

# With or without context

Automatic text categorization using semantic kernels

Johan Eklund

VALFRID 2016

Johan Eklund (2016). *With or without context: Automatic text categorization using semantic kernels*

Dissertation at the Swedish School of Library and Information Science,  
the University of Borås

Cover: Jennifer Tydén, Daniel Birgersson, Mecka Reklambyrå AB

Print: Responstryck, Borås, 2016

Series: Skrifter från Valfrid, nr. 60

ISBN (printed version) 978-91-981654-8-7

ISBN (digital version) 978-91-981654-9-4

ISSN 1103-6990

Available at: <http://urn.kb.se/resolve?urn=urn:nbn:se:hb:diva-8949>

Typeset by the author using L<sup>A</sup>T<sub>E</sub>X

# Contents

<b>Preface</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Problem statement . . . . .	6
1.2 Research questions . . . . .	12
<b>I Toward a theory of subject classification</b>	<b>15</b>
<b>2 Metatheoretic perspectives</b>	<b>16</b>
2.1 Definitions . . . . .	16
2.2 Metatheoretic perspectives . . . . .	18
2.2.1 Semantics and semiotics . . . . .	19
2.2.2 Induction and underdetermination . . . . .	20
2.2.3 Text categorization and ceteris paribus . . . . .	23
2.2.4 Text categorization and instrumentalism . . . . .	24
2.2.5 Text categorization and positivism . . . . .	27
<b>3 Document classification</b>	<b>29</b>
3.1 Definitions . . . . .	30
3.1.1 Class and classification . . . . .	30
3.1.2 Classification scheme . . . . .	32

3.1.3	Document subject . . . . .	34
3.2	Relations in classification schedules . . . . .	35
3.2.1	Syntactical relations . . . . .	37
3.2.2	Hierarchical relationships . . . . .	39
3.3	Classification as language use . . . . .	40
<b>4</b>	<b>Subject classification</b>	<b>44</b>
4.1	Set theory . . . . .	46
4.1.1	Operations on sets . . . . .	47
4.2	Formal languages . . . . .	50
4.2.1	Document collections as formal languages . . . . .	51
4.2.2	Classification schedules as formal languages . . . . .	52
4.3	Category theory . . . . .	55
4.3.1	Definitions . . . . .	55
4.3.2	Document collections as categories . . . . .	58
4.3.3	Classification and category theory . . . . .	62
4.3.4	Subobject classifiers . . . . .	63
4.3.5	The space of classifiers on $D$ . . . . .	67
4.4	The algebraic structure of subject spaces . . . . .	69
4.4.1	Semantics and syntax: a model-theoretic perspective . . . . .	74
4.4.2	First-order languages and binary classifiers . . . . .	78
4.5	Order theory . . . . .	80
4.5.1	Basic terminology of order theory . . . . .	80
4.5.2	Hasse diagram . . . . .	82
4.5.3	Lattice . . . . .	83
4.5.4	Order theory and classification . . . . .	84
4.6	Graph theory . . . . .	86
4.6.1	Basic concepts of graph theory . . . . .	86
4.6.2	Classification schedules as graphs . . . . .	88

4.7	Topology . . . . .	91
4.7.1	Basic definitions in topology . . . . .	92
4.7.2	Basis and subbasis . . . . .	93
4.7.3	Neighborhood and homeomorphism . . . . .	94
4.7.4	Distinguishability and connectedness . . . . .	95
4.7.5	Subject classification and topology . . . . .	96
4.7.6	Dimension . . . . .	103
4.7.7	Dimensionality of classification . . . . .	107
4.8	Concluding remarks . . . . .	112

## **II Automatic text categorization in theory and practice 114**

<b>5</b>	<b>Automatic text categorization</b>	<b>115</b>
5.1	Overview . . . . .	115
5.2	Tokenization and normalization . . . . .	116
5.3	Feature selection and frequency laws . . . . .	118
5.3.1	Heaps' law . . . . .	119
5.4	Document representation . . . . .	124
5.4.1	Term weighting by tf-idf . . . . .	127
5.4.2	Term weighting by divergence from randomness	134
5.5	Supervised and unsupervised classification . . . . .	139
5.5.1	Unsupervised classification (Clustering) . . . . .	139
5.5.2	$k$ -means clustering . . . . .	139
5.5.3	Hierarchical clustering . . . . .	141
5.5.4	Supervised classification . . . . .	142
5.5.5	$k$ -nearest neighbor classification . . . . .	143
5.5.6	Naïve Bayesian inference . . . . .	144
5.5.7	Perceptrons and feedforward neural networks .	145
5.6	Elements of statistical learning theory . . . . .	149
5.6.1	Empirical risk minimization . . . . .	149

5.6.2	Vapnik-Chervonenkis dimension . . . . .	152
5.6.3	Structural risk minimization and regularization	155
<b>6</b>	<b>Support vector machines</b>	<b>158</b>
6.1	Introduction . . . . .	158
6.2	Comparative performance . . . . .	158
6.3	Quadratic programming . . . . .	160
6.3.1	Primal and dual form . . . . .	162
6.3.2	Lagrange multipliers . . . . .	163
6.3.3	Karush-Kuhn-Tucker conditions . . . . .	164
6.4	Linear SVM using a hard margin . . . . .	166
6.4.1	The SVM optimization problem in the primal form . . . . .	170
6.4.2	The primal form and the KKT conditions . . .	171
6.4.3	The optimization problem in the dual form . .	173
6.5	Soft-margin SVM . . . . .	175
6.5.1	C-SVM . . . . .	175
6.5.2	$\nu$ -SVM . . . . .	179
6.6	Kernel methods for SVM . . . . .	181
6.6.1	Kernels . . . . .	182
6.6.2	The Riesz representation theorem . . . . .	183
6.6.3	Reproducing kernel Hilbert space . . . . .	184
6.6.4	The kernel trick . . . . .	186
6.6.5	Mercer's theorem . . . . .	189

<b>III</b>	<b>Experiments with semantic kernels</b>	<b>192</b>
<b>7</b>	<b>Semantic kernels</b>	<b>193</b>
7.1	Document vectors and tensor calculus . . . . .	195
7.1.1	Formal definition of a semantic kernel . . . . .	199
7.1.2	The metric tensor of Mercer kernels . . . . .	200
7.2	Distributional semantics . . . . .	203
7.3	Methods for measuring semantic similarity . . . . .	204
7.3.1	Latent semantic analysis . . . . .	205
7.3.2	Random indexing . . . . .	207
7.3.3	Pointwise mutual information . . . . .	209
<b>8</b>	<b>Experimental setup</b>	<b>212</b>
8.1	General procedure . . . . .	213
8.2	Selection of reference collections . . . . .	214
8.2.1	Reuters-21578 . . . . .	214
8.2.2	OHSUMED . . . . .	215
8.2.3	20 Newsgroups . . . . .	217
8.3	Generation of document representations . . . . .	218
8.4	Term weighting . . . . .	219
8.4.1	Tf-idf . . . . .	219
8.4.2	Divergence from randomness . . . . .	220
8.5	Generation of semantic kernels . . . . .	220
8.5.1	Pointwise mutual information (PMI) . . . . .	221
8.5.2	Latent semantic analysis (LSA) . . . . .	222
8.5.3	Random indexing (RI) . . . . .	222
8.6	Training and testing of SVM classifiers . . . . .	224
8.6.1	Variables . . . . .	224
8.6.2	Configuration of SVM hyperparameters . . . . .	225
8.6.3	Sampling procedure . . . . .	226
8.6.4	Evaluation . . . . .	227

8.7	Software used in this work . . . . .	232
<b>9</b>	<b>Results</b>	<b>233</b>
9.1	Results for the Reuters-21578 collection . . . . .	237
9.1.1	Using the <i>tf-idf</i> weighting scheme . . . . .	239
9.1.2	Using the <i>dfr</i> weighting scheme . . . . .	244
9.1.3	Comparison between the semantic kernels . . .	248
9.2	Results for the Ohsumed collection . . . . .	250
9.2.1	Using the <i>tf-idf</i> weighting scheme . . . . .	251
9.2.2	Using the <i>dfr</i> weighting scheme . . . . .	256
9.2.3	Comparison between the semantic kernels . . .	260
9.3	Results for the 20 Newsgroups collection . . . . .	262
9.3.1	Using the <i>tf-idf</i> weighting scheme . . . . .	264
9.3.2	Using the <i>dfr</i> weighting scheme . . . . .	269
9.3.3	Comparison between the semantic kernels . . .	274
<b>10</b>	<b>Conclusions</b>	<b>276</b>
	<b>Bibliography</b>	<b>280</b>
	<b>Publikationer i serien Skrifter från VALFRID</b>	<b>300</b>

## Abstract

In this thesis text categorization is investigated in four dimensions of analysis: theoretically as well as empirically, and as a manual as well as a machine-based process. In the first four chapters we look at the theoretical foundation of subject classification of text documents, with a certain focus on classification as a procedure for organizing documents in libraries. A working hypothesis used in the theoretical analysis is that classification of documents is a process that involves translations between statements in different languages, both natural and artificial. We further investigate the relationships between structures in classification languages and the order relations and topological structures that arise from classification. In the following chapter we give an overview of machine-based (or algorithmic) classification as a process typically involving machine learning. In this section of the thesis the components of the machine classification process are described, including the generation of document representations (typically being document vectors), as well as the training and classification phase. We also present an assortment of important classification and clustering algorithms.

A classification algorithm that gets a special focus in the subsequent chapters is the *support vector machine* (SVM), which in its original formulation is a binary classifier in linear vector spaces, but has been extended to handle classification problems for which the object categories are not linearly separable. To this end the algorithm utilizes a category of functions called *kernels*, which induce feature spaces by means of high-dimensional and often non-linear maps. For the empirical part of this study we investigate the classification performance

of semantic kernels generated by different measures of semantic similarity. One category of such measures is based on the *latent semantic analysis* and the *random indexing* methods, which generate term sense vectors by using co-occurrence data from text collections. Another semantic measure used in this study is *pointwise mutual information*. In addition to the empirical study of semantic kernels we also investigate the performance of a term weighting scheme called *divergence from randomness*, that has hitherto received little attention within the area of automatic text categorization.

The result of the empirical part of this study shows that the semantic kernels generally outperform the “standard” (non-semantic) linear kernel, especially for small training sets. A conclusion that can be drawn with respect to the investigated datasets is therefore that statistical semantic information in the kernel in general improves its classification performance, and that the difference between the standard kernel and the semantic kernels is particularly large for small training sets. One possible interpretation of this result is that the use of semantic kernels can to a certain extent compensate for a lack of training data. Another clear trend in the result is that the *divergence from randomness* weighting scheme yields a classification performance surpassing that of the commonly used *tf-idf* weighting scheme.

## Sammanfattning

I denna avhandling undersöks textkategorisering i fyra analysdimensioner: teoretiskt såväl som empiriskt, och som en manuell respektive en maskinell process. I de första fyra kapitlen analyserar vi den teoretiska grunden för ämnesklassifikation av textdokument, med ett särskilt fokus på klassifikation som en procedur för organisation av dokument i bibliotek. En arbetshypotes som används i den teoretiska analysen är att klassifikation av dokument är en process som involverar översättningar mellan utsagor i olika språk, såväl naturliga som artificiella. Vi undersöker vidare relationerna mellan strukturer i klassifikationspråk och de ordningsrelationer och topologiska strukturer som uppstår vid klassificering. I det följande kapitlet ger vi en översikt över maskinell (eller algoritmisk) klassifikation som en process som i allmänhet involverar maskininlärning. I detta avsnitt av avhandlingen beskrivs de olika komponenterna i den maskinella klassifikationsprocessen, inklusive generering av dokumentrepresentationer (vanligen dokumentvektorer) samt tränings- och klassifikationsfasen. Vi presenterar också ett urval av viktiga metoder för klassifikation och klusteranalys.

En klassifikationsalgoritm som får ett särskilt fokus i de följande kapitlen är *supportvektormaskinen* (SVM), vilken i sin ursprungliga formulering är en binär klassificerare i linjära vektorrum, men som har anpassats för att hantera klassifikationsproblem för vilka objektkategorierna inte är linjärt separerbara. För detta syfte använder algoritmen en kategori av funktioner som kallas *kärnor* (eng. *kernels*), som inducerar egenskapsrum genom högdimensionella och ofta icke-linjära mappningar. För den empiriska delen av studien undersöker vi klassifikationsprestandan hos semantiska kärnor genererade av olika

mått på semantisk likhet. En kategori av sådana mått är baserad på *latent semantisk analys* samt *slumpindexering*, vilka genererar term-betydelsevektorer genom att använda samförekomstdata från textkollektioner. Ett annat semantiskt mått som används i denna studie är *punktvis ömsesidig information* (eng. *pointwise mutual information*). Förutom den empiriska studien av semantiska kärnor undersöker vi även prestandan hos ett termviktningsschema som kallas *avvikelse från slumpmässighet* (eng. *divergence from randomness*), som hittills har fått ringa uppmärksamhet inom automatisk textkategorisering.

Resultatet av den empiriska delen av denna studie visar att de semantiska kärnorna i allmänhet presterar bättre än den “vanliga” (icke-semantiska) linjära kärnan, särskilt för små träningsmängder. En slutsats som kan dras med avseende på de undersökta datamängderna är därför att statistisk semantisk information i kärnan i allmänhet förbättrar klassifikationsprestandan, och att skillnaden mellan standardkärnor och de semantiska kärnorna är särskilt stor för små träningsmängder. En möjlig tolkning av detta resultat är att användningen av semantiska kärnor i viss mån kan kompensera för en brist på träningsdata. En annan tydlig trend i resultatet är att termviktningsschemat *avvikelse från slumpmässighet* ger en klassifikationsprestanda som överträffar det ofta använda viktningsschemat *tf-idf*.

