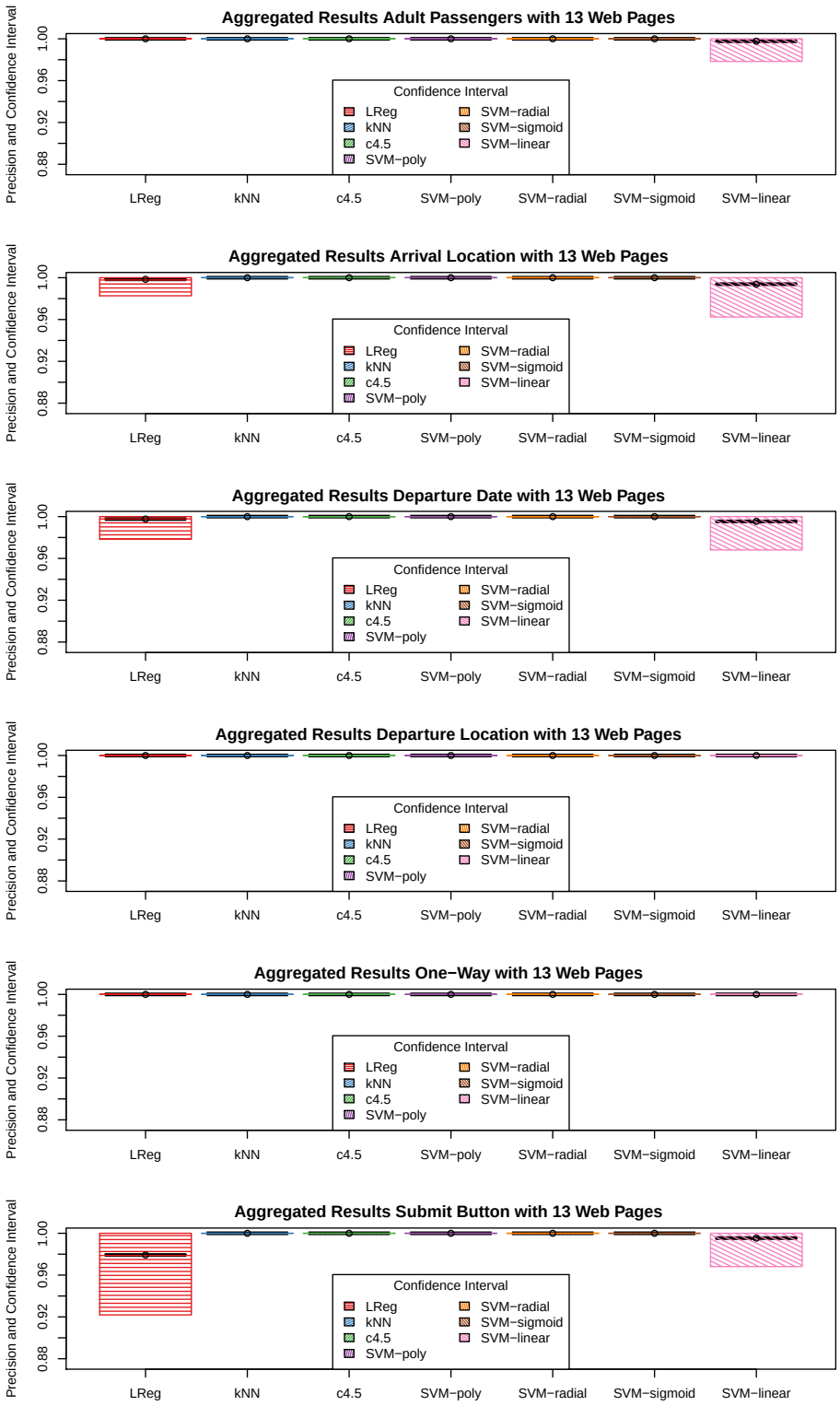


Figure 7.14: Details of the results with optimized parameters for each ML technique for the *flight connection search*. From left to right the reader sees the logistic regression, kNN, c4.5 decision tree, SVM with polynomial, radial, sigmoid and linear kernel. Each plot shows a different task, from top to bottom adult passengers, arrival location, departure date, departure location, one-way, and submit button. For details see table D.10 in appendix D.

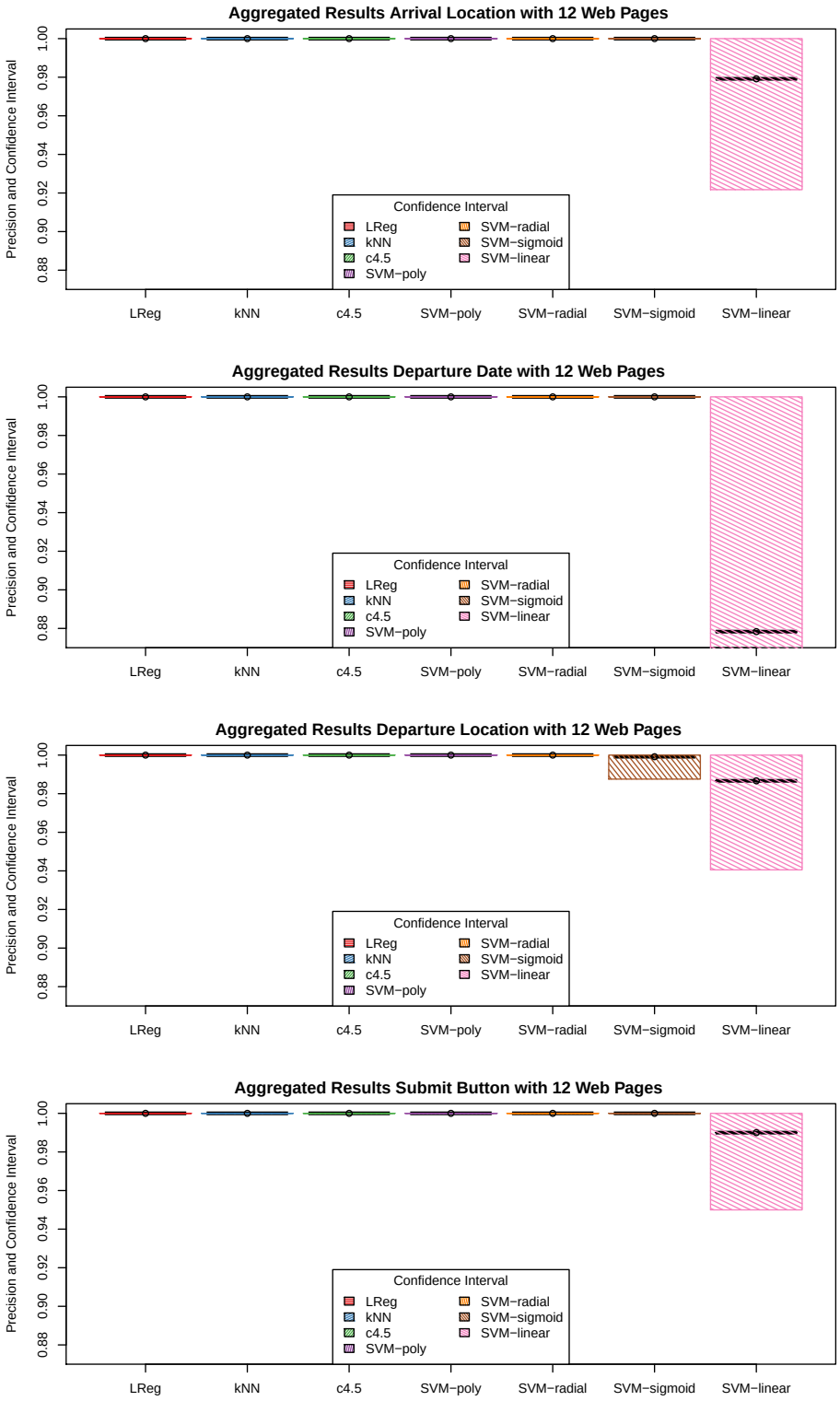


Figure 7.15: Details of the results with optimized parameters for each ML technique for the *train connection search*. From left to right the reader sees the logistic regression, kNN, c4.5 decision tree, SVM with polynomial, radial, sigmoid and linear kernel. Each plot shows a different task, from top to bottom arrival location, departure date, departure location, one-way, and submit button. For details see table D.10 in appendix D.