# Preface

This work is partly the outcome of my determination to combine two of my great interests – mathematics and computing. It has been an unadulterated joy to get the opportunity to use mathematics as a tool and language to express and analyze various ideas throughout the work with this thesis. Another interest that has grown to become a fascination during my PhD studies is the one for language. It stands clear that language is an indispensable vehicle of human thought on many different levels: communication, web development, music, visual art, mathematics etc. Much has been said and written about the profusion of information in current society, but it also needs to be stressed that it is difficult to imagine information detached from language. It is hardly a coincidence that classification, another basic cognitive activity, stands in a close relationship to language and language use.

I want to extend my thanks to my supervisor, professor Sándor Darányi, and others who have contributed with ideas, inspiration and guidance through the process of producing this text. A special mention goes to professor Jan Nolin for many helpful suggestions during the concluding part of this project.

Last, but not least, I want to express my gratefulness to my family for being a continual support.

# Chapter 1

## Introduction

One of the prominent tasks of the library is to efficiently provide access to written knowledge. Because of the extensive production of printed literature, and in later years digital documents, it was soon realized that the information contained in the library could not just be stored randomly or according to some simple principle like alphabetic order or accession order. The library needs to be structured according to *subject content*, i.e. what the documents are about. Not only does such a structuring provide easier access to a particular document with special relevance for a certain information need, but it will also facilitate *discovery* in the sense that the library user may find other documents of interest in the proximity of the target document.

For this reason the praxis of *knowledge organization* emerged, the objective of which is to place documents (typically under the influence or direct action of an information professional such as a librarian) in such a way so as to optimize their chance of being retrieved. In addition, records are being kept about the documents as *surrogates* in a catalog. This process, called *cataloguing*, typically involves a formal description of the documents' bibliographic properties and also

involves an assignment, called *indexing*, of relevant subject terms to the documents. Another important activity with the same objective to induce structure in the document repository of the library is *subject classification*, which refers to a procedure that entails an analysis of the documents with respect to their subject content, an identification of appropriate codes from a classification vocabulary, and the assignment of the selected codes to the documents.

The dramatic growth in document production over the last couple of decades, and the advancing availability of digitally stored and transmitted information, has also increased the need for computer-based tools that can aid in filtering and extracting relevant items from the information storage, as well as adding a rational structure to the bulk of information (Stavrianou et al., 2007; Nisa & Qamar, 2014). The research field *automatic document classification* is an area that has emerged in the intersection between traditional knowledge organization and modern computer science research on pattern recognition. We can characterize automatic document categorization from two perspectives: as a process and as a research area. The overall objective of the notion from a process perspective is to assign documents to one or several categories by machine-based (or more precisely: algorithmic) means. Even if it is theoretically possible that such an assignment of categories could be performed by a fixed set of machine-implementable rules, this task is normally performed by the aid of *machine learning* (Baeza-Yates & Ribeiro-Neto, 2011, p. 282).

In this work the terms *document categorization* and *document classification* are used interchangeably for stylistic variation, and are therefore considered synonymous. Jacob (2004) argues that there is a fundamental difference between these two terms, and that a conflation of these terms should be avoided. Categorization is, according to Jacob (2004), defined as “the process of dividing the world into

groups of entities whose members are in some way similar to each other”, whereas classification “involves the orderly and systematic assignment of each entity to one and only one class within a system of mutually exclusive and nonoverlapping classes”. The stipulative definition of *classification* that Jacob provides is, however, questionable. It entails a redundancy, since two classes are mutually exclusive if and only if they are nonoverlapping. Also, the restriction imposed on classification as a process involving the assignment of an entity to precisely one class is not made in Spärck Jones (1970), where the author proposes the existence of overlapping classes. As discussed in chapter 3, it also is the case that documents are typically classified according to content-related properties such as topic or genre, from which it follows that documents assigned to the same class also are to some extent similar to each other. It could be argued that categorization entails a top-down process that involves the division of a universe of entities into a collection of groups, whereas classification involves a bottom-up process by assigning single documents to groups according to some kind of criterion. The end result will nonetheless be a grouping of documents according to some kind of similarity condition. Consequently, we also find the terms *text classification* (e.g. Baeza-Yates & Ribeiro-Neto, 2011) and *text categorization* (e.g. Sebastiani, 2005) used interchangeably in the research literature.

Although classification has traditionally been an activity carried out by information specialists, the increasing production of digital documents and the advent of new information infrastructures such as the World Wide Web has also raised the interest for automated knowledge organization services (see e.g. Yi, 2006). The use of machine-based classification does not entail a paradigmatically new approach to classification, although there obviously are conspicuous differences on a procedural level. Since the early 1990s there have been a few em-

pirical studies conducted on the potential of traditional classification schemes, such as the DDC and the LCC schemes, for automatic classification of digital resources. We will briefly present a few of those in order to exemplify the methodology used in the research on applying traditional knowledge structures in an automatized setting.

Larson (1992) studied the extent to which the LCC codes could be automatically assigned by classification systems trained on information contained in titles and subject headings in document records. The general procedure was to generate representation vectors (see section 5.4) from the document metadata and out of these construct vectors representative for each class  $c$  by accumulating the information contained in the vectors pertaining to the documents in  $c$ . One obvious possibility, mentioned in the article, is to form the *centroid* of all such document vectors. The document-class similarity measure used and compared were the *dot product* and a *probabilistic* measure. The best result, a classification accuracy of 46.6%, was obtained using the first subject heading stemmed with respect to plural forms, using a probabilistic decision function.

Thompson et al. (1997) evaluated the potential usefulness of DDC codes for automatic classification by studying the clusters of classes formed around a sample of classification codes. One of the prominent objectives of that study, conducted in the frame of the *Scorpion* project (see e.g. Shafer, 2001) at OCLC, was to investigate the *class integrity* in the DDC database, i.e. the extent to which classes are separable by the metadata assigned to the classes. The methodological approach was to perform a classification of the concept definitions pertaining to the classes. A class is in this study said to have high integrity if it is not mixed up with any other class during this process. To this end information contained in the Editorial Support System (ESS), used to maintain the DDC database, was utilized to generate and cluster tf-idf

weighted class vectors. The similarity measures used were dot product and the *cosine* measure (of which the latter has been commonly used as a similarity measure in *information retrieval*, cf. the presentation in section 5.4). A general result from that study was that a high level of class integrity was obtained, although *self-matches* (i.e. the target class ranked as number one in the ranked list of similar classes) occurred only rarely.

Frank & Paynter (2004) performed a study similar in scope as (Larson, 1992) but with an approach that utilizes the hierarchical structure of the LCC scheme. The general methodology was to train a system of SVM classifiers on Library of Congress Subject Headings (LCSH) assigned to the document records. A *round robin* procedure (see Fürnkranz, 2002) was applied, meaning a binary SVM classifier was trained for *each pair* of classes in the target structure. For each pair of classes and a document  $d$  the corresponding SVM classifier produces a “vote” on the predicted class, whereby the class obtaining the highest number of votes “wins” and is assigned as the predicted class for  $d$ . An extensive number (about 800,000) of training instances were used to train this configuration of classifiers. In the evaluation of the trained system it was noted that in 80.27% of the cases the correct top level class was found, whereas in only 16.12% of the cases the correct level-7 class was identified.

## 1.1 Problem statement

What is the precise meaning of concepts like *class* and *classification*? These notions may be taken for granted or be used in a practical, operational sense, but to provide adequate definitions is not straightforward. A working hypothesis that permeates this thesis is that document classification is essentially an activity that involves translations

between different *languages*, both in the input to as well as the output from the classification process. In order to study document classification empirically it is important that we proceed from a solid theoretical understanding of what document classification actually means, and therefore we will devote a considerably large part of this work to theoretically investigating this concept from different perspectives. Several authors writing from the perspective of library and information science have argued for the need of a formal theory of document classification, and proposed an outline of what such a theory may contain. Spärck Jones (1970) claims that the emergence of automatic document classification has raised new questions concerning the principles on which document classification is based, and how a classification theory may be used for a particular information retrieval purpose. Picking up on Spärck Jones' request for a general theory of classification, Hjørland & Pedersen (2005) write: "Although many different approaches have been tried, this may still be the case in 2005." In the same article Hjørland & Pedersen claim that any theory of classification has to take into consideration that classification of documents always involves a specific *purpose*, and that the notion of a purpose may be difficult to capture in a formal theory. Mokhtar & Yusof (2015) call classification an "understudied" concept and state that the lack of understanding of this notion may hazard the management of digital information.

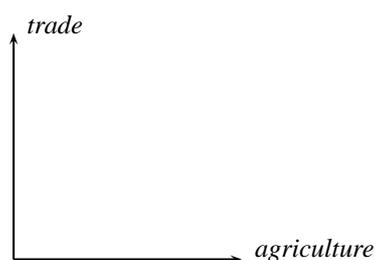
A restriction that is commonly made in the organization of resources in libraries is the requirement that the descriptors used for classification and indexing should be selected from strictly defined lists of words, so called *controlled vocabularies*. The linguistic observation underlying the use of controlled vocabularies (rather than *free* vocabularies) is the semantic variation inherent in natural languages. For instance, it is often the case that several terms can be used for the

same concept (*synonymy*), or conversely that the same term may in different contexts denote different concepts (*homonymy* or *polysemy*). Terms may belong to the semantic scope, but may have different levels of specificity (*hyponymy*, *hypernymy*), or there may be an association between the terms that cannot easily be described in terms of a specific semantic relation. In the terminology used in thesauri we typically find relations like *broader term*, *narrower term*, *related term*, and *use for*. Likewise, in classification systems we often find that the classification codes have been arranged in a hierarchical fashion. It could be argued that the organization of resources in libraries is not only obtained by grouping these resources according to the descriptors they have in common, but also that the semantic relations that are assumed to hold between the descriptors provide an overall context that facilitates the localization of resources relevant to a specific information need.

Many methods for text categorization by machine-based means exist, some of which are briefly reviewed in chapter 5. In this thesis we are studying a particular method for automatic document categorization, called *support vector machines* (SVMs), presented in detail in chapter 6. This classification method has a sound theoretical basis in statistical learning, can be adapted to handle nonlinear classification problems, and has shown good comparative performance against other classifiers. SVMs belong to the category of *supervised* machine learning algorithms, meaning that they need to be trained on pre-categorized data before they can perform classification with reasonable accuracy.

In machine classification the vector space model (see chapter 5.4) has for several decades been a popular representation scheme for text documents, due to its simplicity, general performance, and sound theoretical basis. However, in its original formulation it represents docu-

ments as vectors of *term weights* – each term being assigned a unique dimension in an orthonormal Euclidean space (see figure 1.1). One conspicuous property of this feature space is that it does not contain any information about *relations* between the terms used to represent the documents. One could say that the original formulation of the vector space model is semantically “ignorant”.



**Figure 1.1.** The term vectors are pairwise orthogonal in the original vector space model.

An emerging research area in computational linguistics is that of statistical semantics, i.e. computational models of semantic relatedness between various units in language, such as words and phrases (Farahat & Kamel, 2011). The underlying idea of such methods is the assumption that the semantic relatedness (or similarity) between words in a particular language can be quantified on the basis of their co-occurrence within specific contexts. This proposition is also known as the *distributional hypothesis*, which states that words with similar meaning tend to be distributed in a similar way in the texts where they occur (Sahlgren, 2008). This principle can be succinctly summarized in the expression attributed to the linguist John Rupert Firth (see e.g. K. W. Church & Hanks, 1990):

You shall know a word by the company it keeps.

This could be said to be an expression of a *contextualist* approach to

semantics, i.e. that word meaning should be established by investigating the context in which the words appear. An assortment of statistical methods and models for quantifying semantic relatedness between linguistic units have been proposed and extensively studied, of which a few have been selected for the empirical study of this thesis. The distributional hypothesis and statistical methods for capturing word senses are presented in chapter 7.

The empirical research focus in this work is to study how the incorporation of information acquired from methods for statistical semantics affects the performance of machine classifiers based on the SVM algorithm. As stated above, the idea to use semantic information to improve access to documents is by itself not a novel theme in library and information science. On the contrary, it is a well-established praxis in knowledge organization to use controlled vocabularies such as thesauri to provide multifaceted entry points to the library resources. However, contrary to the binary relations present in such vocabularies we find that a common denominator between the mentioned methods for statistical semantics is that they do not specify the *types* of relationships that exist between words, but rather their *degree* of relatedness. This approach is comparable to Eleanor Rosch's *prototype theory* (Rosch, 1975), which stipulates that words in language are not equally related to various concepts (in the binary sense of either-or), but that words can be ranked according to their degree of "relatedness" to a particular concept.

In this work we use the information acquired from methods for statistical semantics to implement a selection of semantic kernels. A *kernel* is in this context a mathematical structure (comparable to a symmetric table) that stipulates how vectors in a space should be (informally speaking) compared. More specifically, the kernel specifies how the *inner product* between vectors is computed. The kernel is in

turn closely related to another mathematical concept that has important applications in theoretical physics, namely that of *metric tensors*. A metric tensor can also be perceived as a tabular structure that defines how measures like the geodesic distance along a path between two points on a curved surface should be computed, which is a generalization of the notion of linear distance on a flat surface.

By incorporating semantic information in the kernel we also change the properties of the document representation space according to the degree of relatedness between the terms defining the document space. Our hypothesis is that the use of semantic kernels will yield a document space that improves the separability of the document categories, a *semantic* vector space in which the orthogonality assumption between the terms no longer holds. We thereby seek to expand on existing studies of automatic text categorization with semantic kernels by comprehensively and comparatively studying the performance of different semantic kernels, using different methods for extracting semantic information from text corpora. In particular, we aim to compare statistical semantic methods utilizing term co-occurrence in larger textual units such as documents, and methods that utilize information from the immediate context of the terms as they appear in the running text.

Another problem that is empirically studied in this work is that of *term weighting*, i.e. how the relationship between documents and their constituent words should be computationally specified. Traditionally, the vector space model has been implemented using a combination of frequency-based measures, most prominently the *tf-idf* weighting scheme. This weighting scheme is based on the assumption that the local (within-document) term frequencies positively correlate to their usefulness as document descriptors, whereas the global (collection-based) term frequencies have a negative correlation to their specificity

(the more frequently the term occurs in the collection, the less significant it is as a document descriptor). In this work we also study the comparative performance of a probabilistic language model for term weighting called *divergence from randomness* (Amati & Van Rijsbergen, 2002). This model is based on the probabilistic assumption that the significance of term frequencies should be put in relation to their degree of divergence from the term distribution of a document collection that (hypothetically) has been generated by a random process. More specifically, we look at a variant of the divergence from a randomness scheme that is based on Bose-Einstein statistics – a model that has, as the name of the statistical model suggests, connotations of theoretical physics. Although the divergence from randomness model has had certain applications within information retrieval, it appears to have received little (if any) attention in the area of machine classification.

## 1.2 Research questions

One major objective of this thesis is to establish a *theoretical framework* that highlights the connections between traditional (manual) subject classification as practised in libraries, machine classification in general, and the SVM algorithm. Another research purpose of this work is to compare different methods for obtaining semantic information from full-text collections, and thereby generate semantic kernels for machine classification of text documents using the SVM algorithm. The methods for statistical semantics selected for this work are pointwise mutual information, latent semantic analysis, and random indexing. These methods differ with respect to how term co-occurrence is measured and quantified. The normalized pointwise mutual information collects information about the amount of informa-

tion that terms provide about each other. The latent semantic analysis method provides information about the extent to which terms co-occur on a document level, whereas the random indexing method utilizes the *local context* of each term to generate context vectors providing information about the distributional patterns of terms.

We also aim to study the comparative classification performance of two term weighting schemes with different theoretical underpinnings: the term frequency/inverse document frequency (tf-idf) scheme, and the divergence from randomness weighting scheme. The classification performance is investigated in three different reference collections (see section 8.2). More specifically, the following general research questions are investigated in this thesis.

With respect to the theoretical understanding of classification:

1. How can subject classification be defined and characterized using a formal theoretic framework?
2. How can the structures of hierarchical classification schedules as well as document structures generated by classification be formally described?

With respect to the empirical study of weighting schemes and semantic kernels:

3. What is the comparative classification performance between the tf-idf and the divergence from randomness weighting schemes, for different sizes of the data used for training?
4. What is the comparative classification performance of the different semantic kernels, and how do they compare to a baseline linear kernel without semantic information?

5. Are the comparative differences similar over different types of document collections?

Research questions 1–2 are primarily investigated in chapter 3 (*Document classification*) and chapter 4 (*Subject classification*). The theoretical foundation underlying research questions 3–5 is presented in chapters 5 (*Automatic text categorization*), 6 (*Support vector machines*), and 7 (*Semantic kernels*). The methodology used for empirically investigating research questions 3–5 is presented in chapter 8 (*Experimental setup*) and the results of the empirical study is presented in chapter 9 (*Results*). Both the theoretical and the empirical findings are summarized and discussed in chapter 10 (*Conclusions*).

## **Part I**

# **Toward a theory of subject classification**

## Chapter 2

# Metatheoretic perspectives

This chapter provides a description of text categorization from a metatheoretic perspective. Initially, basic concepts and approaches are presented, followed by an analysis of research from a philosophy of science perspective.

### 2.1 Definitions

*Text categorization* is the process of assigning text documents to one or more groups called *classes* or *categories*. If this process is carried out using computer software, without manual intervention, we refer to this process as *automatic* text categorization. If the documents are assigned to groups *without* class labels, this procedure is usually called *text clustering* (Baeza-Yates & Ribeiro-Neto, 2011, p. 282).

We can formally characterize text categorization as follows (Sebastiani, 2005). Let  $D$  be a set of documents and  $C$  a set of categories. Further, let the symbol  $T$  represent the statement "is assigned to the category" and the symbol  $F$  the statement "is not assigned to

the category”. Text categorization can then be written as a function

$$\varphi : D \times C \rightarrow \{T, F\} \quad (2.1)$$

Put another way, the function assigns to each pair of documents and categories a value that specifies whether the document is included in the category or not. This function is called a *target function* since it specifies the desired output for certain classification decisions. The goal of automatic text categorization is to induce a function

$$\psi : D \times C \rightarrow \{T, F\} \quad (2.2)$$

such that  $\psi$  approximates  $\varphi$  as much as possible. We call the induced function  $\psi$  a *classifier* (Baeza-Yates & Ribeiro-Neto, 2011, p. 283).

For the evaluation of an automatic categorization of a set of documents, the automatic categorization is typically compared to a manually constructed categorization, whereby different evaluation measures are calculated (see section 8.6.4). This is regarded as a necessary procedure to determine the performance of a certain classifier (Baeza-Yates & Ribeiro-Neto, 2011, p. 325). The phenomenon that these measures are considered to quantify, the degree of correspondence between the automatic categorization and the manual categorization, is called *classification performance* (Joachims, 2002, p. 27).

The objective of subject classification, which is the kind of classification that is typically associated with text categorization, is to determine what documents “are about” and on the basis of this analysis assign the documents to categories that best correspond to the identified content of the documents. This kind of classification is called *intensional* (see e.g. Marradi, 1990) since it is based on specific properties (intensions) of the content, which are matched against (implicitly stated) member conditions of the category scheme at hand. An

important difference between the analysis made by the human classifier and the machine-based analysis is that the human analysis is typically much richer and complex, involving a deeper understanding of the language the document is written in, as well as contextual factors involved in the creation of the document. Beside investigating formal properties like authorship, title, publisher and so on, which may provide an initial clue about the category of the document, the human classifier also performs a deeper linguistic analysis of the text, involving the syntactic and narrative structure of the text, anaphora and pragmatic aspects on the discursive context of the text.

It can be argued that the machine-based analysis is typically more superficial, treating the text merely as a *multiset* (bag) of words, while discarding word order (see chapter 5). This is known as the *bag-of-words* representation of textual content (Baeza-Yates & Ribeiro-Neto, 2011, p. 62). The frequency distribution of particular words in the text, separated from their location within the structure of text, is then used as a basis for the representation of the content of the document for automatic classification. This bag-of-words approach can be compared to the assignment of keywords or descriptors to documents in library catalog. This means that the machine-based content analysis is strongly focused on linguistic *tokens* such as words and word sequences (phrases), while often discarding the semantic properties of the text.

## 2.2 Metatheoretic perspectives

This section identifies the key problems and methodological components involved in text categorization.

### 2.2.1 Semantics and semiotics

Semantics and semiotics are a branch of linguistics dealing with the study of the *meaning* of linguistic units. More specifically, *lexical semantics* deals with the meaning of units in language called *words* (Cruse, 2004, p. 13). How “meaning” should be understood is, however, not unproblematic and several non-equivalent interpretations have been provided by various scholars in different fields. This can be exemplified with the contrasting views of the philosopher Charles Peirce and the linguist Ferdinand de Saussure (Kj rup, 1999, p. 236). According to Peirce the word is a *sign* that refers to a set of entities external to the sign, and this meaning is given to the word by an *interpretant*. From Peirce’s viewpoint the question “what does the word  $x$  mean?” can be translated “what does  $x$  refer to, in the interpretation given by  $y$ ?”. De Saussure, on the other hand, states that the meaning of a word is part of the *essence* of the word, and any external reference is of no essential relevance to the linguist. The word consists of, in this interpretation, both a symbol (the *signifier*) and a content (the *signified*). What is, then, the practical difference between these notions and which consequences may they entail for the content representation of a document? What appears to be the case is that both these views involve the idea of an association between linguistic units and corresponding cognitive notions, which commonly are called *concepts*.

From an operational perspective the view of de Saussure appears to be adequate for *closed* systems since it makes semantics into a system where the meaning of the word is defined by its relation-in-use to other words. Peirce’s idea of reference to a category outside the word seems deemphasized in de Saussure’s theory. To illustrate the problem of reference we can take as an example the notion of *unicorns*. We

may regard *unicorn*, as any other symbol in the English language, as a linguistic expression with an associated intensional category defined by other words in the same language. Regardless of whether there exists a *referent* to this symbol, in the realist sense of something observable, it is still possible to outline a category for this symbol and we can identify documents that *are about* this category. To describe and represent the content of documents it is therefore not necessary, or always even possible, to identify the referents of the contained linguistic units. From an operational perspective the essential property of each linguistic unit is the set of *relations* it has to other linguistic units. In a system for automatic document categorization it is therefore desirable that semantic relations between words and phrases are stored in a processable representation. For instance, the existence of equivalence (synonym) relations or hierarchical (hypernym / hyponym) relations between words contained in documents reveal conceptual relations between these documents that would go unnoticed using a string-level processing of the text. Approaches within *distributional semantics* are aimed at statistically detecting semantic relations between words by investigating the co-occurrence of terms in a set of contexts (see section 7.2).

### 2.2.2 Induction and underdetermination

Research on automatic text categorization involves a twofold focus on content representation and the use of various classification algorithms. Each set of variables entails essentially different research questions. The researcher's focus may be targeted on the properties that are most useful to describe the content of the documents as well as separate the documents from each other. This may for instance involve an analysis of the semantic properties of the documents, but syntactical

aspects may also be of interest. If the research focus is on the classification method the approach may be comparatively stated, involving a juxtaposition of different mathematical formulations of the classifier in order to find significant differences in terms of classification performance. A subproblem may be to find an optimal configuration of parameters for the actual method. With regard to SVM, which is the algorithm used in the empirical part of this work, the choice of parameters for class separability and imperviousness to outliers and mislabeled data is a crucial factor for classification performance.

An interesting observation with regard to automatic categorization is that the induced classifier in fact is in itself a *theory* about the relation between document content (according to the used representation form) and document category (according to human-produced examples). Now, Rosenberg (2005, p. 117) states that any scientific theory that is formulated on a positive form and with applicability to all objects in a certain domain also entails a proposition on a *negative* form. We can summarize this observation using the following expression in predicate logic. Let  $C$  be a predicate denoting the property "is of kind  $C$ " and  $A$  a predicate expressing the property "has the quality  $A$ ". Then it holds that

$$(\forall x : C(x) \Rightarrow A(x)) \iff (\forall x : \neg A(x) \Rightarrow \neg C(x))$$

Expressed in words: if we can positively state that all objects being of kind  $C$  has the quality  $A$  then it follows that if an object  $x$  does *not* have quality  $A$  then  $x$  cannot be of kind  $C$ . This rule of deduction is known as *modus tollens* (see e.g. A. Church, 1996, p. 104). The proposition "all swans are white" entails a corresponding, *dual*, proposition: "if a thing is not white, then it is not a swan". Since the proposition is expressed on a hypothetical form (if – then) it can not

be applied to deduce that there *exist* white swans, but *if* swans exist they are white. To infer the existence of white swans it is therefore required that we a priori know that swans exist.

Karl Popper argued that scientific theories should be evaluated by means of *falsification* rather than verification (Popper, 1992, p. 18). The basis behind reasoning is that a single counterexample is sufficient to invalidate a universal proposition such as “all swans are white” (Howell, 2013, p. 44). On the basis of this observation the researcher should, while evaluating a research hypothesis, actively search for instances *contradicting* the hypothesis rather than single-mindedly collect cases that support it. If we now turn back to the situation of automatic classification, and more specifically the machine learning process involved in supervised classification, we find that the inductive mechanism of the system’s training component is in fact behaving like a researcher following this “recommendation” of searching for positive and negative indications of the current hypothesis. The steps involved can be outlined as follows:

1. Produce a *current* hypothesis  $h$  from a space  $\mathcal{H}$  of hypotheses.
2. Apply  $h$  to the training set of documents.
3. Evaluate  $h$  by a quantitative measurement of the capacity of  $h$  to identify positive instances as positive, as well as negative instances as negative.
4. If the stipulated number of iterations has been reached, finish the training. Otherwise, produce a new hypothesis  $h^+$  from  $\mathcal{H}$  and let it be the current hypothesis  $h$ . Proceed to step 2.

The hypothesis producing the highest classification performance by the end of training session is selected as the eventual classifier for the problem at hand. The analogy with the reasoning of Rosenberg is that

the classifier, i.e. the theory induced by training, is shaped by information about correctly as well as incorrectly classified documents. However, the established classifier is not regarded as a *universal* theory since it is normally accepted that it will misclassify certain instances even after extensive training.

### 2.2.3 Text categorization and *ceteris paribus*

In analogy with the scientific endeavor to explain a phenomenon in terms of an isolated set of causes, with all the other circumstances considered constant, the classifier is typically a function of a reduced number of parameters. This *ceteris paribus* assumption (“all other things being equal”, (Rosenberg, 2005, p. 49) is a deliberate simplification of the content – category relation in order to make the classifier computationally feasible. The classifier is normally not based on an endeavor to capture *all* the factors that may affect the category membership of a document. The mathematical formulation of the qualifier is hoped to capture a *sufficient* number of parameters to perform well on the classification problem at hand. It is a reasonable assumption that the cognitive basis for human-produced classification stretches beyond the simple vocabulary of the document. Still, the parameters used by the algorithmic classifier is normally derived from a narrow family of properties, such as the frequency distribution of words – *ceteris paribus*.

A concrete example of a deliberately simplified assumption is found in the *naïve Bayesian inference* method (see section 5.5.6). Given a document  $d$ , a class  $c$  and a set of features  $\mathcal{F}$ , the probability  $P(d|c)$  is translated into the product  $\prod_{f_i \in \mathcal{F}} P(f_i|c)$  on the assumption that these probabilities are *independent*. The presence of terms is a type of feature for which this assumption certainly is not true, but the

theory is still expressed in terms of presence / absence with all other factors (including the probabilistic dependencies between terms) held constant.