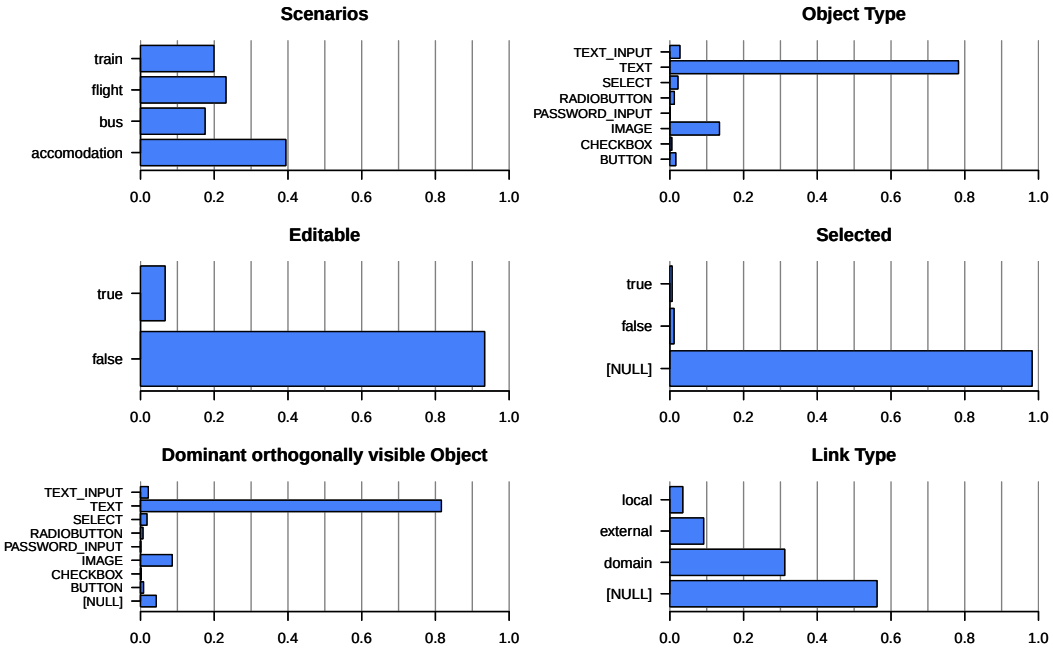


Figure 7.16: Nominal Features as Bar Charts

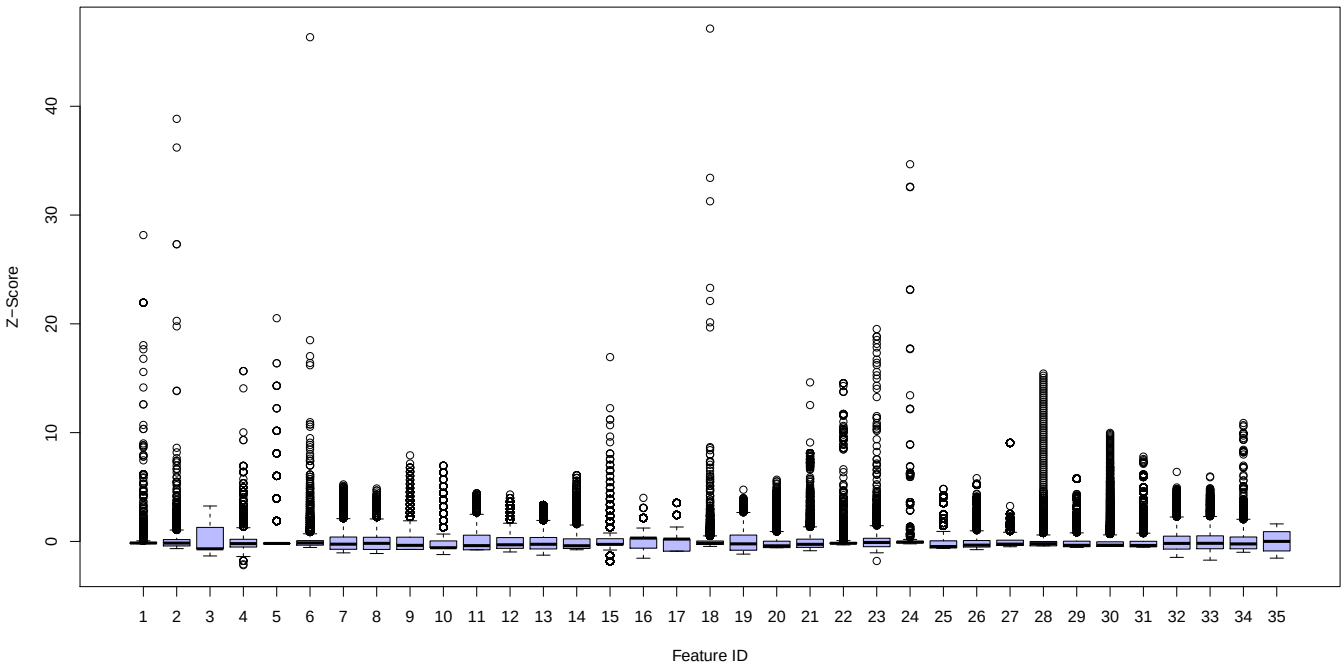


Figure 7.17: Z-Score for numerical Features

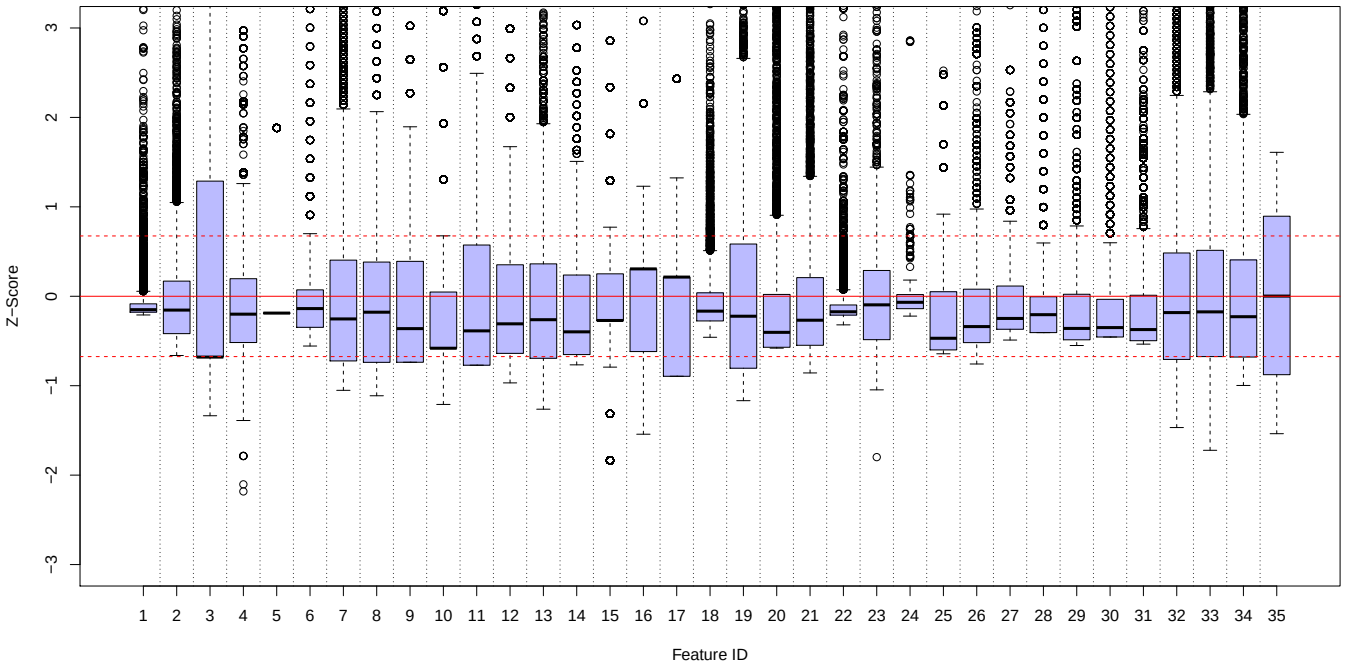


Figure 7.18: Z-Score for numerical Features with zoom

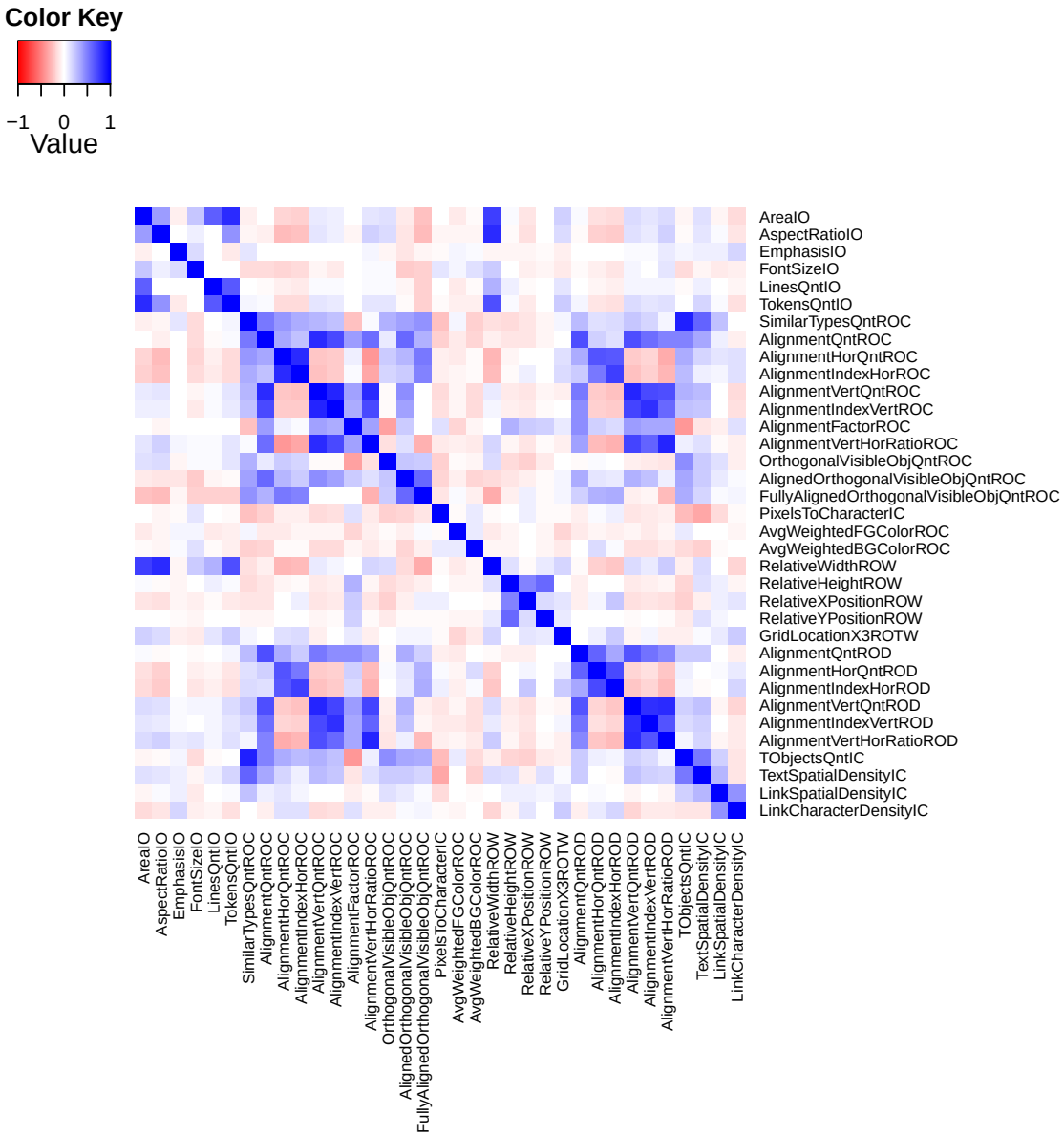


Figure 7.19: Heat map illustrating the Correlation Matrix using Pearson's method

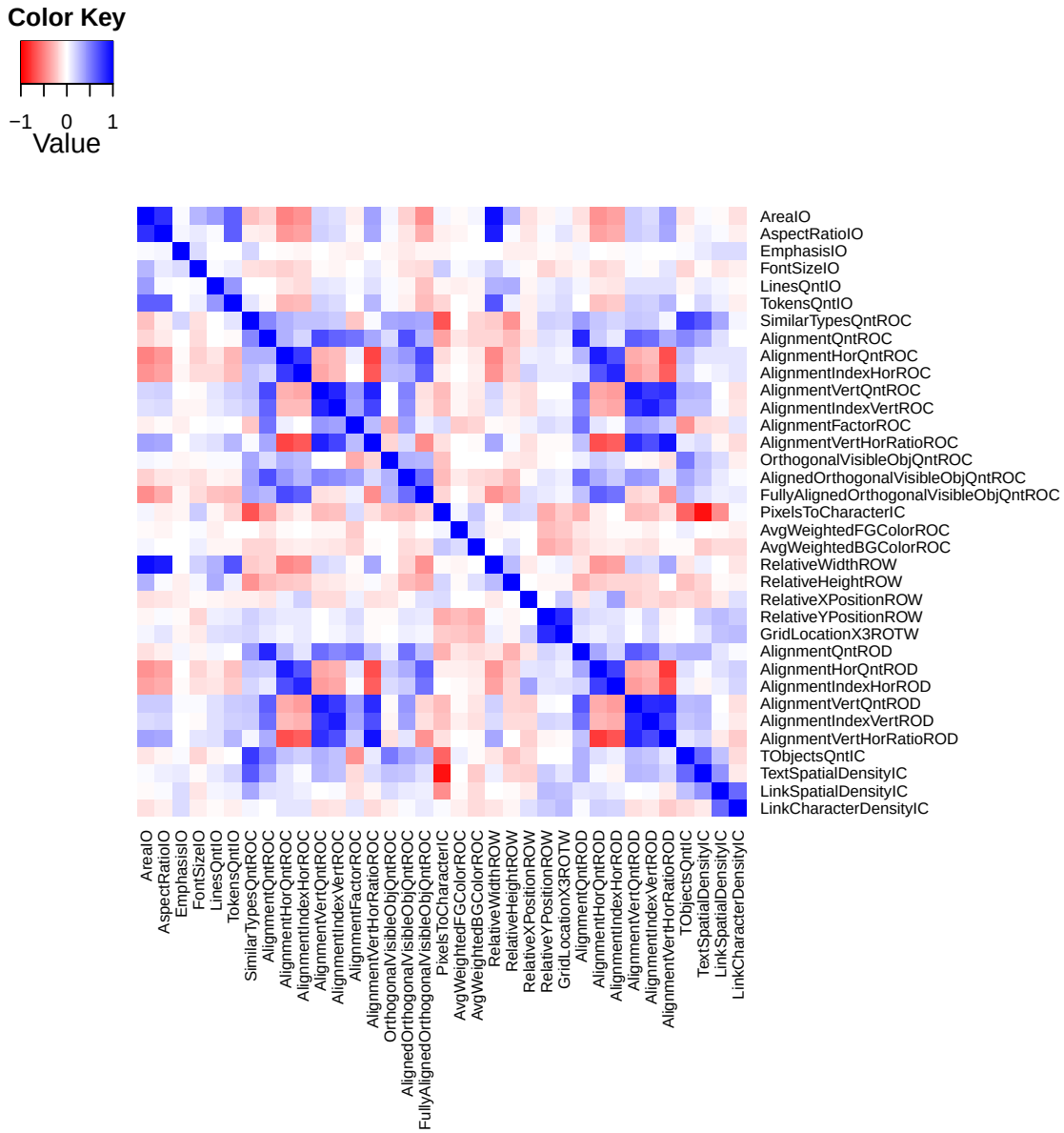


Figure 7.20: Heat map illustrating the Correlation Matrix using Spearman's technique

CHAPTER 8

Conclusion

This master thesis shows how machine learning methods can be used to identify web objects on new web pages by their visual perceivable characteristics. This identification problem is central for different application scenarios and research areas. Prominent examples are meta search engines, support for blind users and web automation.

The proposed approach yields robust results and high precision in identifying web objects. A useful feature of this approach is that it is language independent. Therefore, the number of examples needed is rather small compared with a labeling approach. In addition, it has been shown that not single features are important to achieve a high classification precision; it is the set of features at a whole that matters.

Figure 8.1 gives an overview of the proposed workflow for web object identification. The approach can be divided into a preprocessing step, a classification process and postprocessing procedure. Only the combination of these three steps enables significant results and a robust web object identification.

Figure 8.1: The proposed workflow for identifying web objects on unseen web pages

8.1 Preprocessing

The preprocessing covers every kind of data preparation for the machine learning algorithm and is therefore important. Especially, the distance computation seems to be a useful approach in order to increase the number of postive observations, since the distance computation raise the number of observations exponentially. As a result, the classifiers have more observations to learn from. In addition, the technique gives some features a numerical and therefore a processable meaning (e.g. colors and text). Apart from the distance computation, it is also essential to perform certain types of data transformation to prevent that the classifiers are driven by outliers. One of them is the relative rank transformation which is of utmost importance for the support vector machines (SVM).

8.2 Machine Learning Algorithms

The best classifiers clearly are the SVM with the polynomial and the radial kernel function. Since the latter needs fewer parameters to estimate, it is more favorable for this kind of problem. However, it seems that compared with the preprocessing and postprocessing, it is not so important which classifier is used, since the worst achieved a 98.913 % precision (the SVM with the linear kernel function).

8.3 Postprocessing

The postprocessing becomes obligatory since the classification results are performed on the distance matrix where the observations are the pairs of the original web objects. Therefore, the sum of each unique web object has to be computed in order to choose the most probable web object (that unique web objects with the highest appearance within the positively classified web object pairs). Theoretically, a classifier can at least misclassify n pairs without any changes in the classification results (n represents the difference of the unique web object between the first and second place). Therefore, this postprocessing makes the results so stable and remarkable.

8.4 Summary of Achievements

The following list summarizes the main achievements of this master thesis:

- The workflow how web objects can be identified with machine learning techniques by using visually perceivable features is demonstrated.
- The results are robust for different domains without using any domain specific knowledge.
- The limitation to visually perceivable features enables a language independent solution compared to text-based approaches.
- A thorough evaluation and analysis shows that the chosen approach yields outstanding results.

- The postprocessing step after classifying the web object pairs yields a robust method for identifying a specific web object on a new web page.
- A set of 49 characteristic features structured into interface, spatial, visual and textual features is elaborated as a joint work together with the TAMCROW team.
- The importance of different features has been evaluated.
- A prototype for visual feature learning is implemented in R and used in the postprocessing step of the feature evaluation workflow.
- The collection of the accommodation scenario has been added to the evaluation corpus which consists of 15 web pages.
- Statistical analyses regarding the visually perceivable features of the web objects have been computed and investigated.
- Explorative data analyses on these features have been performed.
- Definition of the feature distance metric *relative distance* has been adapted together with the TAMCROW team.

APPENDIX A

Annotated Web Pages

This chapter provides a list of annotated web pages which were used to evaluate the classification methods described in chapter 5. Annotated in this context means, that the web objects on the different web pages were marked with an additional attribute. This attribute differs from scenario to scenario. In general the following list of attributes were used for the following scenarios.

- ***Flight-, Train-, Bus- and All Connection*** **Search**: Departure Location, Arrival Location, Departure Date, Number of adult passangers, One-Way trip and Submit Button
- ***Accommodation Search***: Location (Where), Arrival Date, Departure Date, Number of Nights, Number of People and Submit Button

The following four tables give an overview about the used web pages for testing the Machine Learning (ML) algorithm within this master thesis (see tables A.1 for the flight connection search, A.2 for the train connection search, A.3 for the bus connection search and A.4 for accommodations search). The *all* scenario consists of the web pages for the flight, bus and train connection searches.

[1] www.britishairways.com/travel/home/public/en_at
[2] www.aa.com/homePage.do?locale=en_US&pref=true
[3] www.emirates.com/at/english/index.aspx
[4] www.qantas.com.au/travel/airlines/home/au/en
[5] book.austrian.com/app/fb.fly?pos=AT&l=de
[6] www.checkfelix.com/flugsuche/de/fluege.html?sourcedomain=at
[7] www.lufthansa.com/online/portal/lh/de/booking
[8] www.airberlin.com/site/start.php?LANG=deu&all=1&MARKT=DE
[9] www.aeroflot.ru/cms/ru/booking
[10] www.rossiya-airlines.ru/ru/tickets/zabronirovati_buy
[11] www.tatarstan.aero/tickets/buy
[12] www.trip.ru
[13] www.aviasales.ru
[14] www.trenitalia.com/cms/v/index.jsp?vgnextoid=ad1ce14114bc9110VgnVCM10000080a3e90aRCRD

Table A.1: List of Web Pages for the Flight Connections Scenario

Web Page	From	To	Departure Date	One-way Trip	Adult Passengers	Search Button	Language
British Airways[1]	+	+	+	+	+	+	en
American Airlines[2]	+	+	+	+	+	+	en
Emirates[3]	+	+	+	+	+	+	en
Quantas[4]	+	+	+	+	+	+	en
Austrian Airlines[5]	+	+	+	+	+	+	de
Checkfelix[6]	+	+	+	+	+	+	de
Lufthansa[7]	+	+	+	+	+	+	de
Air Berlin[8]	+	+	+	+	+	+	de
Aeroflot[9]	+	+	+	+	+	+	ru
Rossiya Airlines[10]	+	+	+	+	+	+	ru
Tatarstan[11]	+	+	+	+	+	+	ru
Trip[12]	+	+	+	+	+	+	ru
Avia Sales[13]	+	+	+	+	+	+	ru

Table A.2: List of Web Pages for the Train Connections Scenario

Web Page	From	To	Departure Date	One-way Trip	Adult Passengers	Search Button	Language
Trenitalia[14]	+	+	+	-	-*	+	en
Eurostar[15]	+	+	+	-	+	+	en
TGV Europe[16]	+	+	+	-	-	+	en
Irishrail[17]	+	+	+	+	-	+	en
ÖBB (Austrian Rail)[18]	+	+	+	-	-	+	de
Deutsche Bahn[19]	+	+	+	-	-	+	de
SBB (Swiss Rail)[20]	+	+	+	-	-	+	de
Saarbahn[21]	+	+	+	-	-	+	de
Tutu[22]	+	+	+	-	-	+	ru
Poezdato[23]	+	+	+	-	-	+	ru
Portal Poisk[24]	+	+	+	-	+	+	ru
RZD (Russian Rail)[25]	+	+	+	-	-	+	ru

Table A.3: List of Web Pages for the Bus Connections Scenario

Web Page	From	To	Departure Date	One-way Trip	Adult Passengers	Search Button	Language
matkahuolto[26]	+	+	+	+	-	+	en
postbus.ch[27]	+	+	+	-	-	+	en
bus eireann[28]	+	+	+	-	-	+	en
goto bus[29]	+	+	+	+	+	+	en
eurolines[30]	+	+	+	-	-	+	de
postbus.at[31]	+	+	+	-	-	+	de
berlin linienbus[32]	+	+	+	+	-	+	de
public express[33]	+	+	+	-	+	+	de
avtovokzal[34]	+	+	+	-	-	+	ru
marshruty[35]	+	+	+	+	-	+	ru
turistua[36]	+	+	+	-	-	+	ru
autovokzal[37]	+	+	+	-	-	+	ru
avpem[38]	+	+	+	-	-	+	ru

[15] www.eurostar.com
[16] www.tgv-europe.com/en
[17] www.irishrail.ie
[18] www.oebb.at
[19] www.bahn.de/p/view/index.shtml
[20] www.sbb.ch/home.html
[21] www.saarbahn.de/de/start
[22] www.tutu.ru/poezda
[23] poezdato.net
[24] poezda.portal-poisk.ru
[25] pass.rzd.ru
[26] http://www.matkahuolto.fi/en
[27] http://www.postbus.ch
[28] http://www.buseireann.ie
[29] http://www.gotobus.com
[30] http://195.110.209.27/ticketshop/DesktopDefault.aspx
[31] http://www.postbus.at/de
[32] http://www.berlinlinienbus.de/index.php
[33] http://www.publicexpress.de/buy-online
[34] http://www.avtovokzal.ru
[35] http://transport.marshruty.ru
[36] http://ticket.turistua.com/ru/bus
[37] http://www.autovokzal73.ru
[38] http://www.avperm.ru
[39] http://www.agoda.com
[40] http://www.booking.com
[41] http://www.easyclicktravel.com
[42] http://www.getaroom.com

Table A.4: List of Web Pages for the Accommodation Search Scenario

Web Page	From Date	To Date	Where	Nights	Adult Passengers	Search Button	Language
Agoda[39]	+	-	+	+	-	+	en
Booking.com[40]	+	+	+	-	+	+	en
Easy Click Travel[41]	+	+	+	-	-	+	en
Get a room[42]	+	+	+	-	+	+	en
Hostels Club[43]	+	-	+	+	-	+	en
Hostels.com[44]	+	+	+	-	+	+	en
HRS[45]	+	+	+	-	+	+	en
olotels.com[46]	+	-	+	+	+	+	en
Otel.com[47]	+	+	+	-	+	+	en
Prestigia[48]	+	+	+	-	-	+	en
Priceline[49]	+	+	+	-	-	+	en
Venere.com[50]	+	+	+	-	+	+	en
VivaStay[51]	+	+	+	-	+	+	en
trivago.com[52]	+	+	+	-	+	+	en
Hostelworld.com[53]	+	-	+	+	+	+	en

[43] http://www.hostelsclub.com
[44] http://www.hotels.com
[45] http://www.hrs.com/web3
[46] http://www.olotels.com
[47] http://www.otel.com
[48] http://www.prestigia.com/en
[49] http://www.priceline.com
[50] http://www.venere.com
[51] http://www.vivastay.com/?lang=en
[52] http://www.trivago.com/?source=US
[53] http://www.hostelworld.com

APPENDIX B

Data Sets

B.1 Iris Data Set from Anderson

The data set provided in table B.1 is published by Anderson in [6].