#### 2.2.4 Text categorization and instrumentalism

Like economics, the research area of automatic text categorization applies mathematical modelling to capture human behavior and thinking. It is not a natural science since its aim is not to survey and explain phenomena in nature. The theories derived are typically not rules but parameter configurations. Therefore, the relationship between features and category membership is not provided as a deductive-nomologic explanation (see Rosenberg, 2005, p. 30) since the *explanans* is usually *implicit* in the induced classifier. Every classifier is based on a “meta-theory” with the formulation

S1. There is a statistical relation between the feature configuration of a document  $d$  and the category membership of  $d$ .

This relationship is, however, not explicitly formulated in a set of statements with the kind of universal validity as various scientific laws are considered to possess. We are not provided with causal explanation as to why a document have been manually assigned to a category. The system rather gives us the following information:

S2. With discrimination function  $\phi$  and the parameter set  $\Theta$ , we achieve in  $n$  % of cases the same categorization as the manual.

We can not after the criteria stated by Hempel construct an argument on logical-deductive form where *explanans* contains a generally valid law. However, we can say that automatic document categorization is a

process having the objective to produce a *result* similar to the human categorization as far as possible, without necessarily reproducing the cognitive process of the human classifier. It is therefore not necessary to pursue the strict causally explanatory power of a theory that includes the cognitive processes leading to a specific categorization of the documents, as long as the artificial process (i.e. the machine categorization) yields the same result to a sufficient extent. This characterizes automatic document categorization as *probabilistically causal* (Rosenberg, 2005, p. 53) in the same sense as the observed correlation between living habits and certain diseases .

Formulated in terms of the conceptual pair *reasons – causes* (see e.g. Rosenberg, 1995, p. 33), we note that the true causal link between documents, the cognitive processes of the human classifier, and the eventual categorization is unfeasible to theorize. If we by  $\oplus$  denote the relationship ”interacts with”, manual classification can be formalized as:

$$\text{documents} \oplus \text{knowledge and preferences} \rightarrow \text{categorization}$$

Since the discrimination function is deterministic, we can describe the machine-based categorization in terms of causality:

$$\text{documents} \oplus \text{classifier} \rightarrow \text{categorization}$$

What we can observe is that both the parameter *documents* as well as the result *categorization* is common for both processes, which also juxtaposes *the knowledge and preferences of the human classifier* and the *discrimination function*. Furthermore, it should be noted that the human classifier, under the prevailing circumstances, can state *reasons* for his/her choice of document category, whereas the discrimination function *causes* the machine-based categorization. Since we do not

have a proper basis for modelling the reasons for the choice of category, even less the cognitive *causes* that are likely to occur, we decide to search for a model that is based on a feasible and essentially different set of parameters that helps to *approximate* the choices of the human classifier.

Based on the observations above, we have good support for claiming that the research area automatic document categorization is highly *instrumentalistic* (Rosenberg 1995, p. 83; Rosenberg 2005, p. 94) since the main objective characterizing the area is *not* to describe an objective reality with a set of (falsifiable) claims, but to find models that create a sufficiently high degree of predictability and order in the information universe. There is an implicit assumption of a *rational choice* (Rosenberg 1995, pp. 78, 84) made by the human classifier, entailing that the choice of category is dependent on the document and not on arbitrary decisions by the human classifier. In a specific categorization situation the human classifier is faced with the task of assigning a document  $d \in D$  to one of the categories  $c_i \in \mathcal{C}$ . It is reasonable to assume that the classifier is working according to principles having a mutual order of preferences, making the classifier to first select the category that best satisfies these preferences, then (if necessary) selects further categories by the same order of preference in the list. This principle is further assumed to be applied in a *consistent* manner, so that  $c_i$  is always chosen over  $c_j$  if the same circumstances conducive to  $c_i$  are present. The predictability that is assumed to follow the principle of rational choice is a theoretical justification for the application of a statistical classification model, rather than a model based on a mapping of the cognitive processes.

### 2.2.5 Text categorization and positivism

Research on automatic document categorization is adhering to the positivist tradition in the sense that there is an emphasis on empirical data collection, quantitative measurement and the testing of hypotheses. A model is rejected or maintained by measuring its ability to associate the documents with the categories to which the documents are manually assigned. In this process there is no assumption about the correctness of the category assignment in a strictly objective and unsituational sense. One problem with such a characterization, that deserves mentioning, is that there has been in the post-positivist tradition a strong emphasis on *falsification* as fundamental tool for (in)validating a scientific theory and *falsifiability* as a fundamental principle for determining which statements that may be considered meaningful (Howell, 2013, p. 44). As we have noted above, the theory we can formulate on basis of the conducted automatic document categorization does not have a *deductive-nomological form*. To begin with, the machine-induced classification model does usually not satisfy all the observed instances of documentary category relations, and the theory resulting from the induced model is usually not a universally valid for all cases automatic document categorization. Further, since a theory of automatic document categorization is usually probabilistically causal, it is not possible to invalidate the theory with a single counterexample.

Hjørland (2005, p. 146) brings up two (purported) examples of *positivism* in library and information science, although on dubious premises. Hjørland claims that studies of consistency between indexers “seem” to be based on the premise that there is *one* correct way of indexing documents – but this is in our opinion not sufficiently justified. A reasonable assumption is that one has simply observed that different indexers generate *different* lists of indexing terms, potentially

causing problems for the retrieval of these documents. These studies are not *a priori* based on the perception of a “correct” indexing. It is also difficult to find support for the claim that researchers conducting these studies consider the indexers as “machines that make mistakes”. A more reasonable description is that the research focus has not been on *explaining* the results, but rather to *map* them – which has involved a quantitative data collection and analysis. This focus is in itself not enough reason to characterize this research tradition as positivistic. As Hjørland himself points out (2005, p. 136) the presence of quantitative methodologies is not a sufficient condition for characterizing research as positivistic.

In research on automatic document categorization the human classifier also plays an important, but anonymous, role. The quality of the automatic categorization is assessed by its similarity to a manually created categorization (a *gold standard*), which is assumed to reflect an agreeable partition of the documents. In studies of automated document categorization the underlying decisions on the manually created categorization are not commonly discussed, e.g. what level of consensus that existed, or how the human classifiers made the categorization decisions. Similar to Hjørland’s description of the depersonification of the indexers it is only assumed that there *is* a categorization against which the machine-based result can be assessed.

## Chapter 3

# Document classification

Classification is one of the fundamental practices of *knowledge organization* and has traditionally had a natural role in the arrangement of the physical assets of the library. The basis for this practice is to enable library users to efficiently retrieve literature on a given topic. Buchanan (1979, p. 11) writes:

When the number of documents becomes too great for a person seeking a particular message to scan through all of them it becomes necessary to organize them; when this task becomes too great to be performed informally it is institutionalised – that is, specialists are appointed to carry out the task.

An idea recurrent in the classification literature is that one of the fundamental objectives of library classification is to generate a *structure* of the library's document collection so to make the resources optimally relocatable. Marcella & Newton (1994, p. 3) write that the object of library classification is to “create and preserve a subject order of maximum helpfulness to information seekers”. In this chapter

we will present some of the basic principles of document classification in libraries and how the aim for an optimal structure is being implemented. In chapter 4 we will endeavor to formulate the notion of classification *structure* in a more precise fashion, using a selection of mathematical theories.

### 3.1 Definitions

In this chapter, and in this work as a whole, we are only concerned with the classification of *text documents*. Other activities that can reasonably be labelled “classification”, such as the scientific classification of phenomena, will not be explicitly considered. The possible event that other aspects of classification could be included in the definitions below, especially the general formulations, is thus coincidental. In this chapter there will not be any deliberate attempts to distinguish between the classification of *textual* documents and the categorization of other document formats such as images. On an abstract conceptual level such a distinction is not necessary, whereas the actual procedures and the classification schemes used will possibly be different.

#### 3.1.1 Class and classification

There are several activities and objectives associated with the term *document classification*, but a common denominator in the literature is that the result of classification is a division of a collection of documents into *groups* (Buchanan, 1979, p. 9). Typically these groups consist of documents that have certain *similarities* with each other, for instance with regard to content, literary form, or target groups of users. Marcella & Newton (1994, p. 3) formulate the following, fairly

user-oriented, definition of (library) classification:

The systematic arrangement *by subject* of books and other learning resources and/or the similar systematic arrangement of catalogue or index entries, in the manner most useful to those who are seeking *either* a definite piece of information *or* the display of the most likely sources for the effective investigation of a subject of their choice.

The definition above stresses the *usefulness* of the structure imposed on the library resources as the well as the use of *document subject* as the basis for partitioning the document collection. Although, strictly speaking, any property of the documents could be used to generate a division of the documents, the generally most useful aspect is considered to be the subject of the document. In a similar vein Taylor & Miller (2006, p. 529) provide the following definition of library classification:

The placing of subjects into categories; in organization of information, classification is the process of determining where an information package fits into a given hierarchy and then assigning the notation associated with the appropriate level of the hierarchy to the information package and to its surrogate record.

In addition to the definition given by Marcella & Newton (1994) the formulation by Taylor & Miller (2006) involves another element central to library classification, namely the procedure of assigning *symbols* or *codes* to the documents. The source of permissible classification codes is normally a formalized structure called a *classification scheme*, a concept that will be treated below.

What emerges as ambiguous in the formulations above and other definitions and examples in the literature is the precise meaning of the

term *class* in the context of document classification. It is variously used as a designation for

1. a *grouping* of objects or concepts (e.g. Reitz, 2004, p. 144),
2. a subset of a document collection, defined by a common subject or any other basis of division (e.g. Buchanan, 1979, p. 12),
3. an element of a classification schedule (e.g. Slavic, 2008, p. 260).

We will endeavor to show that these apparently inequivalent definitions of *class* converge into the same kind of dual relation as the dichotomy between a *word* (a sign in a language) and its *senses* (the significations of the word).

### 3.1.2 Classification scheme

A *classification scheme* consists of a set of classification codes, one or several ordering relations on the classes and typically a set of *codes* assigned to the classes according to the *notational rules* of the scheme. The set of notated classes together with any ordering relations as well as instructions for the use of the classes is called a *schedule* (Foskett, 1996, p. 147). The core of the classification scheme, i.e the collection of classification codes, will in this work be referred to as a *classification vocabulary*. As a service to the user an *alphabetical index* may also be provided in the classification scheme.

If all fundamental classification subjects in the scheme are *pre-coordinated* and the corresponding codes explicitly listed in the schedule, the classification scheme is called *enumerative*. Typically such systems are also ordered by *hierarchical* relations between the codes. Prominent examples of enumerative schemes with universal scope and extensive usage in libraries are the Dewey Decimal Classification

(DDC) system, the Universal Decimal Classification (UDC) system, and the Library of Congress Classification (LCC) system. As an example of the hierarchical structure in the DDC system, we find that the concept of *violin* is contained in the following structure in the DDC schedule, edition 22 (Dewey et al., 2003):

```
700      The arts - Fine and decorative arts
780      Music
787      Stringed instruments (Chordophones)
787.2    Violins
```

If the classification scheme is intended to be used by *post-coordination* at indexing time, i.e. the eventual classification code is *synthesized* when a particular document is about to be classified, the classification approach is called *synthetic* or *faceted*. The pivotal example of a system encouraging faceted classification is the Colon Classification system developed in the 1930s by the Indian librarian and classification theorist S. R. Ranganathan. As an illustration of this system consider the following oft-cited classification problem (see e.g. Chan, 1994, p. 391):

Research in the cure of tuberculosis of lungs by x-ray  
conducted in India in the 1950s.

having the classification code

L,45;421:6;253:f.44'N5

Here, the first comma sign indicates that the descriptor code 45 (Lungs) pertains to the *personality* facet of the class Medicine (code L). Further, the first semicolon indicates that the descriptor code 421 (Tuberculosis) is a *property* of the lungs and the first colon specifies that the descriptor code 6 (Treatment) is an energy / activity facet of tuberculosis, and so on.

### 3.1.3 Document subject

An organizational process closely related to that of subject classification is subject *indexing*, i.e. the process of assigning keywords to documents. (Chu & O'Brien, 1993) identifies three distinct steps in the process of subject indexing:

1. A *subject analysis* of the document.
2. An *expression in natural language* ("the indexers' words") of the identified subject content of the documents.
3. A *translation* to and expression of the subject content in an indexing language (which is typically a *controlled* vocabulary).

In every phase of the indexing process the indexer has to make a *decision* based on professional considerations. Although the meaning of *subject* may be evident to the information professional, the question is justifiably raised: what is referred to by the *subject* of a document and how does this term relate to *topic* and *concept*? Hjørland (1992) points out that the identification of the subject content of a document may be an ostensibly unproblematic task. For instance, there may be a discrepancy between the title of the document and its actual subject matter. Hjørland further argues that persons from different disciplines with different foci may even have diverging views on what is the core content of a particular document. As a consequence Hjørland (1992, p. 183, 185) suggests that to be useful subject analysis should not only in a mechanical way determine what a document is "about" but also identify the "epistemological potentials" of the document – in other words how the document in question can be of use, presently and in the future.

Langridge (1989, p. 8-9) states that the subject content of a document is identified in response to two basic questions about the document:

1. What is it?
2. What is it about?

In other words, the subject is determined by the *form* of document (which pertains to the angle from which the document is written and the target audience that is implied) as well as *topic* of the document. For instance, a document with the title "The history of writing" has the *form* of a historical treatment, i.e. the angle of the document is to describe a phenomenon from the perspective of its historical development. The *topic* of the document, i.e. its actual subject matter, is *writing*.

### 3.2 Relations in classification schedules

Something that can be discerned in the above discussion of the document classification process is that the classification schedule, i.e. the *vocabulary* restraining the classifier, has an important influence on the resulting categorization and structuration of the documents. We will therefore take a brief look at prominent principles for the construction of classification schedules, as suggested in the literature.