Table B.1: Iris Data Set from Anderson

Setosa				Versicolor				Virginica			
Sepal Length	Sepal Width	Petal Length	Petal Width	Sepal Length	Sepal Width	Petal Length	Petal Width	Sepal Length	Sepal Width	Petal Length	Petal Width
5.1	3.5	1.4	0.2	7	3.2	4.7	1.4	6.3	3.3	6	2.5
4.9	3	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4	1.3	6.3	2.9	5.6	1.8
5	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3	5.8	2.2
5.4	3.9	1.7	0.4	5.7	2.8	4.5	1.3	7.6	3	6.6	2.1
4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
5	3.4	1.5	0.2	4.9	2.4	3.3	1	7.3	2.9	6.3	1.8
4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8
4.9	3.1	1.5	0.1	5.2	2.7	3.9	1.4	7.2	3.6	6.1	2.5
5.4	3.7	1.5	0.2	5	2	3.5	1	6.5	3.2	5.1	2
4.8	3.4	1.6	0.2	5.9	3	4.2	1.5	6.4	2.7	5.3	1.9
4.8	3	1.4	0.1	6	2.2	4	1	6.8	3	5.5	2.1
4.3	3	1.1	0.1	6.1	2.9	4.7	1.4	5.7	2.5	5	2
5.8	4	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4
5.7	4.4	1.5	0.4	6.7	3.1	4.4	1.4	6.4	3.2	5.3	2.3
5.4	3.9	1.3	0.4	5.6	3	4.5	1.5	6.5	3	5.5	1.8
5.1	3.5	1.4	0.3	5.8	2.7	4.1	1	7.7	3.8	6.7	2.2
5.7	3.8	1.7	0.3	6.2	2.2	4.5	1.5	7.7	2.6	6.9	2.3
5.1	3.8	1.5	0.3	5.6	2.5	3.9	1.1	6	2.2	5	1.5
5.4	3.4	1.7	0.2	5.9	3.2	4.8	1.8	6.9	3.2	5.7	2.3
5.1	3.7	1.5	0.4	6.1	2.8	4	1.3	5.6	2.8	4.9	2
4.6	3.6	1	0.2	6.3	2.5	4.9	1.5	7.7	2.8	6.7	2
5.1	3.3	1.7	0.5	6.1	2.8	4.7	1.2	6.3	2.7	4.9	1.8
4.8	3.4	1.9	0.2	6.4	2.9	4.3	1.3	6.7	3.3	5.7	2.1
5	3	1.6	0.2	6.6	3	4.4	1.4	7.2	3.2	6	1.8
5	3.4	1.6	0.4	6.8	2.8	4.8	1.4	6.2	2.8	4.8	1.8
5.2	3.5	1.5	0.2	6.7	3	5	1.7	6.1	3	4.9	1.8
5.2	3.4	1.4	0.2	6	2.9	4.5	1.5	6.4	2.8	5.6	2.1
4.7	3.2	1.6	0.2	5.7	2.6	3.5	1	7.2	3	5.8	1.6
4.8	3.1	1.6	0.2	5.5	2.4	3.8	1.1	7.4	2.8	6.1	1.9
5.4	3.4	1.5	0.4	5.5	2.4	3.7	1	7.9	3.8	6.4	2
5.2	4.1	1.5	0.1	5.8	2.7	3.9	1.2	6.4	2.8	5.6	2.2
5.5	4.2	1.4	0.2	6	2.7	5.1	1.6	6.3	2.8	5.1	1.5
4.9	3.1	1.5	0.2	5.4	3	4.5	1.5	6.1	2.6	5.6	1.4
5	3.2	1.2	0.2	6	3.4	4.5	1.6	7.7	3	6.1	2.3
5.5	3.5	1.3	0.2	6.7	3.1	4.7	1.5	6.3	3.4	5.6	2.4
4.9	3.6	1.4	0.1	6.3	2.3	4.4	1.3	6.4	3.1	5.5	1.8
4.4	3	1.3	0.2	5.6	3	4.1	1.3	6	3	4.8	1.8
5.1	3.4	1.5	0.2	5.5	2.5	4	1.3	6.9	3.1	5.4	2.1
5	3.5	1.3	0.3	5.5	2.6	4.4	1.2	6.7	3.1	5.6	2.4
4.5	2.3	1.3	0.3	6.1	3	4.6	1.4	6.9	3.1	5.1	2.3
4.4	3.2	1.3	0.2	5.8	2.6	4	1.2	5.8	2.7	5.1	1.9
5	3.5	1.6	0.6	5	2.3	3.3	1	6.8	3.2	5.9	2.3
5.1	3.8	1.9	0.4	5.6	2.7	4.2	1.3	6.7	3.3	5.7	2.5
4.8	3	1.4	0.3	5.7	3	4.2	1.2	6.7	3	5.2	2.3
5.1	3.8	1.6	0.2	5.7	2.9	4.2	1.3	6.3	2.5	5	1.9
4.6	3.2	1.4	0.2	6.2	2.9	4.3	1.3	6.5	3	5.2	2
5.3	3.7	1.5	0.2	5.1	2.5	3	1.1	6.2	3.4	5.4	2.3
5	3.3	1.4	0.2	5.7	2.8	4.1	1.3	5.9	3	5.1	1.8

APPENDIX C

Detailed Features

C.1 Details of the Web Object's Feature

This section provides futhrer information for the web object features which have been introduced in chapter 3 in subsection 3.3.

In order to keep the following tables more readable the following abbreviations have been applied to the texts in the table.

- : **#:** Number
- **vert.:** vertical
- **hor.:** horizontal
- **orth.:** orthogonally

$$AreaIO = width_{object} * height_{object} \tag{C.1}$$

$$AspectRatioIO = \frac{width_{object}}{height_{object}} \tag{C.2}$$

$$AlignmentQntROC = \frac{AlignmentHorQntROC}{AlignmentVertROC} \tag{C.3}$$

$$AlignmentFactorROC = \frac{(AlignmentQntROC + 1)}{(TObjectsQntIC + 1)} \tag{C.4}$$

$$AlignmentVertHorRatioROC = \frac{(AlignmentVertQntROC + 1)}{(AlignmentHorQntROC + 1)} \tag{C.5}$$

$$\frac{\sum_{o \in c(s)} \Delta HSV(color(s), color(o)) \cdot area(o)}{\sum_{o \in c(s)} area(o)} \tag{C.6}$$

in the equation ΔHSV is the HSV color distance among the selected object s and an arbitrary object o within the context of s. The variable $c(s)$ stands for the set of all web objects within the context of the web object s. The function $area(o)$ returns the error of the object o.

$$AlignmentQntROD = AlignmentHorQntROD + AlignmentVertQntROD \tag{C.7}$$

$$AlignmentVertHorRatioROD = \frac{(AlignmentVertQntROD + 1)}{(AlignmentHorQntROD + 1)} \tag{C.8}$$

$$\frac{(w_n + s_n + d_n)}{3} \tag{C.9}$$

where w_n, s_n and d_n are computed as following:

$$w_n = \begin{cases} 1 - (400 - w)/300 & \text{if } w \leq 400, \\ (w - 400)/300 + 1 & \text{if } w \geq 400; \end{cases}$$

where w is a font weight, $w \in \{100, 200, \ldots, 900\}$; 400 corresponds to the normal text without "weight", 700 is a *bold* text.

$$s_n = \begin{cases} 1 & \text{if } s = normal, \\ 2 & \text{otherwise;} \end{cases}$$

where s_n is a font style.

$$d_n = \begin{cases} 1 & \text{if } d = none, \\ 2 & \text{otherwise;} \end{cases}$$

where d_n is a text decoration.

This section also provides the following tables C.1, C.2, C.3 and C.4

C.2 Details of the nominal Web Object's Features

The following table are provided within this section C.5, C.6, C.7, C.8, C.9 and C.10.

C.3 Descriptive Statistics for Web Object's Features

Figure C.1 provides the descriptive statistics for the web object features for the all connection scenario.

Table C.1: Details for Interface Features

Name	Scale	Range	Calculation	HtmlButton	HtmlCheckbox	HtmlFileUpload	HtmlImage	HtmlPasswordInput	HtmlRadiobutton	HtmlSelect	HtmlText	HtmlTextArea	HtmlTextInput	Relative to
Object Type	nominal	Domain		x	x	x	x	x	x	x	x	x	x	
Editable	nominal	$x \in \{true, false\}$		x	x	x	x	x	x	x	x	x	x	
Selection	nominal	$x \in \{true, false\}$			x				x					
Dominant orthogonally visible object	nominal	Domain		x	x	x	x	x	x	x	x	x	x	context
Similar Types within the context	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	context
Link Type	nominal	$x \in \{local, domain, external\}$		x	x	x	x	x	x	x	x	x	x	page
# of Objects	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	context

Table C.2: Details for Spatial Features

Name	Scale	Range	Calculation	HtmlButton	HtmlCheckbox	HtmlFileUpload	HtmlImage	HtmlPasswordInput	HtmlRadiobutton	HtmlSelect	HtmlText	HtmlTextArea	HtmlTextInput	Relative to
Area	ratio	$x \in \mathbb{R}_{\geq 0}$	(C.1)	x	x	x	x	x	x	x	x	x	x	
Aspect Ratio	ratio	$x \in \mathbb{R}_{\geq 0}$	(C.2)	x	x	x	x	x	x	x	x	x	x	
# of Alignments	ratio	$x \in \mathbb{N}_0$	(C.7)	x	x	x	x	x	x	x	x	x	x	context
# of horizontal Alignments	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	context
# of vertical Alignments	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	context
Horizontal Alignment Index	nominal	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	context
Vertical Alignment Index	nominal	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	context
Alignment Factor	ratio	$x \in \mathbb{R}_{\geq 0}$	(C.4)	x	x	x	x	x	x	x	x	x	x	context
Ratio of vertical to horizontal Alignments	ratio	$x \in \mathbb{R}_{\geq 0}$	(C.5) and (C.8)	x	x	x	x	x	x	x	x	x	x	context
# of orthogonally visible objects	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	context
# of orthogonally visible aligned objects	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	context
# of orthogonally visible fully aligned objects	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	context
Pixel to character ratio	ratio	$x \in \mathbb{N}_0$									x			context
Relative Width	ratio	$x \in \mathbb{R}_{\geq 0}$		x	x	x	x	x	x	x	x	x	x	page
Relative Height	ratio	$x \in \mathbb{R}_{\geq 0}$		x	x	x	x	x	x	x	x	x	x	page
Relative x-position	ratio	$x \in \mathbb{R}_{\geq 0}$		x	x	x	x	x	x	x	x	x	x	page
Relative y-position	ratio	$x \in \mathbb{R}_{\geq 0}$		x	x	x	x	x	x	x	x	x	x	page
3x3-Grid Locations	nominal	$\{x \in \mathbb{N}_0 \vert x \leq 511\}$		x	x	x	x	x	x	x	x	x	x	page
# of Alignments	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	document
# of horizontal Alignments	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	document
# of vertical Alignments	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	document
Horizontal Alignment Index	nominal	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	document
Vertical Alignment Index	nominal	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	document
Ratio of vertical to horizontal Alignments	ratio	$x \in \mathbb{R}_{\geq 0}$		x	x	x	x	x	x	x	x	x	x	document
Text Density	ratio	$\{x \in \mathbb{R} \vert 0 \leq x \leq 1\}$		x	x	x	x	x	x	x	x	x	x	context
Link Density	ratio	$\{x \in \mathbb{R} \vert 0 \leq x \leq 1\}$		x	x	x	x	x	x	x	x	x	x	context

Table C.3: Details for Visual Features

Name	Scale	Range	Calculation	HtmlButton	HtmlCheckbox	HtmlFileUpload	HtmlImage	HtmlPasswordInput	HtmlRadiobutton	HtmlSelect	HtmlText	HtmlTextArea	HtmlTextInput	Relative to
Foreground Color	nominal	RGBA-Value		x		x				x	x	x	x	
Background Color	nominal	RGBA-Value		x	x	x	x	x	x	x	x	x	x	
Emphasis	ratio	$x \in \mathbb{R}_{\geq 0}$	(w+s+d)/3	x		x				x	x	x	x	
Font Size	ratio	$x \in \mathbb{R}_{\geq 0}$		x		x				x	x	x	x	
Avg. weighted Foreground Color Distance	ratio	$x \in \mathbb{R}_{\geq 0}$	(C.6)	x		x				x	x	x	x	context
Avg. weighted Background Color Distance	ratio	$x \in \mathbb{R}_{\geq 0}$		x	x	x	x	x	x	x	x	x	x	context

Table C.4: Details for Textual Features

Name	Scale	Range	Calculation	HtmlButton	HtmlCheckbox	HtmlFileUpload	HtmlImage	HtmlPasswordInput	HtmlRadiobutton	HtmlSelect	HtmlText	HtmlTextArea	HtmlTextInput	Relative to
Text of Object	nominal	any String		x		x	x			x	x	x	x	
Number of Lines	ratio	$x \in \mathbb{N}_0$	(C.7) and (C.7)	x	x	x	x	x	x	x	x	x	x	
Number of Tokens	ratio	$x \in \mathbb{N}_0$		x	x	x	x	x	x	x	x	x	x	
Text Above	nominal	any String		x		x				x	x	x	x	context
Text Right	nominal	any String		x		x				x	x	x	x	context
Text Below	nominal	any String		x		x				x	x	x	x	context
Text Left	nominal	any String		x		x				x	x	x	x	context
Text of the nearest orthogonally visible object	nominal	any String		x		x				x	x	x	x	context
Text of the nearest object	nominal	any String		x		x				x	x	x	x	context
Character Density of Links	ratio	$x \in \mathbb{R}_{\geq 0}$		x	x	x	x	x	x	x	x	x	x	

Table C.5: Web Objects per Scenario

Type	accomodation	bus	flight	train
absolute frequency	3,556	1,582	2,094	1,801
relative frequency	39.37%	17.51%	23.18%	19.94%

Table C.6: Web Objects per Object Type

type	BUTTON	CHECKBOX	IMAGE	PASSWORD_INPUT	RADIOBUTTON	SELECT	TEXT	TEXT_INPUT
absolute frequency	144	47	1,216	3	110	196	7,075	242
relative frequency	1.59%	0.52%	13.46%	0.03%	1.22%	2.17%	78.32%	2.68%

Table C.7: Web Objects per Editability

type	false	true
absolute frequency	8,435	598
relative frequency	93.38%	6.62%

Table C.8: Web Objects per Selectability

type	[NULL]	false	true
absolute frequency	8,876	102	55
relative frequency	98.26%	1.13%	0.61%

Table C.9: Web Objects per the the dominant orthogonally visible object

type	[NULL]	BUTTON	CHECKBOX	IMAGE	PASSWORD_INPUT	RADIOBUTTON	SELECT	TEXT	TEXT_INPUT
absolute frequency	385	73	18	775	2	59	157	7,373	191
relative frequency	4.26%	0.81%	0.20%	8.58%	0.02%	0.65%	1.74%	81.62%	2.11%

Figure C.1: Descriptive Statistics for the Web Object's Features for the all connection search scenario

Feature Name	Mean	StdDev	Skewness	Kurtosis	Min	25% Quant	Median	75% Quant	Max
ArealO	3,664.49	17,520.11	16.46	332.57	1.00	544.00	1,036.00	2,184.00	497,200.00
AspectRatioIO	5.04	7.59	19.58	611.01	0.01	1.87	3.87	6.33	300.00
EmphasisIO	1.12	0.17	1.01	-0.17	0.89	1.00	1.00	1.33	1.67
FontSizeIO	12.50	2.52	4.52	40.70	7.00	11.20	12.00	13.00	52.00
LinesQntIO	1.09	0.48	8.26	92.68	1.00	1.00	1.00	1.00	11.00
TokensQntIO	2.66	4.77	18.27	664.17	0.00	1.00	2.00	3.00	224.00
SimilarTypesQntROC	22.38	21.29	1.58	2.90	0.00	7.00	17.00	31.00	134.00
AlignmentQntROC	5.95	5.35	1.39	2.00	0.00	2.00	5.00	8.00	32.00
AlignmentHorQntROC	1.96	2.66	2.59	9.21	0.00	0.00	1.00	3.00	23.00
AlignmentIndexHorROC	0.92	1.59	2.50	8.60	-1.00	0.00	0.00	1.00	12.00
AlignmentVertQntROC	4.02	5.21	1.70	2.82	0.00	0.00	2.00	7.00	27.00
AlignmentIndexVertROC	1.93	3.03	1.82	2.85	-1.00	0.00	1.00	3.00	15.00
AlignmentFactorROC	0.28	0.22	1.66	2.84	0.01	0.13	0.22	0.36	1.00
AlignmentVertHorRatioROC	3.07	3.94	2.23	5.83	0.05	0.50	1.50	4.00	27.00
OrthogonalVisibleObjQntROC	3.52	1.92	2.80	25.22	0.00	3.00	3.00	4.00	36.00
AlignedOrthogonalVisibleObjQntROC	1.67	1.08	0.17	-0.49	0.00	1.00	2.00	2.00	6.00
FullyAlignedOrthogonalVisibleObjQntROC	0.81	0.90	0.81	-0.07	0.00	0.00	1.00	1.00	4.00
PixelsToCharacterIC	1,348.46	2,629.68	25.15	911.25	141.86	622.10	911.15	1,449.72	125,343.00
AvgWeightedFGColorROC	0.35	0.30	0.99	0.63	0.00	0.11	0.29	0.53	1.80
AvgWeightedBGColorROC	0.11	0.18	2.62	6.87	0.00	0.00	0.03	0.11	1.15
RelativeWidthROW	0.10	0.11	4.21	28.12	0.00	0.04	0.07	0.12	1.72
RelativeHeightROW	0.02	0.07	9.61	104.47	0.00	0.01	0.01	0.02	1.00
RelativeXPositionROW	0.47	0.66	10.63	161.67	-0.71	0.15	0.41	0.66	13.34
RelativeYPositionROW	0.72	3.36	23.86	676.84	-0.03	0.25	0.49	0.78	117.19
GridLocationX3ROTW	59.28	92.20	2.16	4.34	0.00	4.00	16.00	64.00	504.00
AlignmentQntROD	12.67	16.73	3.08	10.58	0.00	4.00	7.00	14.00	110.00
AlignmentHorQntROD	4.05	8.28	6.83	57.13	0.00	1.00	2.00	5.00	79.00
AlignmentIndexHorROD	2.03	4.99	8.51	98.26	0.00	0.00	1.00	2.00	79.00
AlignmentVertQntROD	8.65	15.70	3.45	13.66	0.00	1.00	3.00	9.00	100.00
AlignmentIndexVertROD	4.32	9.49	4.63	27.49	0.00	0.00	1.00	4.00	99.00
AlignmentVertHorRatioROD	4.90	9.12	3.65	15.67	0.01	0.37	1.50	5.00	76.00
TObjectsQntIC	30.83	21.00	1.28	2.32	0.00	16.00	27.00	41.00	165.00
TextSpatialDensityIC	0.13	0.08	1.37	3.21	0.00	0.08	0.12	0.17	0.58
LinkSpatialDensityIC	0.09	0.09	3.17	20.86	0.00	0.03	0.07	0.13	1.09
LinkCharacterDensityIC	0.49	0.32	0.01	-1.25	0.00	0.21	0.49	0.77	1.00

Table C.10: Web Objects per Link Type

[NULL]	domain	external	local
5,075	2,816	828	314
56.18%	31.17%	9.17%	3.48%

C.4 Correlation between Web Object's Features

Pearson Correlation

Figure C.2 illustrates the correlation matrix as a heat map by using Pearson's correlation coefficient.

Spearman Correlation

Figure C.3 illustrates the correlation matrix as a heat map by using Spearman's correlation coefficient.