In an article discussing the role of classification for information retrieval Spärck Jones (1970) suggests that classification schemes can be analyzed in response to the following questions. Given a classification scheme and a set of objects:

1. Is the relation between the properties of the objects and the classes of the scheme *monothetic* or *polythetic*? A monothetic

class is established by a minimal set of properties that are necessary and sufficient conditions for membership in the class. A polythetic class, on the other hand, is characterized by a set of properties such that all objects of the class have many of the properties but not all, and every property is in turn possessed by many of the objects in the class (Van Rijsbergen, 1979, chp. 3).

2. Is the relation between the objects *exclusive* or *overlapping*? Stated differently, can an object be a member of several classes? If so, the classes are said to have an overlapping relation, otherwise an exclusive relation.
3. Is the relation between the classes *ordered* or *unordered*? In other words, are the classes naturally comparable so that they can be arranged in (for instance) a linear or hierarchical order – or is any ordering of the classes in principle arbitrary?

In an attempt to analyze the generation of classification schedules Buchanan (1979, p. 17) suggests that class relations appearing in such schedules can be divided into *syntactical* and *hierarchical* relations. A syntactical relationship appears when two or more classes are pre-coordinated to form a combined category. To be precise, this aspect of Buchanan's analysis refers to the *syntactical* generation of *subject formulations*, and not explicitly to the eventual classification codes assigned to the subject formulations. Hierarchical relations, whether such exist between subject formulations or between codes denoting subjects, are characterized by the subordination or inclusion of classes. We will below give a brief characterization of the respective relations.

### 3.2.1 Syntactical relations

Buchanan (1979, p. 18) states that syntactical relations manifest in two different kinds, as *simple* and *composite* classes. He further explains that, as the names suggest, a simple class defines “one kind of thing” whereas composite classes entail “different kinds of things”. From a pragmatic perspective these definitions are problematic since it is clearly the case that what constitutes a *kind* in a given situation is typically dependent on context-bound *expectations* and *purposes* of classifying a set of objects. As pointed out by Hjørland & Pedersen (2005), in relation to an example involving the categorization of geometric figures,

There is no natural or best way to decide whether form or colour is the most important property to apply when classifying the figures. ... It simply depends on the purpose of the classification. We accordingly suggest that a classification is always required for a purpose ...

To resolve this issue it has to be assumed that “kind” in the scope of Buchanan’s treatment of class relations refers to the category of objects that is *explicitly* and *minimally* suggested by the subject terms describing the class. As we notice, this definition of a “simple” class is dependent on the association between class and subject descriptors. Typically, a collection of objects would not adequately indicate the kind attached to a class. It has to be assumed that a term and a phrase unambiguously denote a set of objects. In other words, Buchanan invokes the link between *class* and *semantics*.

#### 3.2.1.1 Simple classes

Decomposing the class relations even further, Buchanan claims that the simple classes can be divided into *elemental* and *superimposed*

classes. An elemental class is said to be defined by just one characteristic property. Conversely, superimposed classes are said to be defined by several characteristics. Again, these definitions are dependent on assumptions on the terms involved in defining the classes. Using a pair of the author's examples, *forests* is said to exemplify an elemental class whereas *tropical forests* is claimed to be a superimposed class. It is not obvious, however, why the class of forests should have just one defining characteristic. It appears therefore to be the case that the issue of whether a simple class is elemental or superimposed is a matter of *decision*. In other words, a set of classes are defined to be elemental and any class that is constructed by a syntactical combination of class symbols is consequently superimposed.

### 3.2.1.2 Composite classes

Composite classes are characterized by "different kinds of things . . . in a relationship of interaction" (Buchanan, 1979, p. 19). When discussing relationships and compositions of classes it is useful to invoke Rudolf Carnap's notion of predicate *intensions*. Carnap defines intensions as follows (Carnap, 1955, p. 42):

... the intension of a predicate ' $Q$ ' for a speaker  $X$  is the general condition which an object  $y$  must fulfill in order for  $X$  to be willing to ascribe the predicate ' $Q$ ' to  $y$ .

In essence, if we regard a class code  $c_1$  assigned to a certain document  $d$  as a *predicate* ascribed to  $d$  it is natural to conceptualize a condition  $\pi_1$  such that  $d$  and all other documents associated with  $c_1$  satisfy  $\pi_1$ . Conversely, to another class code  $c_2$  we can conceptualize a condition (or property)  $\pi_2$  such that every document  $d$  such that  $c_2$  is assigned to  $d$  satisfies  $\pi_2$ . It is further reasonable to assume that Buchanan's notion of "relationship of interaction" entails *conjunction*

of intensions. We will expand on this idea, and how it illuminates expressions such as *broader* and *narrower* terms, in the chapter formalizing subject classification (chapter 4).

A *complex* class is said to be a composition of classes such that the component classes are separable from each other (Buchanan, 1979, p. 19). Complex classes are also characterized by being a combination of concepts that are normally distinct from each other (McIlwaine, 2000, p. 262). For instance, in the class “statistics for librarians” the components do not blend. Statistics is not a kind of librarian and a librarian is not a kind of statistics. In contrast, the components of a *compound* class blend so that each component class becomes an aspect of the other.

### 3.2.2 Hierarchical relationships

Hierarchical relationships in class schedules appear due to natural *subordination / inclusion* of classes or to situation-based *properties* or *activities*. The first-mentioned kind is also called the *generic* relation, characterized by a subordinative *genus – species* relationship (Hutchins, 1975, p. 43; Buchanan, 1979, p. 22; McIlwaine, 2000, p. 8). A *genus* in the context of documentary languages is a generic class, whereas a *species* is a subclass or facet of a genus. Hutchins (1975, p. 34) states that genus – species relations can be expressed by *meaning postulates* which should always hold if the relation is always generic. For instance, a violin is also a string instrument (but not necessarily vice versa). A meaning postulate expressing this condition is:

violin  $\longrightarrow$  string instrument

We can also use first-order logic to express this relationship even more explicitly:

$$\forall x(\text{VIOLIN}(x) \longrightarrow \text{STRING\_INSTRUMENT}(x))$$

In first-order logic it is clear that the relationship holds for *every* object, given that the predicate VIOLIN is applicable to the object. Building on Carnap’s notion that two expressions are synonymous in a language  $\mathcal{L}$  if they have the same intension in  $\mathcal{L}$  (Carnap, 1955, p. 42) we can state that the intension that is true for “string instrument” is also true for “violin”, but not vice versa.

The other kind of hierarchical relationship is the *property of* or *part of* relation (Buchanan, 1979, p. 22; McIlwaine, 2000, p. 8). For instance, the subject of “mutation in dandelions” expresses a property of dandelions. However, it is not a generic relationship since neither subconcept is a kind of the other.

### 3.3 Classification as language use

In the above analysis of classification schemes we have noted that library classification, from a certain perspective, is the act of assigning symbols from classification schedules to documents. We have also seen that the generation of classification symbols is a process involving the generation of new symbols out of existing symbols. Typically, the classification scheme contains a set of rules constraining the generation of symbols. It is therefore justifiable to claim that the vocabulary of a classification schedule is a *formal language*. Some mathematical details related to the notion of formal languages will be presented in chapter 4.

Ranganathan (1989, p. 31) builds on this notion of classification schedules as languages by stating that a classification schedule (which he calls a “system of Class Numbers”) is an artificial language, and that classification is a “translation of the name of a specific subject from a natural language to a classificatory language”. This definition of document classification presumes, as noted above,

1. that documents have or contain specific *subjects*,
2. that the dominant subjects of a document are readily identifiable and nameable, and
3. that it is possible to translate a named subject to a symbol in a classification vocabulary.

Hutchins (1975, p. 7) utilizes the expression *documentary language* (DL) in reference to artificial languages *designed* for the purpose of describing document content in a formalized fashion. The designation “artificial” for DLs is in contrast to *natural language* (NL), which refers to any language used for ordinary human communication, and which is typically not the result of a deliberate design process. The family of documentary languages is partitioned into *indexing languages* (ILs) used to assign subject descriptors to documents and *classificatory languages* (CLs) employed to assign classification symbols (Hutchins, 1975, p. 9). Though DLs share a common denominator with NLS in that both language categories contain signs (or symbols) having a referential meaning, and exhibit sense relations between their respective symbols (Hutchins, 1975, p. 33) there are also important differences, a few of which are listed below.

1. As already mentioned above, a DL is designed for a specific purpose (to describe document content). It is typically limited to precisely that task and may not be suitable for other situations in

which natural languages are used to mediate information, such as to communicate factual information, to incite a certain behavior in the recipient, or to express emotions (Hutchins, 1975, p. 8).

2. A documentary language can not describe itself and therefore not function as its own *metalanguage* (Hutchins, 1975, p. 7).
3. Whereas *synonymy* and *homonymy* are common features in NLs, documentary languages typically have a *standardized* vocabulary in which the amount of synonymy and homonymy is largely reduced (Hutchins, 1975, p. 9). While discussing the semantics of classificatory languages Ranganathan sharpens this characterization by claiming that “there is a strict one-to-one correspondence between class numbers and names of specific subject”, i.e. classificatory languages lack synonymy and homonymy entirely (Ranganathan, 1989, p. 35).

Hutchins (1975, p. 7, 12) states that a DL consists of a set of *descriptors* (the *vocabulary* of the DL) which are syntactically combined to form *descriptor phrases* (the *sentences* of the DL). A descriptor phrase has the production rule (Hutchins, 1975, p. 69)

$$DF \longrightarrow R_1(+R_2 + \dots + R_n)$$

where each  $R_i$  denotes a *role indicator* on the form

$$R_i \longrightarrow r_i + k_j$$

where  $r_i$  is the expression (form) of the role indicator and  $k_j$  is a descriptor attached to the role indicator. Formally, the production rule of a descriptor phrase can be viewed as an  $n$ -ary relation (Gardin, 1973,

p. 148). Assuming that the role indicators of the DF are combined conjunctively (cf. Ranganathan, 1989, p. 34) the  $n$ -ary relations can be translated into conjunctions of *binary* relations. We have noted several examples of descriptor phrases above; in the example of how the Colon Classification is applied and in the discussion about syntactical relations between subject formulations in Buchanan's analysis.

We will in the following chapter proceed to discuss the notion of *subject classification*, using an assortment of formal theories.

## Chapter 4

# Subject classification

The purpose of this chapter is to investigate one of the core concepts in library and information science – subject classification – in light of a selection of relevant mathematical theories. We thereby expect to outline the components of a formal theory of subject classification, in order to reach a reasonably precise definition of the concept and to develop a theoretical tool for analyzing and comparing classification schemes as well as classifications of document collections. In section 3.1.1 we noted that the term *classification* is variably used to denote the process of assigning labels to documents as well as the process of assigning a structure to a set of documents. One of the objectives of this chapter is to show how these definitions complement each other. Further, a discernible property of the physical library is that the conceptual arrangement of documents in classes, and the ordering of documents within a class (typically an alphabetical ordering) results in a “geometrical” structure of the documents within the physical space of the library. Expressed succinctly, the conceptual structuring of documents results in a physical structure of the documents in the library, such that the location of a specific document in the library is a con-

sequence of the underlying conceptual structure. We endeavor in this chapter to formally express and analyze the relationship between the conceptual structure of a classification scheme and the geometrical structure of a classified document collection.

Initially we will lay the groundwork for characterizing documents and classifications of documents as operations on *sets*. We then proceed to present the mathematical notion of *formal languages* and investigate how classification schedules can be regarded and analyzed as such. We have stated in chapter 3 that the act of classifying a collection of documents can be regarded as the act of formulating statements in a documentary language (DL). We have also noted that libraries typically organize their collections by imposing a certain structure (for instance by subject classification) on the documents. A study of such structures could be conducted in two principal ways: by investigating the *internal* “anatomy” of the structures, and by investigating how the structures interact with each other (i.e. an *external* perspective on the structures). In order to study the internal structure of collections we will use *topology*, and for the purpose of studying the interaction between structures we will use a relatively new mathematical area called *category theory*. Following this treatment of document classification as language use we turn our focus to formal structures that emerge by the classification of document collections. The combination of formal languages, category theory, and topology is to our knowledge a novel approach to the theory of subject classification.

## 4.1 Set theory

One of the fundamental branches of modern mathematics is *set theory* (Kolmogorov et al., 1975, p. 1). The concept of a *set* can informally be understood as the mathematical equivalent of a collection of objects. It is therefore natural to involve sets in the formalization of any (unordered) group of abstract or concrete things and phenomena. Not unexpectedly, sets are virtually ubiquitous in the formalization of collections of documents, terms, classes, users etc within information science. For instance, Baeza-Yates & Ribeiro-Neto (2011, p. 58) define an information retrieval model as a four-tuple  $(\mathbf{D}, \mathbf{Q}, \mathcal{F}, r)$ , where  $\mathbf{D}$  is a set of document representations,  $\mathbf{Q}$  is a set of query representations,  $\mathcal{F}$  is the underlying formal framework of the model, and  $r$  is a ranking function. We have also (see section 2.1) presented the general definition of a *target function* in text categorization. This function has the general form  $\varphi : D \times C \rightarrow \{T, F\}$ , where  $D$  denotes a set of documents,  $C$  a set of classes (or categories), and  $\{T, F\}$  is a set of truth values (also called *truth space*, see section 4.3.4).

As a simple example of a set consider the colors of the flag of France, which can be presented in set notation as follows:

$$F = \{\text{blue, white, red}\}$$

An expression on the form  $x \in S$  is a proposition stating that the object denoted by the variable  $x$  is an element (or member) of the set  $S$ . In the example set above,  $\text{red} \in F$  holds, but not  $\text{blue} \in F$  (typically written  $\text{blue} \notin F$ ). In the Zermelo-Fraenkel (ZF) set theory one of the central axioms, the *axiom of extensionality*, has the following formulation (see e.g. Hrbacek & Jech, 1999, p. 267):

**Axiom of extensionality.** *Let  $A$  and  $B$  be sets. If every element of  $A$*

is an element of  $B$ , and every element of  $B$  is an element of  $A$ , then  $A = B$ .