Figure C.2: Correlation Matrix of all connection search senarios by using Pearson's correlation coefficient

Figure C.3: Correlation Matrix of all connection search scenarios by using Spearman's correlation coefficient

Name	AreaIO	AspectRatioIO	EmphasisIO	FontSizeIO	LinesQntIO	TokensQntIO	SimilarTypesQntROC	AlignmentQntROC	AlignmentHorQntROC	AlignmentIndexHorROC	AlignmentVertQntROC	AlignmentIndexVertROC	AlignmentFactorROC	AlignmentVertHorRatioROC	OrthogonalVisibleObjQntROC	AlignedOrthogonalVisibleObjQntROC	FullyAlignedOrthogonalVisibleObjQntROC	PixelsToCharacterIC
ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
1	1.0	0.82	0.02	0.29	0.39	0.64	-0.25	-0.17	-0.48	-0.43	0.17	0.13	-0.07	0.37	0.05	-0.19	-0.46	0.04
2	0.82	1.0	0.05	0.09	0.04	0.63	-0.07	-0.09	-0.42	-0.37	0.18	0.14	-0.04	0.35	0.03	-0.11	-0.34	-0.07
3	0.02	0.05	1.0	0.15	-0.02	0.0	0.16	-0.01	-0.04	-0.05	0.0	-0.05	-0.06	0.01	-0.05	-0.09	-0.06	-0.04
4	0.29	0.09	0.15	1.0	0.03	0.06	-0.13	-0.14	-0.19	-0.15	-0.02	-0.05	0.01	0.09	-0.03	-0.13	-0.14	0.2
5	0.39	0.04	-0.02	0.03	1.0	0.4	0.01	-0.01	-0.12	-0.15	0.1	0.09	0.02	0.13	0.03	-0.05	-0.24	-0.05
6	0.64	0.63	0.0	0.06	0.4	1.0	0.1	-0.03	-0.3	-0.26	0.18	0.16	-0.02	0.29	0.03	-0.06	-0.28	-0.16
7	-0.25	-0.07	0.16	-0.13	0.01	0.1	1.0	0.48	0.32	0.25	0.26	0.21	-0.22	0.02	0.34	0.36	0.35	-0.67
8	-0.17	-0.09	-0.01	-0.14	-0.01	-0.03	0.48	1.0	0.31	0.18	0.7	0.6	0.55	0.35	0.16	0.69	0.3	-0.37
9	-0.48	-0.42	-0.04	-0.19	-0.12	-0.3	0.32	0.31	1.0	0.77	-0.32	-0.27	0.07	-0.73	0.31	0.41	0.71	-0.14
10	-0.43	-0.37	-0.05	-0.15	-0.15	-0.26	0.25	0.18	0.77	1.0	-0.35	-0.27	-0.01	-0.64	0.28	0.3	0.63	-0.08
11	0.17	0.18	0.0	-0.02	0.1	0.18	0.26	0.7	-0.32	-0.35	1.0	0.84	0.41	0.88	-0.01	0.49	-0.14	-0.28
12	0.13	0.14	-0.05	-0.05	0.09	0.16	0.21	0.6	-0.27	-0.27	0.84	1.0	0.36	0.74	-0.01	0.44	-0.12	-0.25
13	-0.07	-0.04	-0.06	0.01	0.02	-0.02	-0.22	0.55	0.07	-0.01	0.41	0.36	1.0	0.26	-0.34	0.36	0.06	0.16
14	0.37	0.35	0.01	0.09	0.13	0.29	0.02	0.35	-0.73	-0.64	0.88	0.74	0.26	1.0	-0.16	0.15	-0.45	-0.13
15	0.05	0.03	-0.05	-0.03	0.03	0.03	0.34	0.16	0.31	0.28	-0.01	-0.01	-0.34	-0.16	1.0	0.3	0.28	-0.25
16	-0.19	-0.11	-0.09	-0.13	-0.05	-0.06	0.36	0.69	0.41	0.3	0.49	0.44	0.36	0.15	0.3	1.0	0.56	-0.26
17	-0.46	-0.34	-0.06	-0.14	-0.24	-0.28	0.35	0.3	0.71	0.63	-0.14	-0.12	0.06	-0.45	0.28	0.56	1.0	-0.18
18	0.04	-0.07	-0.04	0.2	-0.05	-0.16	-0.67	-0.37	-0.14	-0.08	-0.28	-0.25	0.16	-0.13	-0.25	-0.26	-0.18	1.0
19	-0.04	-0.04	0.01	0.04	-0.07	0.0	0.0	-0.11	-0.04	-0.01	-0.04	-0.05	-0.2	-0.01	0.0	-0.08	-0.01	0.04
20	0.05	-0.01	0.01	0.08	-0.04	-0.06	-0.17	-0.17	-0.1	-0.08	-0.1	-0.11	-0.17	-0.02	0.01	-0.15	-0.12	0.23
21	0.95	0.89	0.02	0.2	0.31	0.66	-0.2	-0.16	-0.47	-0.42	0.16	0.13	-0.03	0.36	0.02	-0.18	-0.43	-0.04
22	0.31	0.04	-0.07	0.05	0.34	0.08	-0.43	-0.27	-0.23	-0.2	-0.11	-0.1	-0.05	0.04	-0.08	-0.26	-0.33	0.16
23	-0.14	-0.12	-0.06	-0.03	-0.04	-0.07	-0.07	-0.11	0.07	0.2	-0.15	-0.14	0.07	-0.14	-0.11	-0.04	0.12	0.14
24	-0.06	0.03	-0.03	-0.17	0.06	0.09	0.19	0.13	0.09	0.08	0.01	0.07	0.16	-0.04	-0.02	0.1	0.08	-0.33
25	0.05	0.1	-0.05	-0.09	0.13	0.16	0.16	0.1	0.04	0.09	0.0	0.06	0.11	-0.02	0.01	0.07	0.05	-0.21
26	-0.12	-0.06	0.05	-0.05	0.05	0.02	0.39	0.85	0.29	0.12	0.56	0.46	0.55	0.26	0.06	0.55	0.25	-0.31
27	-0.43	-0.38	0.01	-0.16	-0.09	-0.26	0.23	0.22	0.88	0.67	-0.35	-0.31	0.11	-0.7	0.2	0.32	0.63	-0.08
28	-0.38	-0.33	-0.02	-0.12	-0.1	-0.22	0.17	0.09	0.68	0.85	-0.39	-0.34	0.03	-0.62	0.15	0.2	0.54	0.0
29	0.2	0.21	0.02	0.03	0.13	0.22	0.24	0.64	-0.32	-0.38	0.92	0.77	0.39	0.82	-0.03	0.41	-0.16	-0.26
30	0.15	0.16	-0.02	-0.02	0.12	0.19	0.22	0.56	-0.26	-0.31	0.78	0.9	0.36	0.69	-0.05	0.38	-0.13	-0.26
31	0.37	0.34	0.0	0.1	0.13	0.29	0.04	0.32	-0.69	-0.62	0.82	0.7	0.21	0.94	-0.13	0.11	-0.44	-0.14
32	-0.1	-0.05	0.04	-0.16	-0.04	-0.01	0.77	0.46	0.25	0.19	0.31	0.26	-0.42	0.09	0.52	0.34	0.26	-0.62
33	0.03	0.08	0.1	-0.02	0.09	0.18	0.64	0.35	0.11	0.04	0.28	0.25	-0.16	0.15	0.25	0.23	0.14	-0.93
34	-0.02	0.02	0.16	-0.11	0.05	0.08	0.35	0.12	0.11	0.07	0.01	0.02	-0.13	-0.06	0.16	0.09	0.07	-0.45
35	-0.12	-0.07	0.15	-0.06	-0.03	-0.06	0.05	-0.01	0.1	0.11	-0.13	-0.11	0.11	-0.15	-0.07	0.03	0.09	0.03

Name	AvgWeightedFGColorROC	AvgWeightedBGColorROC	RelativeWidthROW	RelativeHeightROW	RelativeXPositionROW	RelativeYPositionROW	GridLocationX3ROTW	AlignmentQntROD	AlignmentHorQntROD	AlignmentIndexHorROD	AlignmentVertQntROD	AlignmentIndexVertROD	AlignmentVertHorRatioROD	TObjectsQntIC	TextSpatialDensityIC	LinkSpatialDensityIC	LinkCharacterDensityIC
ID	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**
1	-0.04	0.05	0.95	0.31	-0.14	-0.06	0.05	-0.12	-0.43	-0.38	0.2	0.15	0.37	-0.1	0.03	-0.02	-0.12
2	-0.04	-0.01	0.89	0.04	-0.12	0.03	0.1	-0.06	-0.38	-0.33	0.21	0.16	0.34	-0.05	0.08	0.02	-0.07
3	0.01	0.01	0.02	-0.07	-0.06	-0.03	-0.05	0.05	0.01	-0.02	0.02	-0.02	0.0	0.04	0.1	0.16	0.15
4	0.04	0.08	0.2	0.05	-0.03	-0.17	-0.09	-0.05	-0.16	-0.12	0.03	-0.02	0.1	-0.16	-0.02	-0.11	-0.06
5	-0.07	-0.04	0.31	0.34	-0.04	0.06	0.13	0.05	-0.09	-0.1	0.13	0.12	0.13	-0.04	0.09	0.05	-0.03
6	0.0	-0.06	0.66	0.08	-0.07	0.09	0.16	0.02	-0.26	-0.22	0.22	0.19	0.29	-0.01	0.18	0.08	-0.06
7	0.0	-0.17	-0.2	-0.43	-0.07	0.19	0.16	0.39	0.23	0.17	0.24	0.22	0.04	0.77	0.64	0.35	0.05
8	-0.11	-0.17	-0.16	-0.27	-0.11	0.13	0.1	0.85	0.22	0.09	0.64	0.56	0.32	0.46	0.35	0.12	-0.01
9	-0.04	-0.1	-0.47	-0.23	0.07	0.09	0.04	0.29	0.88	0.68	-0.32	-0.26	-0.69	0.25	0.11	0.11	0.1
10	-0.01	-0.08	-0.42	-0.2	0.2	0.08	0.09	0.12	0.67	0.85	-0.38	-0.31	-0.62	0.19	0.04	0.07	0.11
11	-0.04	-0.1	0.16	-0.11	-0.15	0.01	0.0	0.56	-0.35	-0.39	0.92	0.78	0.82	0.31	0.28	0.01	-0.13
12	-0.05	-0.11	0.13	-0.1	-0.14	0.07	0.06	0.46	-0.31	-0.34	0.77	0.9	0.7	0.26	0.25	0.02	-0.11
13	-0.2	-0.17	-0.03	-0.05	0.07	0.16	0.11	0.55	0.11	0.03	0.39	0.36	0.21	-0.42	-0.16	-0.13	0.11
14	-0.01	-0.02	0.36	0.04	-0.14	-0.04	-0.02	0.26	-0.7	-0.62	0.82	0.69	0.94	0.09	0.15	-0.06	-0.15
15	0.0	0.01	0.02	-0.08	-0.11	-0.02	0.01	0.06	0.2	0.15	-0.03	-0.05	-0.13	0.52	0.25	0.16	-0.07
16	-0.08	-0.15	-0.18	-0.26	-0.04	0.1	0.07	0.55	0.32	0.2	0.41	0.38	0.11	0.34	0.23	0.09	0.03
17	-0.01	-0.12	-0.43	-0.33	0.12	0.08	0.05	0.25	0.63	0.54	-0.16	-0.13	-0.44	0.26	0.14	0.07	0.09
18	0.04	0.23	-0.04	0.16	0.14	-0.33	-0.21	-0.31	-0.08	0.0	-0.26	-0.26	-0.14	-0.62	-0.93	-0.45	0.03
19	1.0	0.15	-0.05	-0.04	0.01	-0.26	-0.24	-0.12	-0.07	-0.02	-0.03	-0.07	0.02	0.07	-0.01	-0.02	-0.08
20	0.15	1.0	0.01	0.14	-0.01	-0.33	-0.27	-0.15	-0.08	-0.08	-0.1	-0.14	-0.03	-0.03	-0.21	-0.14	-0.15
21	-0.05	0.01	1.0	0.27	-0.1	0.02	0.07	-0.11	-0.42	-0.37	0.2	0.15	0.36	-0.11	0.08	0.0	-0.11
22	-0.04	0.14	0.27	1.0	0.01	-0.04	-0.04	-0.3	-0.21	-0.18	-0.16	-0.15	-0.01	-0.24	-0.18	-0.04	-0.05
23	0.01	-0.01	-0.1	0.01	1.0	0.01	0.21	-0.1	0.11	0.37	-0.18	-0.15	-0.18	-0.15	-0.18	-0.08	0.12
24	-0.26	-0.33	0.02	-0.04	0.01	1.0	0.83	0.17	0.14	0.11	0.02	0.18	-0.06	0.02	0.22	0.27	0.22
25	-0.24	-0.27	0.07	-0.04	0.21	0.83	1.0	0.12	0.08	0.13	0.0	0.14	-0.04	0.0	0.11	0.25	0.26
26	-0.12	-0.15	-0.11	-0.3	-0.1	0.17	0.12	1.0	0.35	0.16	0.65	0.54	0.25	0.31	0.3	0.11	0.03
27	-0.07	-0.08	-0.42	-0.21	0.11	0.14	0.08	0.35	1.0	0.75	-0.34	-0.28	-0.76	0.12	0.03	0.13	0.19
28	-0.02	-0.08	-0.37	-0.18	0.37	0.11	0.13	0.16	0.75	1.0	-0.39	-0.32	-0.68	0.07	-0.06	0.06	0.18
29	-0.03	-0.1	0.2	-0.16	-0.18	0.02	0.0	0.65	-0.34	-0.39	1.0	0.83	0.86	0.26	0.29	0.0	-0.14
30	-0.07	-0.14	0.15	-0.15	-0.15	0.18	0.14	0.54	-0.28	-0.32	0.83	1.0	0.71	0.23	0.26	0.04	-0.09
31	0.02	-0.03	0.36	-0.01	-0.18	-0.06	-0.04	0.25	-0.76	-0.68	0.86	0.71	1.0	0.12	0.18	-0.08	-0.2
32	0.07	-0.03	-0.11	-0.24	-0.15	0.02	0.0	0.31	0.12	0.07	0.26	0.23	0.12	1.0	0.59	0.26	-0.14
33	-0.01	-0.21	0.08	-0.18	-0.18	0.22	0.11	0.3	0.03	-0.06	0.29	0.26	0.18	0.59	1.0	0.43	-0.08
34	-0.02	-0.14	0.0	-0.04	-0.08	0.27	0.25	0.11	0.13	0.06	0.0	0.04	-0.08	0.26	0.43	1.0	0.59
35	-0.08	-0.15	-0.11	-0.05	0.12	0.22	0.26	0.03	0.19	0.18	-0.14	-0.09	-0.2	-0.14	-0.08	0.59	1.0

APPENDIX D

Detailed Results

D.1 Different Class Balancing

Table D.1 shows detailed results for figure 7.2 in chapter 7. The decimal places have been truncated to 4.

Table D.1: Detailed Results for NP-Ratio for figure 7.2 in chapter 7. That table included 69,920 trained classifiers.

Algorithm	NP-Ratio	Transformation	Precision	CI Lower	CI Upper
Logistic regression	1	none	0.9888	0.9467	1.0000
Logistic regression	2.5	none	0.9737	0.9097	1.0000
Logistic regression	5	none	0.9432	0.8507	1.0000
Logistic regression	10	none	0.8819	0.7529	1.0000
Logistic regression	1	PCA	0.9785	0.9205	1.0000
Logistic regression	2.5	PCA	0.9561	0.8741	1.0000
Logistic regression	5	PCA	0.9204	0.8121	1.0000
Logistic regression	10	PCA	0.8805	0.7509	1.0000
Logistic regression	1	z-score	0.9890	0.9473	1.0000
Logistic regression	2.5	z-score	0.9723	0.9067	1.0000
Logistic regression	5	z-score	0.9416	0.8479	1.0000
Logistic regression	10	z-score	0.8808	0.7512	1.0000
C4.5	1	none	0.9977	0.9786	1.0000
C4.5	2.5	none	0.9986	0.9838	1.0000
C4.5	5	none	0.9984	0.9824	1.0000
C4.5	10	none	0.9973	0.9763	1.0000

D.2 Feature Importance

Table D.2: Details for the feature Importance of the logistic regression with non transformation and NP-ratio 1

Feature Name	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.05$
LinesQntIO	0.7129	0.7162	0.7185
AlignmentFactorROC	0.6200	0.6205	0.6224
AlignmentHorQntROC	0.6200	0.6213	0.6268
AlignedOrthogonalVisibleObjQntROC	0.6200	0.6202	0.6212
AlignmentIndexHorROC	0.6197	0.6200	0.6218
AlignmentHorQntROD	0.6162	0.6166	0.6187
AlignmentIndexVertROC	0.6050	0.6050	0.6051
FullyAlignedOrthogonalVisibleObjQntROC	0.6029	0.6038	0.6065
AreaIO	0.6025	0.6063	0.6088
AlignmentIndexHorROD	0.6021	0.6027	0.6049
AlignmentQntROC	0.6020	0.6030	0.6073
AspectRatioIO	0.6019	0.6027	0.6056
AlignmentVertQntROC	0.6018	0.6026	0.6063
RelativeXPositionROW	0.6018	0.6027	0.6071
AlignmentIndexVertROD	0.6018	0.6018	0.6019
AvgWeightedBGColorROC	0.6017	0.6026	0.6063
BottomTxtOfOrthVisibleObjsROC	0.6017	0.6023	0.6059
RelativeCenterEuklidean	0.6017	0.6030	0.6072
RelativeHeightROW	0.6017	0.6020	0.6042
RelativeWidthROW	0.6017	0.6025	0.6059
TextOfNearestTxtObjROC	0.6017	0.6025	0.6062
AvgWeightedFGColorROC	0.6017	0.6023	0.6061
GridLocationX3ROTW	0.6017	0.6025	0.6052
AlignmentVertHorRatioROC	0.6017	0.6025	0.6055
AlignmentVertHorRatioROD	0.6017	0.6019	0.6040
AlignmentVertQntROD	0.6017	0.6021	0.6033
RelativeYPositionROW	0.6017	0.6022	0.6042
AlignmentQntROD	0.6016	0.6019	0.6027
BackgroundColorIO	0.6016	0.6016	0.6017
LeftTxtOfOrthVisibleObjsROC	0.6016	0.6017	0.6017
LinkCharacterDensityIC	0.6016	0.6022	0.6053
MostSimilarTxtOfOrthVisibleSideObjects	0.6016	0.6019	0.6035
UpperTxtOfOrthVisibleObjsROC	0.6016	0.6024	0.6051
SimilarTypesQntROC	0.6016	0.6016	0.6021
LinkSpatialDensityIC	0.6016	0.6017	0.6018
OrthogonalVisibleObjQntROC	0.6016	0.6016	0.6017

Feature Name	$\alpha = 0.001$	$\alpha = 0.01$	$\alpha = 0.05$
RightTxtOfOrthVisibleObjsROC	0.6016	0.6016	0.6018
TextIO	0.6016	0.6016	0.6020
TextOfNearestOrthVisibleObjsROC	0.6016	0.6020	0.6032
TObjectsQntIC	0.6016	0.6022	0.6048
PixelsToCharacterIC	0.5725	0.5785	0.5819
EmphasisIO	0.5383	0.5390	0.5572
TokensQntIO	0.5383	0.5386	0.5404
FontSizeIO	0.5382	0.5386	0.5406
ForegroundColorIO	0.5382	0.5382	0.5383

D.3 Logistic Regression

This section provides the results for the logistic regression. Due to the nature of this technique the different sub results are included in the different subsections.