In other words, two sets are *equal* if they contain precisely the same elements. It immediately follows that a set is *completely defined* by the elements being members of the set. From this axiom we can also deduce two fundamental properties of sets.

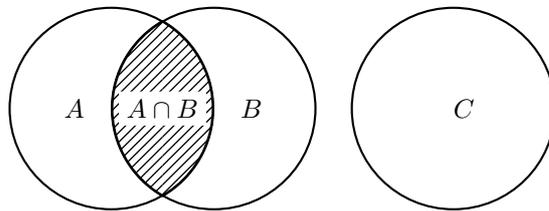
1. No existing order between the elements is “preserved” by the set.
2. The membership relation is not qualified by a quantity, which means that a set cannot preserve any multiplicity of an element.

A set can be defined in two ways: by using a member condition (see e.g. Deskins, 1995, p. 2) or *extensionally* by explicitly listing all its elements. The number of elements that are members of a set  $S$  is written  $|S|$  and is called the *cardinality* of  $S$  (Hrbacek & Jech, 1999, p. 65).

#### 4.1.1 Operations on sets

Let  $A$  and  $B$  be sets.  $B$  is called a *subset* of  $A$ , written  $B \subseteq A$ , if all elements in  $B$  are also elements of  $A$ . If  $B \subseteq A$  holds and there is at least one element in  $A$  that is not an element of  $B$  we call  $B$  a *proper subset* of  $A$  (Deskins, 1995, p. 3). The *intersection* between  $A$  and  $B$ , written  $A \cap B$ , is the set of all elements  $x$  such that  $x$  is an element of *both*  $A$  and  $B$  (cf. the crosshatched area in figure 4.1). Conversely, the *union* of  $A$  and  $B$ , written  $A \cup B$ , is the set of all elements  $x$  such that  $x$  is an element of *at least one of* the sets  $A$  and  $B$ .

The *empty set*, commonly denoted  $\emptyset$ , is the set that does not contain any elements. Two sets  $A$  and  $B$  are said to be *disjoint* if their



**Figure 4.1.** A Venn diagram illustrating basic set operations and set relations.

intersection is the empty set (i.e. the sets have no element in common). In figure 4.1 above, the sets  $A$  and  $C$  are disjoint. A *partition*  $\mathfrak{P}(A)$  of a set  $A$  is a collection of subsets of  $A$  such that the subsets are pairwise disjoint and their union equals  $A$  (Hrbacek & Jech, 1999, p. 31). Formally:

$$\begin{aligned} \mathfrak{P}(A) &= \{A_1, A_2, \dots, A_n\} \quad \text{such that} \\ A_i \cap A_j &= \emptyset \quad \text{for all } i \neq j \\ \bigcup_{i=1}^n A_i &= A \end{aligned}$$

A partition can therefore be alternatively defined as a division of a set  $A$  into subsets, such that each element in  $A$  is in precisely one subset in the partition. Finally, the power set  $\mathcal{P}(A)$  of a set  $A$  is the set of all subsets of  $A$  (Deskins, 1995, p. 11). For instance, the power set of  $\{0, 1\}$  is  $\{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$ .

#### 4.1.1.1 Functions on sets

A *function*  $f$  between two sets  $A$  and  $B$  (often declaratively written  $f : A \rightarrow B$ ) is, informally, a mechanism for stating associations between the elements of  $A$  and  $B$  respectively. More precisely, a function  $f : A \rightarrow B$  is a binary relation, i.e. a *set* of ordered pairs  $(a, b)$ , such

that  $a \in A$  and  $b \in B$ . The set  $A$  is called the *domain* of  $f$  and  $B$  the *range* of  $f$  (Hrbacek & Jech, 1999, p. 19, 23). If an ordered pair  $(a, b)$  belongs to  $f$  we say that  $f$  *maps* the element  $a$  onto the element  $b$ . It is further stipulated that  $f$  does not map a given element  $a \in A$  onto more than one element  $b \in B$ . In other words,  $f$  maps elements between  $A$  and  $B$  in a *deterministic* fashion. A more formal way to state this condition on functions is given as follows.

**Condition on functions.** *For any function  $f : A \rightarrow B$  it holds that if  $(a, b) \in f$  and  $(a, c) \in f$  then  $b = c$ .*

Implied in this notion is that the function constitutes a *rule* for mapping elements of a domain  $A$  on to the elements of a range  $B$  (Munkres, 2000, p. 16). This rule can be specified as a formula, for instance  $f(x) = x^2 - 1$ , or even  $f = \{(x, x^2 - 1) | x \in \mathbb{R}\}$  to make the domain of the function explicit (Hrbacek & Jech, 1999, p. 24). The latter notation also makes it clear that a function is also a set.

#### 4.1.1.2 Equivalence classes

The notion of *equivalence* among a collection of objects entails a property that is equal between the objects. For instance, among polygons any two rectangles (regardless of size and the proportion between the lengths of the edges) are equivalent with regard to the number of edges making up the shape. Also, any two documents that are assigned the same class label can be regarded as equivalent with regard to the property of class membership. Formally, an *equivalence relation*  $\simeq$  over a set  $A$  is a binary relation consisting of all pairs of objects for which equivalence holds (Deskins, 1995, p. 10). An equivalence relation is by definition *reflexive* ( $x \simeq x$  for all  $x \in A$ ), *symmetric* (if  $x \simeq y$  then also  $y \simeq x$ ) and *transitive* (if  $x \simeq y$  and  $y \simeq z$  then also

$x \simeq z$ ). It is easily verified that, for example, the relation “is born in the same year as” satisfies the conditions for an equivalence relation.

Given a set  $A$ , an equivalence relation  $\simeq$  over  $A$ , and an element  $x \in A$ , the *equivalence class* of  $x$  (written  $[x]$ ) is defined as the set of all elements in  $A$  for which equivalence with  $x$  holds (Hrbacek & Jech, 1999, p. 30). Formally:

$$[x] := \{y \in A : x \simeq y\}$$

The set of all equivalence classes defined on the set  $A$  over the equivalence relation  $\simeq$  is called the *quotient set* of  $A$  by  $\simeq$ , and is denoted by  $A/\simeq$  (Bourbaki, 2004, p. 115).

It is easily verified that two equivalence classes over the same equivalence relation are either equal or disjoint. For any function  $f : A \rightarrow B$  an equivalence relation  $\simeq_f$  is naturally induced by considering the values in the domain of  $f$  yielding the same output from the function, i.e.

$$x \simeq_f y \text{ iff } f(x) = f(y)$$

## 4.2 Formal languages

In order to facilitate a working definition of documents as well as document collections we need a conceptual framework for expressing their “constituents”, the building blocks of which we can proceed to generate such objects. To this end we present the notion of a *formal language*. Let  $A = \{a_1, a_2, \dots\}$  be a finite alphabet of symbols. We call a sequence of symbols from  $A$  (as well as the symbols themselves) a *word*. We further define a binary operation  $\oplus$ , called *concatenation*,

such that for any two words  $w$  and  $v$

$$w \oplus v = wv$$

Let  $A^*$  be the set of all possible words over  $A$ . Clearly, if  $w, v \in A^*$  then also  $w \oplus v \in A^*$ . We say that  $A^*$  is *closed* (see e.g. Hrbacek & Jech, 1999, p. 60) under concatenation. Let us further assume that there is an *empty word*  $\varepsilon$  in  $A^*$ , such that for all  $w \in A^*$  it holds that

$$w \oplus \varepsilon = \varepsilon \oplus w = w$$

Every subset of  $A^*$  is called a *formal language* (Crespi Reghizzi et al., 2013, p. 8). For instance, given the alphabet  $\{0, 1, 2, \dots, 9\}$  we can define two languages  $L_1$  and  $L_2$  as

$$\begin{aligned} L_1 &:= \{12, 375, 48, 610\} \\ L_2 &:= \{35, 197, 643, 1784, 40162\} \end{aligned}$$

The algebraic structure  $(A^*, \oplus, \varepsilon)$  is called a *monoid* (Pierce, 1991, p. 4), more specifically a *free monoid* over  $A$  (Mac Lane, 1998, p. 50).

#### 4.2.1 Document collections as formal languages

It is also possible to define a textual document collection in terms of a free monoid over a vocabulary. Following Mehler et al. (2007) we define a set  $T$  of *types*, which can be thought of as the equivalent of *lexemes*, and a set  $S$  of *forms* (morphological variations of the types in  $T$ ). Each instance of a form in a text string is in turn called a *token*. The *concatenation* of two forms yields a *text string* (which is the equivalent of a list of forms). For example, the forms “hello” and “world” can be concatenated to form the text string “hello world”. We further define the *length*  $\ell(w)$  of a string  $w$  as the number of tokens

contained in the  $w$ . The string (from Gertrude Stein’s poem *Sacred Emily*)

*rose is a rose is a rose is a rose*

contains 10 tokens but only 3 forms (and 3 types). Letting  $\oplus$  denote concatenation we now define a text document (or just “document”, since that is the only kind of document we are considering here) as a string over  $S$ , i.e.

1. every form in  $S$  is a document, and
2. if  $w$  and  $v$  are documents, then also  $w \oplus v$  is a document.

#### 4.2.2 Classification schedules as formal languages

According to Hutchins (1975) as well as Ranganathan (1989) a classification schedule can be regarded as a *language* with a distinct vocabulary, semantic interpretations, and syntactic rules.

Consider an alphabet  $A$  of atomic classification symbols. We call this alphabet *base of notation* (see e.g. Ranganathan, 1989, p. 102). We define a *classification vocabulary* as a formal language  $\Omega$  such that every element  $\omega \in \Omega$  is *valid* in the sense that it can be obtained from  $A$  by the iteration of a set of rules. The words in  $\Omega$  are called *classification codes*. We further assume that every element  $\omega$  has a *semantic interpretation*, which is the *subject* assigned to  $\omega$ . To  $\Omega$  we therefore associate a corresponding *subject space*  $B$  having an implicit functional relationship  $\mu : \Omega \rightarrow B$ , with  $\mu(\omega) = b$  meaning that the symbol  $\omega$  *denotes* the subject  $b$ . This is in line with the characterization of classificatory languages described by Ranganathan (1989, p. 35):

In a classificatory language, there is a strict one-one correspondence between class numbers and names of specific subjects, *i.e.* the name of each specific subject can be translated into one and only one class number and each class number can denote one and only one specific subject  
 ...

The number of possible classification codes in  $\Omega$  is called the *capacity* of  $\Omega$  (Ranganathan, 1989, p. 103). If every classification code in  $\Omega$  consists of  $n$  symbols it is easily verified that the capacity of  $\Omega$  is  $|A|^n$ . For instance, a classification vocabulary  $\Omega$  defined on an alphabet with 5 available symbols and all codes in  $\Omega$  having a code length of 4, will have the capacity  $5^4 = 625$ .

**Proposition 4.2.1.** *For any classification vocabulary  $\Omega$ , defined over an alphabet  $A$ , with a maximal code length of  $n$  the capacity of  $\Omega$  is (cf. Ranganathan, 1989, p. 104)*

$$k(A, n) = \frac{|A|^{n+1} - |A|}{|A| - 1} \quad (4.1)$$

*Proof.* The capacity of  $\Omega$  with the *exact* code length  $n$  is  $|A|^n$ . Therefore, the capacity *up to* code length  $n$  has to be

$$|A| + |A|^2 + \cdots + |A|^n \quad (4.2)$$

It is easily verified that  $(|A| + |A|^2 + \cdots + |A|^n)(|A| - 1) = |A|^{n+1} - |A|$ . Proposition 4.2.1 immediately follows.  $\square$

**Proposition 4.2.2.**  *$\Omega$  is at most countably infinite, *i.e.*  $\Omega \leq \aleph_0$ , assuming that every element in  $\Omega$  has finite length.*

*Proof.* We note that  $\Omega \subseteq A^*$ . It follows that  $|\Omega| \leq |A^*|$ . If there are no restrictions on the length of the symbols in  $A^*$ , beyond the criterion

that the length is always finite, it is easily shown that  $A^*$  has infinite cardinality. Assume that  $|A^*| = n$ , where  $n$  is any positive integer. If we now *add* symbols to  $A^*$  by concatenating every string  $w \in A^*$  with an arbitrary symbol  $\sigma \in A$  we get that  $|A^*| = 2n$  since to every original string  $w$  there also exists a corresponding string  $w\sigma$ . This leads to a contradiction of our original assumption and consequently  $A^*$  is infinite. We will now show that  $A^*$  is *countable* by demonstrating that we can assign a unique natural number to each string  $s \in A^*$ .

We let  $A = \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}\}$  and impose an (arbitrary) ordering on  $A$  to facilitate an enumeration of the elements in  $A$ , for instance  $\mathbf{a} \prec \mathbf{b} \prec \mathbf{c} \prec \mathbf{d} \prec \mathbf{e}$ . We now let the notation  $\mathbf{a} \otimes n$  denote a string consisting of  $n$  instances of the symbol  $\mathbf{a}$  (e.g.  $\mathbf{a} \otimes 3 = \mathbf{aaa}$ ) and proceed to define a successor function  $S : A^* \rightarrow A^*$  as follows:

$$\begin{aligned} S(\mathbf{a}) &= \mathbf{b} \\ S(\mathbf{b}) &= \mathbf{c} \\ S(\mathbf{c}) &= \mathbf{d} \\ S(\mathbf{d}) &= \mathbf{e} \\ S(\mathbf{e}) &= \mathbf{aa} \\ S(x_1 \oplus x_2 \oplus \cdots \oplus x_n) &= \begin{cases} x_1 \oplus x_2 \oplus \cdots \oplus S(x_n) & \text{if } x_n \prec \mathbf{e} \\ S(x_1) \oplus (\mathbf{a} \otimes (n-1)) & \text{if } x_1 \prec \mathbf{e} \text{ and } x_2, \dots, x_n = \mathbf{e} \\ \mathbf{a} \otimes (n+1) & \text{otherwise} \end{cases} \end{aligned}$$

This yields a sequence of “successor strings” analogous to the generation of numbers in a positional system (for instance the decimal system). We can now recursively define a one-to-one map  $\psi : A^* \rightarrow \mathbb{N}$  as follows:

$$\begin{aligned} \psi(\mathbf{a}) &= 1 \\ \psi(S(w)) &= \psi(w) + 1 \quad \text{if } \mathbf{a} \prec w \end{aligned}$$

□

We will elaborate further on the notion of classification vocabularies as formal languages in sections 4.4.1 and 4.7.5 with the objective to show the relationship between a classification schedules as formal languages and the structure imposed by using a classificatory language.