Preprocessing for the Logistic Regression

Table D.3: Detailed Results for logistic regression for figure 7.4 in chapter 7. 52,440 classifiers.

Transformation	Scenario	Precision	CI Lower	CI Upper	Potential
None	Accommodation	0.9985	0.9830	1.0000	0.0015
PCA	Accommodation	0.9868	0.9405	1.0000	0.0132
Z-Score	Accommodation	0.9985	0.9830	1.0000	0.0015
None	All Connections	0.9786	0.9206	1.0000	0.0182
PCA	All Connections	0.9722	0.9063	1.0000	0.0235
Z-Score	All Connections	0.9791	0.9218	1.0000	0.0176
None	Bus	0.9929	0.9587	1.0000	0.0071
PCA	Bus	0.9946	0.9650	1.0000	0.0054
Z-Score	Bus	0.9929	0.9587	1.0000	0.0071
None	Flight	0.9949	0.9660	1.0000	0.0026
PCA	Flight	0.9692	0.8995	1.0000	0.0244
Z-Score	Flight	0.9949	0.9660	1.0000	0.0026
None	Train	1.0000	1.0000	1.0000	0.0000
PCA	Train	0.9875	0.9423	1.0000	0.0104
Z-Score	Train	1.0000	1.0000	1.0000	0.0000

D.4 Support Vector Machines - Linear Kernel

Table D.4: Details for the SVM with the linear kernel represented in figures 7.6 in chapter 7 ordered by Precision and Potential. The results contain the data of 10,260 trained classifiers.

Cost	Precision	CI Lower	CI Upper	Potential	Best
0.1	0.9899	0.9499	1.0000	0.0101	true
10	0.9867	0.9409	1.0000	0.0128	false
100	0.9867	0.9409	1.0000	0.0128	false
1	0.9858	0.9384	1.0000	0.0137	false

D.5 Support Vector Machines - Polyomial Kernel

Table D.5: Details for the SVM with the polynomial kernel represented in figures 7.7 in chapter 7 ordered by Precision and Potential. The results contain the data of 131,100 trained classifiers.

Cost C	γ	Dim	Precision	CI Lower	CI Upper	Potential	Best
0.01	0.05	2	1.0000	1.0000	1.0000	0.0000	true
1	0.05	2	1.0000	1.0000	1.0000	0.0000	true
0.01	0.125	2	1.0000	1.0000	1.0000	0.0000	true
0.01	0.25	2	1.0000	1.0000	1.0000	0.0000	true
0.01	0.5	2	1.0000	1.0000	1.0000	0.0000	true
0.01	0.05	3	1.0000	1.0000	1.0000	0.0000	true
0.01	0.05	4	1.0000	1.0000	1.0000	0.0000	true
1	0.05	4	1.0000	1.0000	1.0000	0.0000	true
10	0.05	4	1.0000	1.0000	1.0000	0.0000	true
0.01	0.125	4	1.0000	1.0000	1.0000	0.0000	true
0.01	0.25	4	1.0000	1.0000	1.0000	0.0000	true
1	0.05	3	0.9995	0.9910	1.0000	0.0005	false
0.01	0.125	3	0.9995	0.9910	1.0000	0.0005	false
0.01	0.25	3	0.9995	0.9910	1.0000	0.0005	false
10	0.05	3	0.9982	0.9810	1.0000	0.0018	false
0.01	0.5	3	0.9982	0.9810	1.0000	0.0018	false
1	0.125	4	0.9982	0.9810	1.0000	0.0018	false
0.01	0.5	4	0.9977	0.9786	1.0000	0.0023	false
1000	0.05	4	0.9973	0.9763	1.0000	0.0027	false
10000	0.05	4	0.9973	0.9763	1.0000	0.0027	false

Cost C	γ	Dim	Precision	CI Lower	CI Upper	Potential	Best
10	0.125	4	0.9973	0.9763	1.0000	0.0027	false
1000	0.125	4	0.9973	0.9763	1.0000	0.0027	false
10000	0.125	4	0.9973	0.9763	1.0000	0.0027	false
1	0.25	4	0.9973	0.9763	1.0000	0.0027	false
10	0.25	4	0.9973	0.9763	1.0000	0.0027	false
1000	0.25	4	0.9973	0.9763	1.0000	0.0027	false
10000	0.25	4	0.9973	0.9763	1.0000	0.0027	false
1	0.5	4	0.9973	0.9763	1.0000	0.0027	false
10	0.5	4	0.9973	0.9763	1.0000	0.0027	false
1000	0.5	4	0.9973	0.9763	1.0000	0.0027	false
10000	0.5	4	0.9973	0.9763	1.0000	0.0027	false
1	0.125	2	0.9968	0.9742	1.0000	0.0032	false
1	0.125	3	0.9968	0.9742	1.0000	0.0032	false
10	0.05	2	0.9963	0.9721	1.0000	0.0037	false
1	0.25	3	0.9954	0.9684	1.0000	0.0046	false
1000	0.05	3	0.9950	0.9666	1.0000	0.0050	false
10000	0.05	3	0.9950	0.9666	1.0000	0.0050	false
10	0.125	3	0.9950	0.9666	1.0000	0.0050	false
1000	0.125	3	0.9950	0.9666	1.0000	0.0050	false
10000	0.125	3	0.9950	0.9666	1.0000	0.0050	false
10	0.25	3	0.9950	0.9666	1.0000	0.0050	false
1000	0.25	3	0.9950	0.9666	1.0000	0.0050	false
10000	0.25	3	0.9950	0.9666	1.0000	0.0050	false
1	0.5	3	0.9950	0.9666	1.0000	0.0050	false
10	0.5	3	0.9950	0.9666	1.0000	0.0050	false
1000	0.5	3	0.9950	0.9666	1.0000	0.0050	false
10000	0.5	3	0.9950	0.9666	1.0000	0.0050	false
1000	0.05	2	0.9950	0.9666	1.0000	0.0046	false
10000	0.05	2	0.9950	0.9666	1.0000	0.0046	false
1000	0.125	2	0.9950	0.9666	1.0000	0.0046	false
10000	0.125	2	0.9950	0.9666	1.0000	0.0046	false
1	0.25	2	0.9950	0.9666	1.0000	0.0046	false
10	0.25	2	0.9950	0.9666	1.0000	0.0046	false
1000	0.25	2	0.9950	0.9666	1.0000	0.0046	false
10000	0.25	2	0.9950	0.9666	1.0000	0.0046	false
10	0.5	2	0.9950	0.9666	1.0000	0.0046	false
1000	0.5	2	0.9950	0.9666	1.0000	0.0046	false
10000	0.5	2	0.9950	0.9666	1.0000	0.0046	false
10	0.125	2	0.9945	0.9649	1.0000	0.0050	false
1	0.5	2	0.9945	0.9649	1.0000	0.0050	false

D.6 Support Vector Machines - Radial Kernel

Table D.6: Details from Figures 7.8 in chapter 7 ordered by Precision and Potential. The results contain the data of 76,475 trained classifiers.

Cost C	γ	Precision	CI Lower	CI Upper	Potential	Best
0.1	0.0005	1.0000	1.0000	1.0000	0.0000	true
1	0.0005	0.9986	0.9838	1.0000	0.0014	false
0.1	0.005	0.9977	0.9786	1.0000	0.0023	false
10	0.0005	0.9950	0.9666	1.0000	0.0050	false
1	0.005	0.9941	0.9632	1.0000	0.0059	false
10	0.005	0.9858	0.9384	1.0000	0.0137	false
10,000	0.0005	0.9849	0.9360	1.0000	0.0146	false
1,000	0.0005	0.9840	0.9337	1.0000	0.0156	false
1,000	0.005	0.9812	0.9269	1.0000	0.0174	false
10,000	0.005	0.9808	0.9258	1.0000	0.0178	false
1	0.05	0.9442	0.8522	1.0000	0.0471	false
10	0.05	0.9378	0.8409	1.0000	0.0517	false
1,000	0.05	0.9368	0.8394	1.0000	0.0531	false
10,000	0.05	0.9368	0.8394	1.0000	0.0531	false
0.1	0.05	0.8979	0.7766	1.0000	0.0503	false
10	0.125	0.7876	0.6237	0.9515	0.1071	false
1,000	0.125	0.7867	0.6226	0.9509	0.1085	false
10,000	0.125	0.7867	0.6226	0.9509	0.1085	false
1	0.125	0.7849	0.6202	0.9496	0.1002	false
0.1	0.125	0.5515	0.3522	0.7508	0.0874	false
0.1	10	0.4229	0.2249	0.6209	0.0023	false
10	0.5	0.3469	0.1562	0.5377	0.0467	false
1,000	0.5	0.3469	0.1562	0.5377	0.0467	false
10,000	0.5	0.3469	0.1562	0.5377	0.0467	false
1	0.5	0.3304	0.1419	0.5189	0.0416	false
0.1	0.5	0.3213	0.1341	0.5084	0.0316	false
10	1	0.3039	0.1196	0.4882	0.0380	false
1,000	1	0.3039	0.1196	0.4882	0.0380	false
10,000	1	0.3039	0.1196	0.4882	0.0380	false
1	1	0.3021	0.1181	0.4861	0.0394	false
10	10	0.2563	0.0813	0.4313	0.0087	false
1,000	10	0.2563	0.0813	0.4313	0.0087	false

Cost C	γ	Precision	CI Lower	CI Upper	Potential	Best
10,000	10	0.2563	0.0813	0.4313	0.0087	false
1	10	0.2508	0.0771	0.4245	0.0073	false
0.1	1	0.2494	0.0760	0.4228	0.0339	false

D.7 Support Vector Machines - Sigmoid Kernel

Table D.7: Details from the SVM with the sigmoid kernel. A graphical illustration can be found at figures 7.9 in chapter 7 ordered by Precision and Potential. The results contain the data of 458,850 trained classifiers.

Cost C	γ	Coefficient c_0	Precision	CI Lower	CI Upper	Potential	Best
0.1	0.00025	-2.5	1.0000	1.0000	1.0000	0.0000	true
1	0.00025	-2.5	1.0000	1.0000	1.0000	0.0000	true
10	0.00025	-2.5	1.0000	1.0000	1.0000	0.0000	true
0.1	0.0025	-2.5	1.0000	1.0000	1.0000	0.0000	true
1	0.0025	-2.5	1.0000	1.0000	1.0000	0.0000	true
0.1	0.025	-2.5	1.0000	1.0000	1.0000	0.0000	true
1	0.025	-2.5	1.0000	1.0000	1.0000	0.0000	true
1	1	-2.5	1.0000	1.0000	1.0000	0.0000	true
10	1	-2.5	1.0000	1.0000	1.0000	0.0000	true
1,000	1	-2.5	1.0000	1.0000	1.0000	0.0000	true
10,000	1	-2.5	1.0000	1.0000	1.0000	0.0000	true
1	10	-2.5	1.0000	1.0000	1.0000	0.0000	true
10	10	-2.5	1.0000	1.0000	1.0000	0.0000	true
1,000	10	-2.5	1.0000	1.0000	1.0000	0.0000	true
10,000	10	-2.5	1.0000	1.0000	1.0000	0.0000	true
0.1	0.00025	-1.25	1.0000	1.0000	1.0000	0.0000	true
1	0.00025	-1.25	1.0000	1.0000	1.0000	0.0000	true
0.1	0.0025	-1.25	1.0000	1.0000	1.0000	0.0000	true
1	1	-1.25	1.0000	1.0000	1.0000	0.0000	true
10	1	-1.25	1.0000	1.0000	1.0000	0.0000	true
1,000	1	-1.25	1.0000	1.0000	1.0000	0.0000	true
10,000	1	-1.25	1.0000	1.0000	1.0000	0.0000	true
1	10	-1.25	1.0000	1.0000	1.0000	0.0000	true
10	10	-1.25	1.0000	1.0000	1.0000	0.0000	true
1,000	10	-1.25	1.0000	1.0000	1.0000	0.0000	true
10,000	10	-1.25	1.0000	1.0000	1.0000	0.0000	true

Cost C	γ	Coefficient c_0	Precision	CI Lower	CI Upper	Potential	Best
0.1	0.00025	-0.5	1.0000	1.0000	1.0000	0.0000	true
1	0.00025	-0.5	1.0000	1.0000	1.0000	0.0000	true
0.1	0.0025	-0.5	1.0000	1.0000	1.0000	0.0000	true
1	1	-0.5	1.0000	1.0000	1.0000	0.0000	true
10	1	-0.5	1.0000	1.0000	1.0000	0.0000	true
1,000	1	-0.5	1.0000	1.0000	1.0000	0.0000	true
10,000	1	-0.5	1.0000	1.0000	1.0000	0.0000	true
1	10	-0.5	1.0000	1.0000	1.0000	0.0000	true
10	10	-0.5	1.0000	1.0000	1.0000	0.0000	true
1,000	10	-0.5	1.0000	1.0000	1.0000	0.0000	true
10,000	10	-0.5	1.0000	1.0000	1.0000	0.0000	true
0.1	0.00025	0	1.0000	1.0000	1.0000	0.0000	true
1	0.00025	0	1.0000	1.0000	1.0000	0.0000	true
0.1	0.0025	0	1.0000	1.0000	1.0000	0.0000	true
0.1	1	0	1.0000	1.0000	1.0000	0.0000	true
1	1	0	1.0000	1.0000	1.0000	0.0000	true
10	1	0	1.0000	1.0000	1.0000	0.0000	true
1,000	1	0	1.0000	1.0000	1.0000	0.0000	true
10,000	1	0	1.0000	1.0000	1.0000	0.0000	true
1	10	0	1.0000	1.0000	1.0000	0.0000	true
10	10	0	1.0000	1.0000	1.0000	0.0000	true
1,000	10	0	1.0000	1.0000	1.0000	0.0000	true
10,000	10	0	1.0000	1.0000	1.0000	0.0000	true
0.1	0.00025	0.5	1.0000	1.0000	1.0000	0.0000	true
1	0.00025	0.5	1.0000	1.0000	1.0000	0.0000	true
0.1	0.0025	0.5	1.0000	1.0000	1.0000	0.0000	true
10	0.125	0.5	1.0000	1.0000	1.0000	0.0000	true
0.1	1	0.5	1.0000	1.0000	1.0000	0.0000	true
1	1	0.5	1.0000	1.0000	1.0000	0.0000	true
10	1	0.5	1.0000	1.0000	1.0000	0.0000	true
1,000	1	0.5	1.0000	1.0000	1.0000	0.0000	true
10,000	1	0.5	1.0000	1.0000	1.0000	0.0000	true
1	10	0.5	1.0000	1.0000	1.0000	0.0000	true
10	10	0.5	1.0000	1.0000	1.0000	0.0000	true
1,000	10	0.5	1.0000	1.0000	1.0000	0.0000	true
10,000	10	0.5	1.0000	1.0000	1.0000	0.0000	true
0.1	0.00025	1.25	1.0000	1.0000	1.0000	0.0000	true
1	0.00025	1.25	1.0000	1.0000	1.0000	0.0000	true
0.1	0.0025	1.25	1.0000	1.0000	1.0000	0.0000	true
1	0.125	1.25	1.0000	1.0000	1.0000	0.0000	true
10	0.125	1.25	1.0000	1.0000	1.0000	0.0000	true

Cost C	γ	Coefficient c_0	Precision	CI Lower	CI Upper	Potential	Best
0.1	1	1.25	1.0000	1.0000	1.0000	0.0000	true
1	1	1.25	1.0000	1.0000	1.0000	0.0000	true
10	1	1.25	1.0000	1.0000	1.0000	0.0000	true
1,000	1	1.25	1.0000	1.0000	1.0000	0.0000	true
10,000	1	1.25	1.0000	1.0000	1.0000	0.0000	true
1	10	1.25	1.0000	1.0000	1.0000	0.0000	true
10	10	1.25	1.0000	1.0000	1.0000	0.0000	true
1,000	10	1.25	1.0000	1.0000	1.0000	0.0000	true
10,000	10	1.25	1.0000	1.0000	1.0000	0.0000	true
0.1	0.00025	2.5	1.0000	1.0000	1.0000	0.0000	true
1	0.00025	2.5	1.0000	1.0000	1.0000	0.0000	true
10	0.00025	2.5	1.0000	1.0000	1.0000	0.0000	true
0.1	0.0025	2.5	1.0000	1.0000	1.0000	0.0000	true
1	0.0025	2.5	1.0000	1.0000	1.0000	0.0000	true
0.1	0.125	2.5	1.0000	1.0000	1.0000	0.0000	true
1	0.125	2.5	1.0000	1.0000	1.0000	0.0000	true
10	0.125	2.5	1.0000	1.0000	1.0000	0.0000	true
1,000	0.125	2.5	1.0000	1.0000	1.0000	0.0000	true
0.1	1	2.5	1.0000	1.0000	1.0000	0.0000	true
1	1	2.5	1.0000	1.0000	1.0000	0.0000	true
10	1	2.5	1.0000	1.0000	1.0000	0.0000	true
1,000	1	2.5	1.0000	1.0000	1.0000	0.0000	true
10,000	1	2.5	1.0000	1.0000	1.0000	0.0000	true
1	10	2.5	1.0000	1.0000	1.0000	0.0000	true
10	10	2.5	1.0000	1.0000	1.0000	0.0000	true
1,000	10	2.5	1.0000	1.0000	1.0000	0.0000	true
10,000	10	2.5	1.0000	1.0000	1.0000	0.0000	true
0.1	1	-1.25	0.9995	0.9910	1.0000	0.0005	false
0.1	1	-0.5	0.9995	0.9910	1.0000	0.0005	false
0.1	0.125	0.5	0.9995	0.9910	1.0000	0.0005	false
1	0.125	0.5	0.9995	0.9910	1.0000	0.0005	false
1,000	0.125	0.5	0.9995	0.9910	1.0000	0.0005	false
0.1	0.125	1.25	0.9995	0.9910	1.0000	0.0005	false
1,000	0.125	1.25	0.9995	0.9910	1.0000	0.0005	false
10	0.0025	-2.5	0.9991	0.9870	1.0000	0.0009	false
0.1	1	-2.5	0.9991	0.9870	1.0000	0.0009	false
10	0.00025	-1.25	0.9991	0.9870	1.0000	0.0009	false
0.1	0.025	-1.25	0.9991	0.9870	1.0000	0.0009	false
10,000	0.125	0.5	0.9991	0.9870	1.0000	0.0009	false
10	0.00025	1.25	0.9991	0.9870	1.0000	0.0009	false
1	0.0025	1.25	0.9991	0.9870	1.0000	0.0009	false

Cost C	γ	Coefficient c_0	Precision	CI Lower	CI Upper	Potential	Best
0.1	0.025	1.25	0.9991	0.9870	1.0000	0.0009	false
10,000	0.125	1.25	0.9991	0.9870	1.0000	0.0009	false
10	0.0025	2.5	0.9991	0.9870	1.0000	0.0009	false
10,000	0.125	2.5	0.9991	0.9870	1.0000	0.0009	false
1	0.0025	-1.25	0.9986	0.9838	1.0000	0.0014	false
1	0.125	-0.5	0.9986	0.9838	1.0000	0.0014	false
1	0.025	0.5	0.9986	0.9838	1.0000	0.0014	false
0.1	0.025	2.5	0.9986	0.9838	1.0000	0.0014	false
1	0.025	2.5	0.9986	0.9838	1.0000	0.0014	false
1	0.125	0	0.9986	0.9838	1.0000	0.0009	false
10	0.125	0	0.9986	0.9838	1.0000	0.0009	false
10	0.00025	-0.5	0.9982	0.9810	1.0000	0.0018	false
1	0.0025	-0.5	0.9982	0.9810	1.0000	0.0018	false
0.1	0.025	-0.5	0.9982	0.9810	1.0000	0.0018	false
10	0.00025	0.5	0.9982	0.9810	1.0000	0.0018	false
1	0.0025	0.5	0.9982	0.9810	1.0000	0.0018	false
1	0.025	1.25	0.9982	0.9810	1.0000	0.0018	false
1,000	0.125	0	0.9982	0.9810	1.0000	0.0014	false
10,000	0.125	0	0.9982	0.9810	1.0000	0.0014	false
0.1	0.125	-2.5	0.9977	0.9786	1.0000	0.0023	false
10,000	0.125	-1.25	0.9977	0.9786	1.0000	0.0023	false
10	0.125	-0.5	0.9977	0.9786	1.0000	0.0023	false
0.1	0.025	0	0.9977	0.9786	1.0000	0.0023	false
0.1	0.025	0.5	0.9977	0.9786	1.0000	0.0023	false
10	0.025	1.25	0.9977	0.9786	1.0000	0.0023	false
10	0.025	2.5	0.9977	0.9786	1.0000	0.0023	false
1,000	0.125	-0.5	0.9977	0.9786	1.0000	0.0018	false
10,000	0.125	-0.5	0.9977	0.9786	1.0000	0.0018	false
0.1	0.125	0	0.9977	0.9786	1.0000	0.0018	false
10	0.00025	0	0.9973	0.9763	1.0000	0.0027	false
1	0.0025	0	0.9973	0.9763	1.0000	0.0027	false
10	0.125	-1.25	0.9973	0.9763	1.0000	0.0023	false
1,000	0.125	-1.25	0.9968	0.9742	1.0000	0.0032	false
10	0.025	0.5	0.9968	0.9742	1.0000	0.0032	false
10	0.0025	1.25	0.9968	0.9742	1.0000	0.0032	false
1,000	0.025	2.5	0.9968	0.9742	1.0000	0.0023	false
0.1	0.125	-1.25	0.9963	0.9721	1.0000	0.0037	false
0.1	0.125	-0.5	0.9963	0.9721	1.0000	0.0037	false
1,000	0.00025	-2.5	0.9959	0.9702	1.0000	0.0041	false
10,000	0.025	1.25	0.9959	0.9702	1.0000	0.0041	false
1,000	0.00025	2.5	0.9959	0.9702	1.0000	0.0041	false

Cost C	γ	Coefficient c_0	Precision	CI Lower	CI Upper	Potential	Best
10,000	0.025	2.5	0.9959	0.9702	1.0000	0.0037	false
10	0.025	-2.5	0.9954	0.9684	1.0000	0.0046	false
10	0.125	-2.5	0.9954	0.9684	1.0000	0.0046	false
10	0.0025	-1.25	0.9954	0.9684	1.0000	0.0046	false
1,000	0.025	1.25	0.9954	0.9684	1.0000	0.0041	false
10	0.0025	0.5	0.9950	0.9666	1.0000	0.0050	false
1,000	0.025	0.5	0.9950	0.9666	1.0000	0.0050	false
10,000	0.025	0.5	0.9950	0.9666	1.0000	0.0046	false
1	0.125	-2.5	0.9945	0.9649	1.0000	0.0055	false
1	0.125	-1.25	0.9945	0.9649	1.0000	0.0055	false
1	0.025	0	0.9945	0.9649	1.0000	0.0055	false
1	0.025	-1.25	0.9941	0.9632	1.0000	0.0059	false
10	0.0025	-0.5	0.9941	0.9632	1.0000	0.0055	false
10	0.0025	0	0.9936	0.9616	1.0000	0.0059	false
1,000	0.0025	2.5	0.9922	0.9570	1.0000	0.0078	false
10,000	0.125	-2.5	0.9918	0.9555	1.0000	0.0078	false
1,000	0.025	0	0.9918	0.9555	1.0000	0.0078	false
1,000	0.125	-2.5	0.9913	0.9541	1.0000	0.0087	false
10,000	0.00025	-2.5	0.9908	0.9527	1.0000	0.0092	false
10	0.025	0	0.9908	0.9527	1.0000	0.0087	false
10,000	0.025	0	0.9908	0.9527	1.0000	0.0087	false
1,000	0.00025	-1.25	0.9904	0.9513	1.0000	0.0096	false
1,000	0.00025	1.25	0.9904	0.9513	1.0000	0.0096	false
10,000	0.0025	2.5	0.9904	0.9513	1.0000	0.0092	false
10,000	0.00025	2.5	0.9899	0.9499	1.0000	0.0101	false
1	0.025	-0.5	0.9899	0.9499	1.0000	0.0096	false
1,000	0.0025	1.25	0.9899	0.9499	1.0000	0.0096	false
10	0.025	-1.25	0.9895	0.9486	1.0000	0.0105	false
1,000	0.00025	-0.5	0.9890	0.9472	1.0000	0.0110	false
1,000	0.00025	0	0.9890	0.9472	1.0000	0.0110	false
1,000	0.00025	0.5	0.9890	0.9472	1.0000	0.0110	false
10,000	0.025	-2.5	0.9890	0.9472	1.0000	0.0105	false
1,000	0.025	-2.5	0.9890	0.9472	1.0000	0.0101	false
10,000	0.0025	1.25	0.9886	0.9459	1.0000	0.0114	false
1,000	0.0025	-2.5	0.9886	0.9459	1.0000	0.0110	false
10,000	0.0025	0.5	0.9872	0.9421	1.0000	0.0124	false
10,000	0.00025	1.25	0.9872	0.9421	1.0000	0.0124	false
1,000	0.0025	0	0.9872	0.9421	1.0000	0.0119	false
1,000	0.0025	0.5	0.9872	0.9421	1.0000	0.0119	false
10	0.025	-0.5	0.9867	0.9409	1.0000	0.0133	false
10,000	0.025	-0.5	0.9867	0.9409	1.0000	0.0128	false

Cost C	γ	Coefficient c_0	Precision	CI Lower	CI Upper	Potential	Best
10,000	0.00025	0	0.9867	0.9409	1.0000	0.0124	false
10,000	0.00025	0.5	0.9867	0.9409	1.0000	0.0124	false
1,000	0.025	-1.25	0.9863	0.9396	1.0000	0.0133	false
10,000	0.025	-1.25	0.9863	0.9396	1.0000	0.0133	false
10,000	0.0025	-0.5	0.9863	0.9396	1.0000	0.0133	false
10,000	0.0025	0	0.9863	0.9396	1.0000	0.0133	false
10,000	0.0025	-2.5	0.9858	0.9384	1.0000	0.0133	false
10,000	0.00025	-1.25	0.9854	0.9372	1.0000	0.0142	false
1,000	0.025	-0.5	0.9854	0.9372	1.0000	0.0142	false
1,000	0.0025	-1.25	0.9849	0.9360	1.0000	0.0142	false
10,000	0.0025	-1.25	0.9849	0.9360	1.0000	0.0142	false
1,000	0.0025	-0.5	0.9849	0.9360	1.0000	0.0142	false
10,000	0.00025	-0.5	0.9844	0.9348	1.0000	0.0146	false
0.1	10	-2.5	0.9817	0.9280	1.0000	0.0000	false
0.1	10	-1.25	0.9817	0.9280	1.0000	0.0000	false
0.1	10	-0.5	0.9817	0.9280	1.0000	0.0000	false
0.1	10	0	0.9817	0.9280	1.0000	0.0000	false
0.1	10	0.5	0.9817	0.9280	1.0000	0.0000	false
0.1	10	1.25	0.9817	0.9280	1.0000	0.0000	false
0.1	10	2.5	0.9817	0.9280	1.0000	0.0000	false

D.8 k Nearest-Neighbors (kNN)

Table D.8 shows the results of the kNN classifiers.

D.9 Aggregated Results

Result Overview

Table D.9: Details for aggregated results represented in figures 7.10 in chapter 7 ordered by Precision and Potential. The results contain the data of 305,900 trained classifiers.

Algorithm	Precision	CI Lower	CI Upper	Potential	Best
SVM-poly	1.000000	1.000000	1.000000	0.000000	true
SVM-radial	1.000000	1.000000	1.000000	0.000000	true
SVM-sigmoid	0.999977	0.998068	1.000000	0.000023	false
c4.5	0.997780	0.979002	1.000000	0.002197	false
1-Nearest-Neighbors (1-NN)	0.997551	0.977831	1.000000	0.000160	false

Algorithm	Precision	CI Lower	CI Upper	Potential	Best
Logistic Regression	0.996064	0.971080	1.000000	0.003524	false
SVM-linear	0.989130	0.947756	1.000000	0.010343	false

Detailed Results per Scenario and Task

Table D.10: Details from the aggregated detailed results. Graphical illustrations can be found in figures 7.12,7.13,7.14, 7.15 and 7.11 in chapter 7 grouped by Scenarios and Tasks. The results contain the data of 305,900 trained classifiers.

Scenarios	Tasks	Algorithms	Precision	CI Lower	CI Upper	Potential
Accomm.	Adult Passangers	Logistic Regression	0.9991	0.9870	1.0000	0.0009
Accomm.	Adult Passangers	1-NN	1.0000	1.0000	1.0000	0.0000
Accomm.	Adult Passangers	c4.5	1.0000	1.0000	1.0000	0.0000
Accomm.	Adult Passangers	SVM-poly	1.0000	1.0000	1.0000	0.0000
Accomm.	Adult Passangers	SVM-radial	1.0000	1.0000	1.0000	0.0000
Accomm.	Adult Passangers	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Accomm.	Adult Passangers	SVM-linear	0.9991	0.9870	1.0000	0.0009
Accomm.	From Date	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Accomm.	From Date	1-NN	0.9986	0.9834	1.0000	0.0014
Accomm.	From Date	c4.5	0.9329	0.8323	1.0000	0.0671
Accomm.	From Date	SVM-poly	1.0000	1.0000	1.0000	0.0000
Accomm.	From Date	SVM-radial	1.0000	1.0000	1.0000	0.0000
Accomm.	From Date	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Accomm.	From Date	SVM-linear	0.9943	0.9640	1.0000	0.0057
Accomm.	Departure Date	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Accomm.	Departure Date	1-NN	1.0000	1.0000	1.0000	0.0000
Accomm.	Departure Date	c4.5	1.0000	1.0000	1.0000	0.0000
Accomm.	Departure Date	SVM-poly	1.0000	1.0000	1.0000	0.0000
Accomm.	Departure Date	SVM-radial	1.0000	1.0000	1.0000	0.0000
Accomm.	Departure Date	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Accomm.	Departure Date	SVM-linear	0.9827	0.9303	1.0000	0.0164
Accomm.	Nights	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Accomm.	Nights	1-NN	1.0000	1.0000	1.0000	0.0000
Accomm.	Nights	c4.5	1.0000	1.0000	1.0000	0.0000
Accomm.	Nights	SVM-poly	1.0000	1.0000	1.0000	0.0000
Accomm.	Nights	SVM-radial	1.0000	1.0000	1.0000	0.0000
Accomm.	Nights	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Accomm.	Nights	SVM-linear	1.0000	1.0000	1.0000	0.0000

Scenarios	Tasks	Algorithms	Precision	CI Lower	CI Upper	Potential
Accomm.	Submit Button	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Accomm.	Submit Button	1-NN	0.9964	0.9725	1.0000	0.0036
Accomm.	Submit Button	c4.5	0.9986	0.9834	1.0000	0.0014
Accomm.	Submit Button	SVM-poly	1.0000	1.0000	1.0000	0.0000
Accomm.	Submit Button	SVM-radial	1.0000	1.0000	1.0000	0.0000
Accomm.	Submit Button	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Accomm.	Submit Button	SVM-linear	0.9857	0.9380	1.0000	0.0143
Accomm.	Where Location	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Accomm.	Where Location	1-NN	0.9286	0.8251	1.0000	0.0000
Accomm.	Where Location	c4.5	0.9993	0.9886	1.0000	0.0000
Accomm.	Where Location	SVM-poly	1.0000	1.0000	1.0000	0.0000
Accomm.	Where Location	SVM-radial	1.0000	1.0000	1.0000	0.0000
Accomm.	Where Location	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Accomm.	Where Location	SVM-linear	0.9721	0.9060	1.0000	0.0264
All	Adult Passangers	Logistic Regression	1.0000	1.0000	1.0000	0.0000
All	Adult Passangers	1-NN	1.0000	1.0000	1.0000	0.0000
All	Adult Passangers	c4.5	1.0000	1.0000	1.0000	0.0000
All	Adult Passangers	SVM-poly	1.0000	1.0000	1.0000	0.0000
All	Adult Passangers	SVM-radial	1.0000	1.0000	1.0000	0.0000
All	Adult Passangers	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
All	Adult Passangers	SVM-linear	1.0000	1.0000	1.0000	0.0000
All	Arrival Location	Logistic Regression	0.9874	0.9427	1.0000	0.0124
All	Arrival Location	1-NN	1.0000	1.0000	1.0000	0.0000
All	Arrival Location	c4.5	1.0000	1.0000	1.0000	0.0000
All	Arrival Location	SVM-poly	1.0000	1.0000	1.0000	0.0000
All	Arrival Location	SVM-radial	1.0000	1.0000	1.0000	0.0000
All	Arrival Location	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
All	Arrival Location	SVM-linear	1.0000	1.0000	1.0000	0.0000
All	Departure Date	Logistic Regression	0.9816	0.9278	1.0000	0.0179
All	Departure Date	1-NN	1.0000	1.0000	1.0000	0.0000
All	Departure Date	c4.5	1.0000	1.0000	1.0000	0.0000
All	Departure Date	SVM-poly	1.0000	1.0000	1.0000	0.0000
All	Departure Date	SVM-radial	1.0000	1.0000	1.0000	0.0000
All	Departure Date	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
All	Departure Date	SVM-linear	0.9618	0.8852	1.0000	0.0382
All	Departure Location	Logistic Regression	0.9997	0.9932	1.0000	0.0003
All	Departure Location	1-NN	1.0000	1.0000	1.0000	0.0000
All	Departure Location	c4.5	1.0000	1.0000	1.0000	0.0000
All	Departure Location	SVM-poly	1.0000	1.0000	1.0000	0.0000
All	Departure Location	SVM-radial	1.0000	1.0000	1.0000	0.0000
All	Departure Location	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000

Scenarios	Tasks	Algorithms	Precision	CI Lower	CI Upper	Potential
All	Departure Location	SVM-linear	1.0000	1.0000	1.0000	0.0000
All	One-Way	Logistic Regression	1.0000	1.0000	1.0000	0.0000
All	One-Way	1-NN	1.0000	1.0000	1.0000	0.0000
All	One-Way	c4.5	1.0000	1.0000	1.0000	0.0000
All	One-Way	SVM-poly	1.0000	1.0000	1.0000	0.0000
All	One-Way	SVM-radial	1.0000	1.0000	1.0000	0.0000
All	One-Way	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
All	One-Way	SVM-linear	1.0000	1.0000	1.0000	0.0000
All	Submit Button	Logistic Regression	0.9953	0.9678	1.0000	0.0047
All	Submit Button	1-NN	1.0000	1.0000	1.0000	0.0000
All	Submit Button	c4.5	1.0000	1.0000	1.0000	0.0000
All	Submit Button	SVM-poly	1.0000	1.0000	1.0000	0.0000
All	Submit Button	SVM-radial	1.0000	1.0000	1.0000	0.0000
All	Submit Button	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
All	Submit Button	SVM-linear	1.0000	1.0000	1.0000	0.0000
Bus	Arrival Location	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Bus	Arrival Location	1-NN	1.0000	1.0000	1.0000	0.0000
Bus	Arrival Location	c4.5	1.0000	1.0000	1.0000	0.0000
Bus	Arrival Location	SVM-poly	1.0000	1.0000	1.0000	0.0000
Bus	Arrival Location	SVM-radial	1.0000	1.0000	1.0000	0.0000
Bus	Arrival Location	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Bus	Arrival Location	SVM-linear	1.0000	1.0000	1.0000	0.0000
Bus	Departure Date	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Bus	Departure Date	1-NN	1.0000	1.0000	1.0000	0.0000
Bus	Departure Date	c4.5	1.0000	1.0000	1.0000	0.0000
Bus	Departure Date	SVM-poly	1.0000	1.0000	1.0000	0.0000
Bus	Departure Date	SVM-radial	1.0000	1.0000	1.0000	0.0000
Bus	Departure Date	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Bus	Departure Date	SVM-linear	0.9946	0.9652	1.0000	0.0054
Bus	Departure Location	Logistic Regression	0.9992	0.9881	1.0000	0.0008
Bus	Departure Location	1-NN	1.0000	1.0000	1.0000	0.0000
Bus	Departure Location	c4.5	1.0000	1.0000	1.0000	0.0000
Bus	Departure Location	SVM-poly	1.0000	1.0000	1.0000	0.0000
Bus	Departure Location	SVM-radial	1.0000	1.0000	1.0000	0.0000
Bus	Departure Location	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Bus	Departure Location	SVM-linear	1.0000	1.0000	1.0000	0.0000
Bus	One-Way	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Bus	One-Way	1-NN	1.0000	1.0000	1.0000	0.0000
Bus	One-Way	c4.5	1.0000	1.0000	1.0000	0.0000
Bus	One-Way	SVM-poly	1.0000	1.0000	1.0000	0.0000
Bus	One-Way	SVM-radial	1.0000	1.0000	1.0000	0.0000