### 4.3 Category theory

Category theory is a quite young branch of mathematics (Pierce, 1991, p. xi), dealing in a generalized way with the characterization of structures and relations between structures (called *categories*) such as sets, partial orders, groups, rings, and vector spaces. We will in this and the subsequent chapter use categories and *functors* between categories to state how collections of information entities can be represented and transformed. A commonly used framework to define processes within machine classification and information retrieval is to use sets and functions between sets. The advantage of such an approach is the conceptual simplicity of the notion of sets, due to the lack of structure within sets. However, an approach based on category theory does not imply any assumptions about any existing structure (or lack of structure) between the elements involved in the formalization. For instance, it is not assumed that a document collection can or should be modeled as a set (which is by definition an unordered category), rather than for instance a partial or total order (see section 4.5).

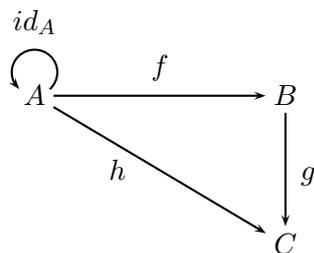
#### 4.3.1 Definitions

A *category* (see Pierce, 1991, p. 1) consists of

1. a collection of *objects*,

2. a collection of *morphisms* between these objects,
3. an *identity morphism* mapping each object in the category onto itself, and
4. a *composition operator*  $\circ$  that is associative over all morphisms.

We will illustrate these components of a category by referring to the diagram in figure 4.2 below. In this diagram we find three objects ( $A$ ,  $B$ , and  $C$ ) as well as three morphisms ( $f$ ,  $g$ , and  $h$ ) between these objects. We also find an identity morphism  $id_A$  mapping the object  $A$  onto itself. The *composition* of two morphisms, for instance  $f$  and  $g$ , is written  $g \circ f$ . The meaning of the composition  $g \circ f$  is that the mapping  $f$  is applied first, mapping  $A$  onto  $B$ , followed by an application of  $g$ , mapping  $B$  onto  $C$ . It follows that the composition  $g \circ f$  is a mapping of  $A$  onto  $C$ . The diagram in figure 4.2 is said to *commute* iff  $h = g \circ f$ .

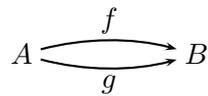


**Figure 4.2.** A simple category diagram.

An example of what the components of diagram 4.2 could represent can be given by considering an information seeking scenario. Let  $A$  denote the entire collection of documents (or, which often is the case, document surrogates) that are available in a certain information system. By an initial query the user obtains a subcollection  $B$  of these

documents. The mapping  $f$  then represents an initial selection of the documents in  $A$ . By a *refining* query the user performs a search within the documents in  $B$ ; thereby obtaining a subcollection  $C$ . The mapping  $g$  represents a selection of the documents in  $B$ . Finally, the arrow  $h$  represents the search strategy that combines both an initial query and a refinement of the query.

In addition to the definitions above we also introduce the notion of a *hom-set* (Mac Lane, 1998, p. 10). Given a category  $\mathbf{C}$  and two objects  $A, B$  in  $\mathbf{C}$ , then the hom-set of  $A$  and  $B$ , written  $\mathbf{hom}_{\mathbf{C}}(A, B)$  is simply the set of all morphisms between  $A$  and  $B$  (see figure 4.3).



**Figure 4.3.** The hom-set  $\mathbf{hom}(A, B)$  is the set of morphisms  $\{f, g\}$ .

An important concept in category theory is that of *functors*, which formalize translations between various categories. Let  $\mathbf{C}$  and  $\mathbf{D}$  be categories. A functor  $F$  is a map  $F : \mathbf{C} \rightarrow \mathbf{D}$  such that (Pierce, 1991, p. 36)

1. every object  $X$  in  $\mathbf{C}$  is mapped to an object  $F(X)$  in  $\mathbf{D}$ , and
2. every morphism  $f : X \rightarrow Y$  in  $\mathbf{C}$  is mapped to a morphism  $F(f) : F(X) \rightarrow F(Y)$  in  $\mathbf{D}$ .

Furthermore, functors preserve identity morphisms and compositions of morphisms. As a simple example, consider the product rule for logarithms. Let  $x$  and  $y$  be positive real numbers. Then, for any positive basis of the logarithm, it holds that  $\log(xy) = \log x + \log y$ . This rule can be expressed as an endofunctor (Pierce, 1991, p. 40) from the category  $\mathbf{Set}$  to itself as shown in figure 4.4.

$$\begin{array}{ccc}
 A \times B & & \log(A) \times \log(B) \\
 \downarrow \times & \xrightarrow{F} & \downarrow + \\
 C & & \log(C)
 \end{array}$$

**Figure 4.4.** The product rule for the logarithm expressed in terms of a functor  $F$ .

### 4.3.2 Document collections as categories

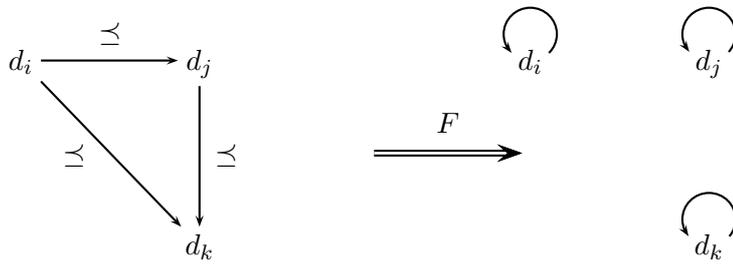
In order to study the notion of document collections using a category-theoretical approach, we begin by tentatively stipulating the existence of category  $\mathbf{Doc}$ , representing any collection of documents. Such a category would at the very least contain:

1. one object for each document in the collection, and
2. identity arrows on the document objects.

Let us begin with the assumption that those are the only morphisms contained in  $\mathbf{Doc}$ , i.e.

$$\mathbf{hom}_{\mathbf{Doc}}(X, Y) = \begin{cases} \{\text{id}_X\} & \text{if } X = Y \\ \emptyset & \text{otherwise} \end{cases}$$

Under this assumption  $\mathbf{Doc}$  is said to be a *discrete* category and is equivalent to a *set*. If we instead stipulate a certain *structure* on the objects in  $\mathbf{Doc}$ , such that for each pair  $(d_j, d_k)$  of objects in  $\mathbf{Doc}$  there is at most one morphism between  $d_j$  and  $d_k$ , we say that  $\mathbf{Doc}$  is a *preorder* (Mac Lane, 1998, p. 11). Examples of preorders in the context of document collections are the linear order of an alphabet arrangement, and the graph structure of a hypertext network.



**Figure 4.5.** A preorder can be transformed into a set by mapping every morphism in the preorder onto the corresponding identity morphisms.

For every preorder  $\preceq$  it is possible to define a functor  $F$  such that  $F$  maps every object onto itself (i.e.  $F$  is *isomorphic* with respect to the objects), and every morphism in  $\mathbf{hom}_{\mathbf{C}}(X, Y)$  onto the identity morphism  $\text{id}_X$  in  $\mathbf{D}$ . If we consider this operation on a higher level such that  $F$  is a functor from the category **Preord** of preordered sets to the category **Set** of sets, we say that  $F$  is a *forgetful* functor (Mac Lane, 1998, p. 14) since it “forgets” the structure induced by the preorder. The conclusion we can make of this so far is that the category **Doc** is a set or can be mapped onto a set by means of a forgetful functor. We note that the category of sets is a natural fundamental framework to use in the formalization of document collections (and other structured or unstructured collections, such as vocabularies).

We have previously, in the context of formal languages (see section 4.2), defined the algebraic structure known as a *monoid*. A *free monoid*  $M^\#(X)$  on a set  $X$  is the monoid generated by forming all possible (finite) strings from  $X$  using the concatenation operator (see e.g. Mac Lane, 1998, p. 50). Hence,  $\mathcal{M}_{S^*} := (S^*, \oplus, \varepsilon)$  is the free monoid over the vocabulary  $S$ .

A collection of monoids together with morphisms between the monoids form a **Mon** category. Consider the length  $\ell(w)$  of a form in  $S$ . In this context we consider length as a function that returns the number of tokens in a string, and therefore  $\ell(w) = 1$ . We define a corresponding length morphism  $\ell^\sharp$  for the elements in  $S^*$ . Since for all  $w, v \in S^*$  it holds that

$$\ell(w \oplus v) = \ell(w) + \ell(v)$$

we find that  $\ell$  is a *homomorphism* between  $S^*$  and  $\mathbb{Z}_{\geq 0}$  (the set of non-negative integers). Let  $\mathcal{M}_Z := (\mathbb{Z}_{\geq 0}, +, 0)$  be the monoid over the set of non-negative integers. We further define  $M_b$  as the forgetful functor  $\mathbf{Mon} \rightarrow \mathbf{Set}$ . Then there is a map  $\ell^\sharp : M^\sharp(S) \rightarrow \mathcal{M}_Z$ , called the *homomorphic extension* of  $\ell$ , such that the diagram in 4.6 commutes (see Pierce, 1991, p. 46):

$$\begin{array}{ccc}
 S & \xrightarrow{\eta} & (M_b \circ M^\sharp)(S) \\
 & \searrow \ell & \downarrow M_b(\ell^\sharp) \\
 & & M_b(\mathcal{M}_Z)
 \end{array}$$

**Figure 4.6.** A diagram in the category **Set**.

A conclusion we can draw from the diagram in figure 4.6 is that it commutes under a very specific condition, namely that  $\ell^\sharp$  is a map from  $M^\sharp(S)$  to  $\mathcal{M}_Z$  iff  $\ell$  is a map from  $S$  to  $\mathbb{Z}_{\geq 0}$ . Expressed formally:

$$\ell^\sharp(M^\sharp(S), \mathcal{M}_Z) \cong \ell(S, M_b(\mathcal{M}_Z))$$

There is a special relationship between the functors  $M^\sharp : \mathbf{Set} \rightarrow$

$\mathbf{Mon}$  and  $M_b : \mathbf{Mon} \rightarrow \mathbf{Set}$  called *adjunction*. Generally, two functors  $F$  and  $G$  are said to be *adjoint* iff there is a bijective relationship (on the form “if and only if”) between  $FX \rightarrow Y$  and  $X \rightarrow GY$ . In this adjunction,  $M^\sharp$  is called the *left adjoint* and  $M_b$  the *right adjoint* (Mac Lane, 1998, p. 81). This is a relationship that “arise[s] everywhere” in mathematics, according to (Mac Lane, 1998, p. vii). We can also illustrate this relationship by defining a document space  $\mathcal{M}_D := (D, *, \varepsilon)$  such that  $\mathcal{M}_{S^*} \sqsubseteq \mathcal{M}_D$ , where we use  $\sqsubseteq$  to denote a *subcategory* (more specifically: the *submonoid*) relation. Then the diagram in figure 4.7 commutes:

$$\begin{array}{ccc}
 M^\sharp(S) & & S \\
 \eta \downarrow & \cong & \downarrow \eta \\
 \mathcal{M}_D & & M_b(D)
 \end{array}$$

**Figure 4.7.** The functor mapping a basis  $S$  onto the free monoid induced by  $S$  and the forgetful functor are adjoints to each other.

In the diagram in figure 4.7 the symbol  $\eta$  denotes the *inclusion map*<sup>1</sup> from one object to another, and  $\cong$  denotes a *bijective* relation. This diagram attempts to visually express the fact that the free monoid  $M^\sharp(S)$  over a vocabulary  $S$  is embedded in the document space  $\mathcal{M}_D$  iff the vocabulary  $S$  is embedded in the underlying set  $D$  of  $\mathcal{M}_D$ .

<sup>1</sup>The reader should note that we use the same inclusion symbol, despite the inclusion operation taking place in two different categories.

Formally:

$$\eta(M^\sharp(S), \mathcal{M}_D) \cong \eta(S, M_b(D))$$

It is easily verified that this relation holds. Assume that  $S$  is not a subset of  $D$ . Then,  $S^*$  cannot be a subcategory of  $\mathcal{M}_D$  since  $S$  is a subset of  $S^*$ . Conversely, assume that  $M^\sharp(S)$  is not embedded in  $\mathcal{M}_D$ . Then it cannot be the case that  $S$  is a subset of  $D$  since  $S$  is a subset of  $S^*$ .

This adjunction sheds interesting light on the relationship between a *basis* (in this case  $S$ ) and the space generated by this basis, into which the document representations are embedded. There is a perfect analogue in the *vector space model* (see chapter 5.4), in which the vector space is generated by the initial set of term vectors and the document vectors are embedded in the resulting space. Additionally, a set of non-overlapping classes form the basis of a *class topology* into which the document classes of a concrete classification may be embedded.