Scenarios	Tasks	Algorithms	Precision	CI Lower	CI Upper	Potential
Bus	One-Way	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Bus	One-Way	SVM-linear	1.0000	1.0000	1.0000	0.0000
Bus	Submit Button	Logistic Regression	0.9992	0.9881	1.0000	0.0008
Bus	Submit Button	1-NN	1.0000	1.0000	1.0000	0.0000
Bus	Submit Button	c4.5	1.0000	1.0000	1.0000	0.0000
Bus	Submit Button	SVM-poly	1.0000	1.0000	1.0000	0.0000
Bus	Submit Button	SVM-radial	1.0000	1.0000	1.0000	0.0000
Bus	Submit Button	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Bus	Submit Button	SVM-linear	0.9892	0.9477	1.0000	0.0108
Flight	Adult Passangers	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Flight	Adult Passangers	1-NN	1.0000	1.0000	1.0000	0.0000
Flight	Adult Passangers	c4.5	1.0000	1.0000	1.0000	0.0000
Flight	Adult Passangers	SVM-poly	1.0000	1.0000	1.0000	0.0000
Flight	Adult Passangers	SVM-radial	1.0000	1.0000	1.0000	0.0000
Flight	Adult Passangers	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Flight	Adult Passangers	SVM-linear	0.9977	0.9784	1.0000	0.0023
Flight	Arrival Location	Logistic Regression	0.9985	0.9827	1.0000	0.0015
Flight	Arrival Location	1-NN	1.0000	1.0000	1.0000	0.0000
Flight	Arrival Location	c4.5	1.0000	1.0000	1.0000	0.0000
Flight	Arrival Location	SVM-poly	1.0000	1.0000	1.0000	0.0000
Flight	Arrival Location	SVM-radial	1.0000	1.0000	1.0000	0.0000
Flight	Arrival Location	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Flight	Arrival Location	SVM-linear	0.9938	0.9624	1.0000	0.0062
Flight	Departure Date	Logistic Regression	0.9977	0.9784	1.0000	0.0023
Flight	Departure Date	1-NN	1.0000	1.0000	1.0000	0.0000
Flight	Departure Date	c4.5	1.0000	1.0000	1.0000	0.0000
Flight	Departure Date	SVM-poly	1.0000	1.0000	1.0000	0.0000
Flight	Departure Date	SVM-radial	1.0000	1.0000	1.0000	0.0000
Flight	Departure Date	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Flight	Departure Date	SVM-linear	0.9954	0.9681	1.0000	0.0046
Flight	Departure Location	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Flight	Departure Location	1-NN	1.0000	1.0000	1.0000	0.0000
Flight	Departure Location	c4.5	1.0000	1.0000	1.0000	0.0000
Flight	Departure Location	SVM-poly	1.0000	1.0000	1.0000	0.0000
Flight	Departure Location	SVM-radial	1.0000	1.0000	1.0000	0.0000
Flight	Departure Location	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Flight	Departure Location	SVM-linear	1.0000	1.0000	1.0000	0.0000
Flight	One-Way	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Flight	One-Way	1-NN	1.0000	1.0000	1.0000	0.0000
Flight	One-Way	c4.5	1.0000	1.0000	1.0000	0.0000
Flight	One-Way	SVM-poly	1.0000	1.0000	1.0000	0.0000

Scenarios	Tasks	Algorithms	Precision	CI Lower	CI Upper	Potential
Flight	One-Way	SVM-radial	1.0000	1.0000	1.0000	0.0000
Flight	One-Way	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Flight	One-Way	SVM-linear	1.0000	1.0000	1.0000	0.0000
Flight	Submit Button	Logistic Regression	0.9792	0.9219	1.0000	0.0092
Flight	Submit Button	1-NN	1.0000	1.0000	1.0000	0.0000
Flight	Submit Button	c4.5	1.0000	1.0000	1.0000	0.0000
Flight	Submit Button	SVM-poly	1.0000	1.0000	1.0000	0.0000
Flight	Submit Button	SVM-radial	1.0000	1.0000	1.0000	0.0000
Flight	Submit Button	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Flight	Submit Button	SVM-linear	0.9954	0.9681	1.0000	0.0046
Train	Arrival Location	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Train	Arrival Location	1-NN	1.0000	1.0000	1.0000	0.0000
Train	Arrival Location	c4.5	1.0000	1.0000	1.0000	0.0000
Train	Arrival Location	SVM-poly	1.0000	1.0000	1.0000	0.0000
Train	Arrival Location	SVM-radial	1.0000	1.0000	1.0000	0.0000
Train	Arrival Location	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Train	Arrival Location	SVM-linear	0.9792	0.9217	1.0000	0.0183
Train	Departure Date	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Train	Departure Date	1-NN	1.0000	1.0000	1.0000	0.0000
Train	Departure Date	c4.5	1.0000	1.0000	1.0000	0.0000
Train	Departure Date	SVM-poly	1.0000	1.0000	1.0000	0.0000
Train	Departure Date	SVM-radial	1.0000	1.0000	1.0000	0.0000
Train	Departure Date	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Train	Departure Date	SVM-linear	0.8783	0.7468	1.0000	0.1092
Train	Departure Location	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Train	Departure Location	1-NN	1.0000	1.0000	1.0000	0.0000
Train	Departure Location	c4.5	1.0000	1.0000	1.0000	0.0000
Train	Departure Location	SVM-poly	1.0000	1.0000	1.0000	0.0000
Train	Departure Location	SVM-radial	1.0000	1.0000	1.0000	0.0000
Train	Departure Location	SVM-sigmoid	0.9992	0.9876	1.0000	0.0008
Train	Departure Location	SVM-linear	0.9867	0.9405	1.0000	0.0133
Train	Submit Button	Logistic Regression	1.0000	1.0000	1.0000	0.0000
Train	Submit Button	1-NN	1.0000	1.0000	1.0000	0.0000
Train	Submit Button	c4.5	1.0000	1.0000	1.0000	0.0000
Train	Submit Button	SVM-poly	1.0000	1.0000	1.0000	0.0000
Train	Submit Button	SVM-radial	1.0000	1.0000	1.0000	0.0000
Train	Submit Button	SVM-sigmoid	1.0000	1.0000	1.0000	0.0000
Train	Submit Button	SVM-linear	0.9900	0.9500	1.0000	0.0083

Table D.8: Details from Figure 7.5 in chapter 7. 26,220 runs

		Accommodation		All Connections		Bus		Flight		Train	
K	Type	PCA	Z-Score	PCA	Z-Score	PCA	Z-Score	PCA	Z-Score	PCA	Z-Score
1	Prec	0.9824	0.9853	0.9979	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI L	0.9283	0.9359	0.9792	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI U	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	Prec	0.9824	0.9853	0.9989	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI L	0.9283	0.9359	0.9858	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI U	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
7	Prec	0.9824	0.9824	0.9989	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI L	0.9283	0.9283	0.9858	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI U	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
15	Prec	0.9735	0.9794	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI L	0.9076	0.9211	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI U	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
31	Prec	0.9265	0.9235	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI L	0.8193	0.8144	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI U	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
63	Prec	0.8824	0.8824	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI L	0.7501	0.7501	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	CI U	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

APPENDIX E

Statical Background

This chapter complements other parts of the thesis with more in-depth information since it would be not practical to have it in the respective chapters.

E.1 Boxplot

A boxplot is a graph with which it is possible to illustrate a sample in a simple but informative way. Furthermore, it reduces the information in such a way, that it is easier to draw conclusions to the underlying distribution of the data (in comparison with a histogram or other visual devices).

Figure E.1 shows such a boxplot. This boxplot visualize some statistic ratios, namely the median, the interquartile range (IQR), the 25% as well as the 75% quantile. Moreover, it is straightforward to find outliers in the sample as the are shown as circles in the plot. The definition of an outlier is defined in equation (E.1). The dashed lines indicate values below the 25% quantile respectively above the 75% quantile.

A sample which is drawn from a normally distributed population should have only a small number of outliers. In addition, the difference of the length between the 25% quantile and the median on the one hand side and the length between the median and the 75% quantile on the other side should be small. Furthermore, the dashed lines length should also be similar. In general, both sides should be more or less symmetric if the data is normally distributed.

$$Outlier(x) = \begin{cases} true & \text{if } x < quantile_{0.25} - IQR * 1.5 \text{ OR} \\ & \quad x > quantile_{0.75} + IQR * 1.5 \\ false & \text{else} \end{cases} \tag{E.1}$$

E.2 Correlation

Correlation is a measurement between two variables, which tries to explain their connection. In general, when someone refers to terms *correlation* or *correlation coefficient* he or she means the

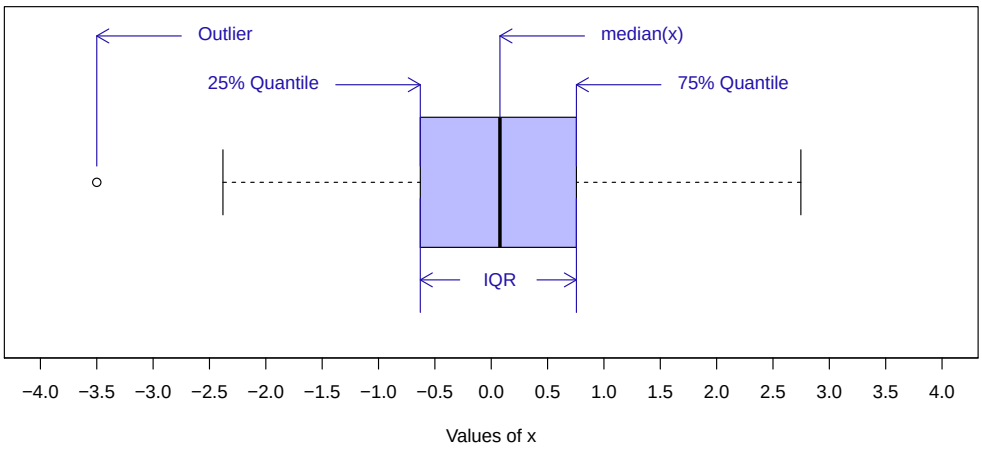


Figure E.1: A Boxplot with description

linear correlation coefficient. As this is the common usage, the author also refers to the linear correlation coefficient as long as not mentioned otherwise. It is often denoted by the Greek letter ρ[1]

The subsections below are dealing with different numeric calculations of correlation. However, what all linear correlation coefficient have in common is, that their domaindomain lies between -1 and 1. 1 indicates that the two variables are perfectly positively correlated, whereas -1 means that this variables are perfectly negatively correlated. If the absolute value of the correlation coefficient is higher than a certain threshold it is very likely that the two populations depend on each other (This value needs to be chosen individually for each application, but in general the author would recommend a value of 0.75.) A common error is, that the reverse is assumed in the case of a correlation which is almost or exactly 0. With other words, drawing the conclusion that two variables are independent by observing a correlation of 0 is wrong.

Pearson Correlation

Pearson's Correlation is the most popular measure to estimate the linear correlation coefficient. The formula for estimating the Pearson correlations for a population($Z = N$) or a sample ($Z = N - 1$) is shown in equation (E.2).

$$\hat{\rho}_{x,y} = \frac{1}{Z} * \frac{\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y} \tag{E.2}$$

[1]Sometimes it is also defined as the function $cor(x, y)$. However, in this master thesis Greek letters are used as they are shorter and as the author believes more often used. In this case ρ_{xy} is equal to $cor(x, y)$.

Spearman

Sometimes Pearson's correlation coefficient is influenced by a few outliers in the data set and result into a value too small or too big. A very simple technique to avoid that and concentrate on most of the data is Spearman's correlations coefficient. The equation (E.3) is similar to the equation (E.2). The only difference is, that the values x_i and y_i have been transformed by its rank as a result μ_x and μ_y is $\frac{N(N+1)}{2N} = \frac{N+1}{2}$. Further information can be found at [49]

$$\hat{\rho}_{x,y} = \frac{1}{Z} * \frac{\sum_{i=1}^{N}(rank(x_i) - \mu_x)(rank(y_i) - \mu_y)}{\sigma_x \sigma_y} \tag{E.3}$$

Connection to the Covariance

The connection between the linear correlation and the covariance is shown in equation (E.4). It means that the correlation of two variables is equal to the covariance of these variables divided by the product of the standard deviations of each variable.

$$\rho_{x,y} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{E.4}$$

E.3 Covariance

The covariance is a measure of the correlation of two random numbers. Its formal definition is shown in equation (E.5). For estimating the covariance from a sample the Pearson's version is shown at (E.6) where the Spearman's version is found at (E.7). For further information see [86] or [87].

$$var(x, y) = \sigma_{x,y} = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y] \tag{E.5}$$

$$\sigma_{x,y} = \frac{1}{Z} * \sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{E.6}$$

$$\sigma_{x,y} = \frac{1}{Z} * \sum_{i=1}^{N}(rank(x_i) - \mu_x)(rank(y_i) - \mu_y) \tag{E.7}$$

E.4 Level of Measurement

In general, statisticians distinguish between four kind of measures. For further details see [69].

- ***Categorical***: Variables of this type are not numerical and it is not possible to order the values. Categorical variables are qualitative. Examples are gender, nationality and color. The proposed measure of central tendency is the *mode*.

- ***Ordinal***: These variables are also not numerical and are considered as qualitative, however it is possible to order them. Examples are grades and ranks in the army. For this scale, the *median* provides an adequate measure for central tendency.

- ***Interval***: Interval scale variables are numerical and therefore called quantitative. Nevertheless it is not possible to judge about the ratio of two values, only their differences (e.g. for a variable holding the temperature in degrees Celsius it is not possible to say, that x_1 is half as hot as x_2, but that there is differences of 5 degrees.) This results from the fact, that the zero value is artificial. The *arithmetic mean* can be used for measuring the central tendency.

- ***Ratio***: As above, the values are also numerical and therefore considered as quantitative. However, for the ratio scale it is possible to measure the ratio between two observations within a feature. Examples are the temperature in degrees Kelvin, length, weight and most numerical data. Within a ratio scaled variable, it is possible to use the *geometric mean* as measure for central tendency.

For every new category, a new measurement for central tendency has been introduced. Nevertheless, for every new category it is possible, that the above used measures of central tendency can be used as well (e.g. the interval scale can use the median, mode and arithmetic mean, but not the geometric mean). As a result it might be interesting that grade point averages are quite common to compare students performances with each other, even that is not methodically correct, since the differences between the grades are not equal.

E.5 Mean

The arithmetic mean of a sample is commonly used to estimate the expected value of a population. Sometimes it is also referred to as the first momentum. Equation (E.8) shows how the arithmetic mean is defined, where n is the number of the different elements for this variable x. The mean is often abbreviated by the letter μ.

$$\mu_x = \frac{1}{n} * \sum_{i=1}^{n} x_i \tag{E.8}$$

What might be misunderstood is the difference between the mean of a sample and that of a population. In most cases the mean of a sample is calculated as it is very unlikely to obtain the data of the whole population. It is common practice to use $\hat{\mu}$ for indicating that the mean was calculated from the sample and is an estimator for the mean of the population. In case it is calculated directly from the population the ”hat” is omitted.

E.6 Median

The median is that value which lies exactly in the middle of the ordered list of all elements for a certain variable. Formally spoken, it is the n^{th} element of an ordered data vector in case the

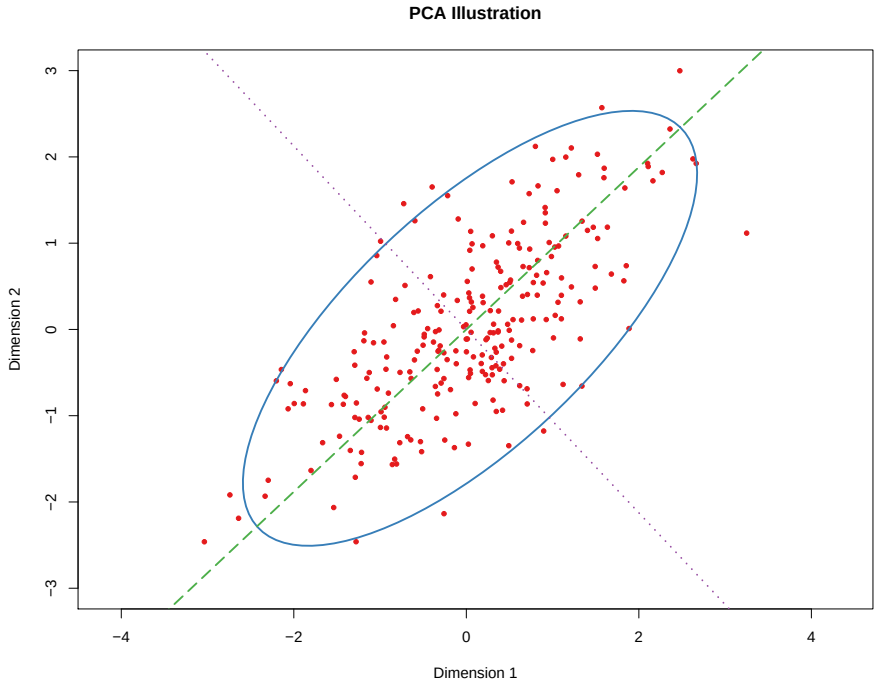


Figure E.2: Illustration of the PCA

number of elements is odd. If the number of elements is even, the median is the arithmetic mean of the $(\frac{n}{2} - 0.5)^{th}$ and $(\frac{n}{2} + 0.5)^{th}$ elements. The median is equal to the 50% quantile.

E.7 Principal Component Analyses

The Principal Component Analysis (PCA) is considered as a fundamental technique for analysing multivariate data. Its main purpose is to reduce the dimension of a data set. This is done by replacing the data with a linear combination of the data. In addition, the resulting dimensions are uncorrelated to each other. This is an desirable feature, since some statistical methods fail if there are linear dependences.

The main idea behind the PCA to optimize the weights of the linear combination in such a way, that the variance of the resulting first principal component is maximized. Then the next (second) principal component lies orthogonal on the first and further maximizes the overall variance. This continues until all dimensions are processed.

Figure E.2 gives a graphical illustration of the data x,y would be mapped to a new resulting coordinate system (first and second principal component).

Further information can be found at [53].

E.8 Standard Deviation

The standard deviation is a measure of volatility within a variable. It is often denoted as σ. It is closely connected to the variance since the $var(x) = \sigma^2$. Therefore, by taking the square root of equation (E.10) for calculating the variance, it is possible to compute the standard deviation.

E.9 Common Transformations

For many ML algorithms it is crucial to transform the input data. The term transformation represents most often a simple function. The original input data x is transformed by a function f which then can be seen as new input data y. Formally, this can be written as $y = f(x)$. Transformation is applied variable-wise and not observation wise. For an input matrix this means column-wise, as it is common practice that the different observation are added row-wise to a matrix. The next subsections will introduce some functions for transforming the original input data.

Z-Transformation or Standardization

The z-score transformation is also known as standardization, because it standardizes the different variables of the input data in such a way that the empirical mean equals 0 and the standard deviation becomes 1. The formula can be found in equation (E.9), where z_i stands for an element i of transformed data vector z[2]. This transformation is applied to a data vector x with the size n where $1 \leq i \leq n$.

$$z_i = \frac{x_i - \hat{\mu}_x}{\hat{\sigma}_x} \tag{E.9}$$

Rank Transformation

The value of the observation is simply transformed by its rank. The relative rank is obtained by dividing the rank by the number of observations.

E.10 Variance

The variance is a measure of deviation for an univariate random variable. The variance is the standard deviation to the power of 2 and therefore denoted as σ^2. The unit of variance is the unit of the random variable to the power of two. This mean, if the random variable X is in unit US-Dollar, than the variance is in US-Dollar^2. Since it seem unusual to work with ^{2} units, the standard deviation would be the more intuitive metric. A formula for calculating the variance

[2] In contrast to above, here the letter z (instead of y) is used, because it is more common in the literature to refer to z-score with the letter z.

is shown in equation (E.10), where $Z = N$ for the population and $Z = N - 1$ for a sample. Further information can be found in Papoulis et al. [77].

$$var(x) = \sigma_x^2 = \frac{1}{Z} \sum_{i=1}^{N} (x_i - \hat{\mu}_x)^2 \tag{E.10}$$

Bibliography

[1] ABBA — Advanced Barrier-free Browser Accessibility. FFG Fit-IT Project 819563, 2009–2010. http://www.dbai.tuwien.ac.at/proj/ABBA/.

[2] TAMCROW — TAsk Mining and CROWd sourcing. FFG Fit-IT Project 829614, 2011–2012. http://www.dbai.tuwien.ac.at/proj/tamcrow/.

[3] Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification (W3C Recommendation 07 June 2011), 2011.

[4] Milton Abramowitz and Irene Stegun. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. Dover Publications, 1965.

[5] Alan Agresti. *Categorical data analysis*, volume 359. Wiley-interscience, 2002.

[6] Edgar Anderson. The irises of the gaspe peninsula. *Bulletin of the American Iris society*, 59:2–5, 1935.

[7] George B Arfken, Hans J Weber, and Frank E. Harris. *Mathematical Methods For Physicists: A Comprehensive Guide*. Academic Press, 6 edition, 2005.

[8] Philippe Balbiani, Jean-François Condotta, and Luis Fariñas Del Cerro. Tractability Results in the Block Algebra. *Journal of Logic and Computation*, 12(5):885–909, October 2002.

[9] Robert Baumgartner, Sergio Flesca, and Georg Gottlob. Visual web information extraction with lixto. In *Proceedings of the international conference on very large data bases*, pages 119–128, 2001.

[10] Asa Ben-Hur and Jason Weston. A userś guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer, 2010.

[11] Christopher M. Bishop. *Pattern recognition and machine learning*, volume 1. Springer, 2006.

[12] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. *Classification and Regression Trees*. 1984.

[13] Lawrence D Brown, T Tony Cai, and Anirban DasGupta. Interval estimation for a binomial proportion. *Statistical Science*, pages 101–117, 2001.

[14] Ahmad C. Bukhari and Yong-Gi Kim. Ontology-assisted automatic precise information extractor for visually impaired inhabitants. *Artificial Intelligence Review*, 38(1):9–24, 2012.

[15] Daniel Cabeza and Manuel Hermenegildo. Distributed WWW programming using(Ciao-) Prolog and the PiLLoW library. *Theory and Practice of Logic Programming*, 1(3):251–282, 2001.

[16] C-H Chang, Mohammed Kayed, R Girgis, and Khaled F Shaalan. A survey of web information extraction systems. *Knowledge and Data Engineering, IEEE Transactions on*, 18(10):1411–1428, 2006.

[17] Y.H. Chen, S.S. Li, and Y.T. Chen. Extracting Topics Information from Conference Web Pages Using Page Segmentation and SVM. In *Technologies and Applications of Artificial Intelligence (TAAI), 2010 International Conference on*, pages 270–277. IEEE, 2010.

[18] Anthony G. Cohn. Qualitative spatial representation and reasoning techniques. In Gerhard Brewka, Christopher Habel, and Bernhard Nebel, editors, *KI-97: Advances in Artificial Intelligence*, volume 1303, pages 1–30. Springer Berlin, Berlin, Germany, May 1997.

[19] Christian Convey, O Karpenko, and Nesime Tatbul. Data integration services, 2001.

[20] Richard Courant and Herbert Robbins. *What is Mathematics?: an elementary approach to ideas and methods*. Oxford University Press, USA, 2 edition, 1996.

[21] V. Crescenzi, G. Mecca, P. Merialdo, et al. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of the international conference on very large data bases*, pages 109–118, 2001.

[22] Fred J. Damerau. A technique for computer detection and correction of spelling errors. *Commun. of the ACM*, 7(3):171–176, March 1964.

[23] S.B. Dong. The hierarchical classification of Web content by the combination of textual and visual features. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, volume 3, pages 1524–1529. IEEE, 2004.

[24] Eduard C. Dragut, Thomas Kabisch, Clement Yu, and Ulf Leser. A hierarchical approach to model web query interfaces for web source integration. In *Proc. of VLDB Endowment*, volume 2, pages 325–336. VLDB, 2009.

[25] David W Embley, Douglas M Campbell, Yuan S Jiang, Stephen W Liddle, Deryle W Lonsdale, Y-K Ng, and Randy D Smith. Conceptual-model-based data extraction from multiple-record Web pages. *Data & Knowledge Engineering*, 31(3):227–251, 1999.

[26] Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working set selection using second order information for training support vector machines. *The Journal of Machine Learning Research*, 6:1889–1918, 2005.

[27] Ruslan R. Fayzrakhmanov. WPPS: A framework for web page processing. In X. Sean Wang, Isabel Cruz, Alex Delis, and Guangyan Huang, editors, *In Proceedings of the 13th International Conference on Web Information Systems Engineering (WISE'2012), Demo Session, Paphos, Cyprus, 28–30 November, 2012*, pages 800–803. Springer, 2012.

[28] Ruslan R. Fayzrakhmanov. WPPS: A novel and comprehensive framework for web page understanding and information extraction. In Bebo White and Pedro Isaías, editors, *Proceeding of the International Conference IADIS WWW/Internet, Madrid, Spain, 18–21 October, 2012*, pages 19–26, Madrid, 2012. IADIS Press.

[29] Ruslan R. Fayzrakhmanov, Max C. Göbel, Wolfgang Holzinger, Bernhard Krüpl, and Robert Baumgartner. A unified ontology-based web page model for improving accessibility. In *Proceedings of the 19th international conference on World Wide Web (WWW'2010), Raleigh, USA, April 26-30,2010*, pages 1087–1088, New York, 2010. ACM.

[30] Ruslan R. Fayzrakhmanov, Christoph Herzog, and Iraklis Kordomatis. Web objects identification for web automation: objects and their features. Technical report DBAI-TR-2012-80, Institute of Information Systems, TU Vienna, Vienna, 2012.

[31] Bettina Fazzinga, Sergio Flesca, Andrea Tagarelli, Salvatore Garruzzo, and Elio Masciari. A wrapper generation system for PDF documents. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 442–446. ACM, 2008.

[32] Emilio Ferrara and Robert Baumgartner. Automatic wrapper adaptation by tree edit distance matching. In Ioannis Hatzilygeroudis and Jim Prentzas, editors, *Combinations of Intelligent Methods and Applications*, volume 8 of *Smart Innovation, Systems and Technologies*, pages 41–54. Springer Berlin Heidelberg, 2011.

[33] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[34] Sergio Flesca, Salvatore Garruzzo, Elio Masciari, and Andrea Tagarelli. Wrapping pdf documents exploiting uncertain knowledge. In *Advanced Information Systems Engineering*, pages 175–189. Springer, 2006.

[35] David Freedman, Robert Pisani, and Roger Purves. *Statistics*. Norton, 4 edition, 2010.

[36] Tim Furche, Georg Gottlob, Giovanni Grasso, Omer Gunes, Xiaoanan Guo, Andrey Kravchenko, Giorgio Orsi, Christian Schallhart, Andrew Sellers, and Cheng Wang. Diadem: domain-centric, intelligent, automated data extraction methodology. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 267–270. ACM, 2012.

[37] Tim Furche, Georg Gottlob, Giovanni Grasso, Xiaonan Guo, Giorgio Orsi, and Christian Schallhart. OPAL: Automated form understanding for the deep web. In *Proc. of WWW 2012*, pages 829–838, New York, 2012. ACM.

[38] Tim Furche, Georg Gottlob, and Christian Schallhart. DIADEM: Domains to Databases. In *Database and Expert Systems Applications*, pages 1–8. Springer, 2012.

[39] Wolfgang Gatterbauer, Bernhard Krüpl, Wolfgang Holzinger, and Marcus Herzog. Web information extraction using eupeptic data in Web tables. *Proc. RAWS*, pages 41–48, 2005.

[40] M. Hammami, Y. Chahir, and L. Chen. Webguard: A web filtering engine combining textual, structural, and visual content-based analysis. *Knowledge and Data Engineering, IEEE Transactions on*, 18(2):272–284, 2006.

[41] John W Harris and Horst Stöcker. *Handbook of mathematics and computational science*. Springer, 1998.

[42] Tamir Hassan. *User-Guided Information Extraction from Print-Oriented Documents*. PhD thesis, Citeseer, 2010.

[43] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009.

[44] Hai He, Weiyi Meng, Clement Yu, and Zonghuan Wu. Automatic integration of web search interfaces with wise-integrator. *The VLDB Journal*, 13(3):256–273, 2004.

[45] Christoph Herzog, Iraklis Kordomatis, Wolfgang Holzinger, Ruslan R. Fayzrakhmanov, and Bernhard Krüpl-Sypien. Feature-based object identification for web automation. In *Proc. of the 28th Annual ACM SAC'13*, pages 742–749, Coimbra, Portugal, 2013. ACM.

[46] A.R. Hevner. The three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19(2):87–92, 2007.

[47] A.R. Hevner and S.T. March. The information systems research cycle. *Computer*, 36(11):111–113, 2003.

[48] A.R. Hevner, S.T. March, J. Park, and S. Ram. Design science in information systems research. *MIS quarterly*, 28(1):75–105, 2004.

[49] Robert V. Hogg and Allen Craig. Introduction to mathematical statistics, 1994.

[50] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`, apr 2010.

[51] Gerald Huck, Peter Frankhauser, Karl Aberer, and Erich Neuhold. Jedi: Extracting and synthesizing information from the web. In *Cooperative Information Systems, 1998. Proceedings. 3rd IFCIS International Conference on*, pages 32–41. IEEE, 1998.

[52] Earl B Hunt, Janet Marin, and Philip J Stone. Experiments in induction. 1966.

[53] Ian T Jolliffe. *Principal component analysis*. Springer verlag, 2002.

[54] J. Kang and J. Choi. Detecting informative web page blocks for efficient information extraction using visual block segmentation. In *Information Technology Convergence, 2007. ISITC 2007. International Symposium on*, pages 306–310. IEEE, 2007.

[55] Alexandros Karatzoglou, David Meyer, and Kurt Hornik. Support Vector Machines in R. *Journal of Statistical Software*, 15(i09), apr 2006.

[56] Ritu Khare and Yuan An. An empirical study on using hidden markov model for search interface segmentation. In *Proc. of the 18th ACM CIKM '09*, page 17, New York, 2009. ACM.

[57] Donald Knuth. Section 5.2. 4: Sorting by merging. *The Art of Computer Programming*, 3:158–168, 1998.

[58] Iraklis Kordomatis, Christoph Herzog, Ruslan R. Fayzrakhmanov, Bernhard Krüpl-Sypien, Wolfgang Holzinger, and Robert Baumgartner. Web object identification for web automation and meta-search. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, WIMS '13, pages 13:1–13:12. ACM, 2013.

[59] Bernhard Krüpl, Marcus Herzog, and Wolfgang Gatterbauer. Using visual cues for extraction of tabular data from arbitrary HTML documents. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1000–1001. ACM, 2005.

[60] Bernhard Krüpl-Sypien, Ruslan R Fayzrakhmanov, Wolfgang Holzinger, Mathias Panzenböck, and Robert Baumgartner. A versatile model for web page representation, information extraction and content re-packaging. In *Proceedings of the 11th ACM symposium on Document engineering*, pages 129–138. ACM, 2011.

[61] N. Kushmerick, B. Grace, et al. The wrapper induction environment. In *Workshop on Software Tools for Developing Agents, AAAI*, volume 98, 1998.

[62] Alberto HF Laender, Berthier Ribeiro-Neto, and Altigran S da Silva. DEByE–data extraction by example. *Data & Knowledge Engineering*, 40(2):121–154, 2002.

[63] King-Lup Liu, Weiyi Meng, Jing Qiu, Clement Yu, Vijay Raghavan, Zonghuan Wu, Yiyao Lu, Hai He, and Hongkun Zhao. AllInOneNews: development and evaluation of a large-scale news metasearch engine. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 1017–1028, New York, NY, USA, 2007. ACM.

[64] L. Liu, C. Pu, and W. Han. Xwrap: An xml-enabled wrapper construction system for web information sources. In *Data Engineering, 2000. Proceedings. 16th International Conference on*, pages 611–621. IEEE, 2000.

[65] Jalal U. Mahmud, Yevgen Borodin, and I. V. Ramakrishnan. Csurf: a context-driven non-visual web-browser. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 31–40, New York, NY, USA, 2007. ACM.

[66] Ian C Marschner. glm2: Fitting generalized linear models with convergence problems. *The R Journal*, 3(2):12–15, 2011.

[67] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 117–128. IEEE, 2002.

[68] David Meyer. Support vector machines: The interface to libsvm in package E1071. 2010.

[69] Joel Michell. Quantitative methods in psychology. *Psychological bulletin*, 100(3):398–407, 1986.

[70] John Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine learning*, 3(4):319–342, 1989.

[71] I. Muslea, S. Minton, and C. Knoblock. Stalker: Learning extraction rules for semistructured, web-based information sources. In *Proceedings of AAAI-98 Workshop on AI and Information Integration*, pages 74–81, 1998.

[72] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003.

[73] Isabel Navarrete and Guido Sciavicco. Spatial Reasoning with Rectangular Cardinal Direction. In *Proceedings of the ECAI 2006 Workshop on Spatial and Temporal Reasoning*, pages 1–9, 2006.

[74] C.K. Nguyen, L. Likforman-Sulem, J.C. Moissinac, C. Faure, and J. Lardon. Web document analysis based on visual segmentation and page rendering. In *2012 10th IAPR International Workshop on Document Analysis Systems*, pages 354–358. IEEE, 2012.

[75] Hoa Nguyen, Thanh Nguyen, and Juliana Freire. Learning to extract form labels. *Proc. of the VLDB Endowment*, 1(1):684–694, 2008.

[76] Shigeyuki Oba, Masa-aki Sato, Ichiro Takemasa, Morito Monden, Ken-ichi Matsubara, and Shin Ishii. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096, 2003.

[77] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, Random variables and stochastic processes*. McGraw-Hill, 4 edition, 2002.

[78] Daryl Pregibon. Logistic regression diagnostics. *The Annals of Statistics*, 9:705–724, 1981.

[79] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[80] J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.

[81] John Ross Quinlan. *C4. 5: Programs for Machine Learning*, volume 1. Morgan kaufmann, 1993.

[82] Juan Raposo, Alberto Pan, Manuel Álvarez, Justo Hidalgo, and A Vina. The wargo system: Semi-automatic wrapper generation in presence of complex data access modes. In *Database and Expert Systems Applications, 2002. Proceedings. 13th International Workshop on*, pages 313–317. IEEE, 2002.

[83] Brian D. Ripley. *Pattern Recognition and Neural Networks*, volume 1. Cambridge University Press, 1996.

[84] Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

[85] Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. In *Journal on Data Semantics IV*, pages 146–171. Springer, 2005.

[86] G. W. Snedecor and Cochran W. G. *Statistical Methods*. Iowa State University Press, 8 edition, 1989.

[87] Murray R. Spiegel and Larry J. Stephens. *Schaumś Outline of Statistics*. McGraw-Hill, 3 edition, 1998.

[88] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[89] Vladimir Vapnik. Statistical learning theory, 1998.

[90] W. N. Venables and Brian D. Ripley. *Modern Applied Statistics with S*, volume 4. Springer New York, 2010.

[91] R. Vulanović and R. Köhler. Syntactic units and structures. In *Quantitative Linguistics*, pages 274–291. de Gruyter, Berlin, 2005.

[92] Eric P Xing, Michael I Jordan, Richard M Karp, et al. Feature selection for high-dimensional genomic microarray data. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 601–608. Citeseer, 2001.

[93] Yanhong Zhai and Bing Liu. Web data extraction based on partial tree alignment. In *Proceedings of the 14th international conference on World Wide Web*, pages 76–85. ACM, 2005.

[94] Zhen Zhang, Bin He, and Kevin Chen-Chuan Chang. Understanding web query interfaces: best-effort parsing with hidden syntax. In *Proc. of the ACM COMAD'04*, pages 107–118, 2004.