### 4.3.3 Classification and category theory

We will now attempt to find a way to precisely define a *class* in category-theoretical terms, i.e. the subcollection of documents to which a certain classification code has been assigned. We noted previously that document collections (and consequently document subcollections) equal or correspond to sets and it would therefore be natural to attempt to define classes in the category **Set**. A problem involved in such an approach is, however, that the fundamental units in the category **Set** are sets, which means that we cannot define in a class in terms of the *content* or *structure* of sets, in this category. Instead, we

have to proceed by trying to define classes by means of morphisms to *other* objects in the same category.

#### 4.3.4 Subobject classifiers

We begin by defining a two-valued *truth space*  $\mathbb{T} := \{0, 1\}$  (see e.g. Nguyen, 1995, p. 144) such that 0 is interpreted as “false” and 1 as “true”. Let  $D$  be an object in the category **Set** and  $S \subseteq D$  any subset of  $D$ . We define a *characteristic function*  $\chi_S : D \rightarrow \{0, 1\}$  (see e.g. Hrbacek & Jech, 1999, p. 91) as follows:

$$\chi_S(x) := \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{if } x \notin S \end{cases}$$

With this function at our disposal it is very easy to define a class  $C_j$  as the subset of  $D$  containing all the documents  $d_i$  for which  $\chi_{C_j}(x) = 1$ . Formally:

$$C_j := \{x \in D : \chi_{C_j}(x) = 1\}$$

However, there is no straightforward way to express this definition using category theory since there are no operations defined for “looking into” and analyzing the objects. Instead, we need to utilize relations between objects within the category **Set** and, if necessary, objects of other categories. Let us begin with investigating the function (or morphism)  $\chi_{C_j}$ .

The *domain* of  $\chi_{C_j}$  is the object representing the entire document collection, and its codomain consists of the truth space. In other words, the diagram in figure 4.8 is a representation of the classifier  $\varphi_j : D \rightarrow \{0, 1\}$ . In order to restrict the diagram to those elements in  $D$  which have been assigned to  $C_j$  we begin by adding the singleton set  $\mathbf{1} = \{1\}$ , which is a terminal object in **Set**. A *terminal object*  $\mathbf{1}$

$$D \xrightarrow{\chi_{C_j}} \mathbb{T}$$

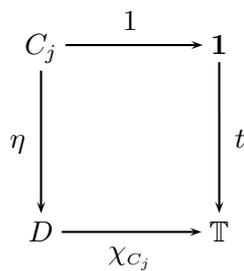
**Figure 4.8.** A diagram representing the characteristic function  $\chi_{C_j}$ .

(see e.g. Pierce, 1991, p. 16) in a category  $\mathbf{C}$  is an object such that for every other object  $A$  in  $\mathbf{C}$  there exists exactly one morphism  $A \rightarrow \mathbf{1}$ . In the category  $\mathbf{Set}$  every singleton set is a terminal object.

$$\begin{array}{ccc} & & \mathbf{1} \\ & & \downarrow t \\ D & \xrightarrow{\chi_{C_j}} & \mathbb{T} \end{array}$$

**Figure 4.9.** The diagram representing  $\chi_{C_j}$  augmented by a mapping from a singleton set containing the value 1 (“truth”).

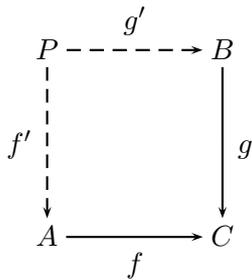
We finally insert the class  $C_j$  into the diagram, using the ordinary assumption that the diagram commutes, i.e.  $t \circ \eta = \chi_{C_j} \circ \eta$ . It would now be tempting to define  $C_j$  as the set which makes the diagram commute. However, it turns out that such a definition is not sufficiently precise. Assume that the diagram commutes and that  $C_j$  is non-empty. Let  $C'_j$  be any proper subset of  $C_j$  and substitute  $C'_j$  for  $C_j$  in the diagram. It turns out that the diagram still commutes! There is also no straightforward category-theoretical way of expressing “the largest set such that...” which means that we need to use another approach, namely that of pullbacks.



**Figure 4.10.** A commutative diagram over the objects  $C_j$  (a class),  $D$  (a document collection),  $\mathbf{1}$  (a terminal object), and  $\mathbb{T}$  (the truth space).

#### 4.3.4.1 Pullback

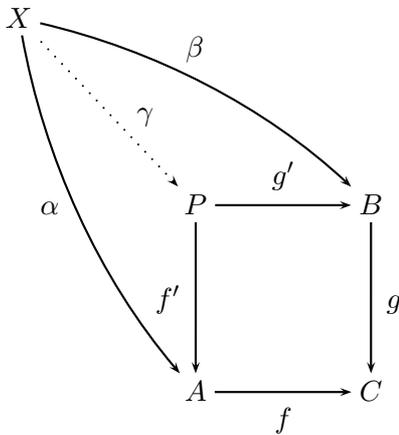
Let  $A$  and  $B$  be objects and  $f : A \rightarrow C$  and  $g : B \rightarrow C$  morphisms in a category  $C$ , as in figure 4.11 below.



**Figure 4.11.** A diagram containing two morphisms  $f$  and  $g$  having different domains, but the same codomain.

A *pullback* (Mac Lane, 1998, p. 71; Pierce, 1991, p. 22) on  $f$  and  $g$  is an object  $P$  together with two morphisms  $f'$  and  $g'$  such that the diagram in figure 4.11 commutes and such that the pullback induced by  $(P, f', g')$  is *universal*. By this latter property is meant that for every object  $X$  together with the morphisms  $\alpha$ ,  $\beta$ , and  $\gamma$  the diagram

in figure 4.12 commutes.



**Figure 4.12.** A diagram illustrating a pullback.

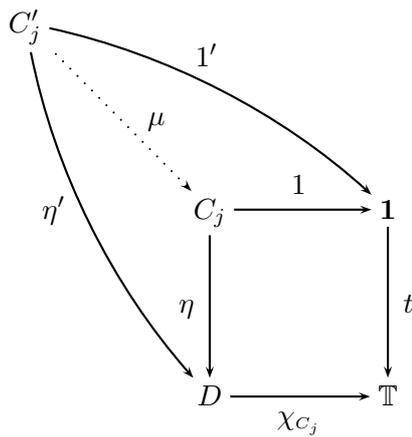
**Proposition 4.3.1.**  $C_j$  is the unique object for which  $(C_j, \eta, 1)$  is a pullback on  $\chi_{C_j}$  and  $t$ . Stated differently, the diagram in figure 4.10 above properly defines the class  $C_j$ .

*Proof.* We insert an object  $C'_j \subseteq D$  together with the morphisms  $\eta'$ ,  $\mu$ , and  $1'$  into the diagram.

Now, assume that that  $C'_j \supset C_j$ , i.e. there are elements in  $C'_j$  which are not in  $C_j$ . Assume further that  $\eta'$  is an inclusion morphism that maps every element in  $C'_j$  onto the same element in  $D$ . Then the diagram clearly does not commute since there are necessarily elements in  $C'_j$  for which  $t \circ 1' \neq \chi_{C_j} \circ \eta'$ . If we on the other hand assume  $C'_j \subseteq C_j$ , i.e. that  $C'_j$  is a subset (up to equality) of  $C_j$ , then there is clearly a mediating map  $\mu$  such that  $\eta' = \eta \circ \mu$  by successive inclusion. Hence,  $C_j$  is precisely the object that induces a pullback on  $\chi_{C_j}$

and  $t$  (see figure 4.13). □

The mapping  $\mathbf{1} \rightarrow \mathbb{T}$  is called a *subobject classifier* since the inclusion map from the class  $C_j$  onto the document collection  $D$  defines a classification on  $D$  (Mac Lane, 1998, p. 105).



**Figure 4.13.** The class  $C_j$  induces a pullback on  $\chi_{C_j}$  and  $t$ .

### 4.3.5 The space of classifiers on $D$

Sebastiani (2005) defines the classification of a set of documents  $D$  as a function

$$\varphi : D \times C \rightarrow \{T, F\} \tag{4.3}$$

where  $C$  is a set of classes and  $\{T, F\}$  is the set of truth values (*true* and *false*), which is equivalent to the truth space  $\mathbb{T}$ . For convenience in the formalization that follows, we will define  $\varphi$  in an essentially equivalent fashion, but switching the order of the sets in the Cartesian

product in the domain:

$$\varphi : C \times D \rightarrow \mathbb{T} \quad (4.4)$$

Using these definitions we proceed to define a set of *binary* classifiers  $\{\varphi_i : D \rightarrow \mathbb{T}\}_i$  (cf. the formalization in Sebastiani, 2005), each classifier  $\varphi_i$  having the definition

$$\varphi_i(d_j) := \varphi(c_i, d_j)$$

In other words, each classifier  $\varphi_i$  is defined by *currying*  $\varphi$ , i.e. mapping  $\varphi$  onto a function that takes only one argument (Pierce, 1991, p. 33). We now consider the space of *all* possible binary classifiers  $\varphi_i$ , which is equivalently the space of all maps  $D \rightarrow \mathbb{T}$ . Following the notational form in (Pierce, 1991, p. 34) we denote this space  $\mathbb{T}^D$ , the *exponential object* of  $\mathbb{T}$  and  $D$ . The exponential object is defined as the set of all possible maps  $D \rightarrow \mathbb{T}$  (which should not be confused with a hom-set, which is a set of arrows contained in a specific diagram). Then to each classifier  $\varphi$  there is a commuting category diagram with the following structure (figure 4.14):

$$\begin{array}{ccc}
 \mathbb{T}^D \times D & \xrightarrow{\text{eval}} & \mathbb{T} \\
 \uparrow \text{curry}(\varphi) \times \text{id}_D & & \nearrow \varphi \\
 C \times D & & 
 \end{array}$$

**Figure 4.14.** To every classifier  $\varphi : C \times D \rightarrow \mathbb{T}$  there exists a function space  $\mathbb{T}^D$  such that the diagram above commutes.

In figure 4.14  $\text{curry}(\varphi)$  denotes a map from  $C$  to the function space  $D \rightarrow \mathbb{T}$ , which makes explicit the (trivial) fact that the currying of a classifier  $\varphi$  into a class-specific binary classifier  $\varphi_i$  is determined by a specific class  $c_i$  in  $C$ . Further, we introduce a morphism  $\text{eval}$  that *applies* (evaluates) a certain classifier  $\varphi_i$  onto a document set  $D$ , thereby yielding an output in the truth space  $\mathbb{T}$  for each document in  $D$ .

**Proposition 4.3.2.**  $|\mathbb{T}^D|$  (the number of binary classifiers in  $\mathbb{T}^D$ ) is  $2^N$ , where  $N = |D|$  (cf. Vapnik, 1998, p. 109)

*Proof.* Since each document in  $D$  can be assigned one of two different “decisions” (0 or 1) it follows that the number of different combinations of decisions over  $D$  is  $2 \times 2 \times \cdots \times 2$  (2 multiplied with itself  $N$  times), which is equivalent to  $2^N$ .  $\square$

#### 4.4 The algebraic structure of subject spaces

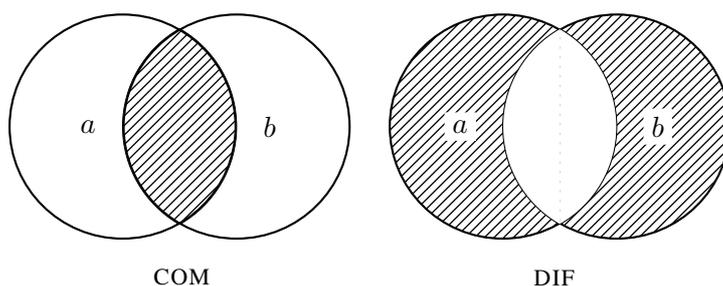
In section 4.2.2 we stipulated that every classification vocabulary is associated with a subject space. Ranganathan (1989, p. 35) refines this idea by claiming that there is a one-to-one relationship between each classification code and the corresponding subject name. In this section we will attempt to analyze the nature and the structure of subject spaces in more precise terms. Hjørland (1992) argues that the *subject* of a document is the “epistemological potential” of the document, i.e. the set of potential information needs to which the document provides an answer (also called by the same author the *aboutness* of the document (Hjørland, 2001)). This definition is, however, rather abstract and does not involve the aspects of a document that are identified during subject analysis. Hjørland (1992) also admits that the subject of a document, as the outcome of a subject analysis, is a quality displayed

by the document from the perspective of a certain discipline.

In the treatment that follows we assume that the subject of a document can also be regarded as a *concept* in the sense of a set of objects (in this case: a set of *ideas*) having certain properties (see e.g. Stock, 2010). Schäuble (1987) investigates the semantic relations of concepts in a thesaurus, based on their role as constituents of a formal language. Given two concepts  $\tilde{a}$  and  $\tilde{b}$  the author defines, somewhat analogously to the set operations union and symmetrical set difference, the following operations:

1. What is *common* between  $\tilde{a}$  and  $\tilde{b}$ ? We denote this operation  $\text{COM}(\tilde{a}, \tilde{b})$ .
2. What is *different* between  $\tilde{a}$  and  $\tilde{b}$ ? We denote this operation  $\text{DIF}(\tilde{a}, \tilde{b})$ .

These two relations can be depicted as *intersection* and *symmetric set difference* respectively (figure 4.15):



**Figure 4.15.** Concept relations visualized as intersection and symmetric set difference.

Schäuble also introduces two constant concepts, the *empty* concept  $\varepsilon$  and the *universal* concept  $\Omega$ , the latter representing the meaning of all terms in the thesaurus taken together. The algebraic relations Schäuble identifies for COM and DIF are listed below.

1.  $\text{DIF}(\tilde{a}, \varepsilon) = \tilde{a}$
2.  $\text{COM}(\tilde{a}, \Omega) = \tilde{a}$
3.  $\text{DIF}(\tilde{a}, \tilde{a}) = \varepsilon$
4.  $\text{COM}(\tilde{a}, \tilde{a}) = \tilde{a}$
5.  $\text{DIF}(\tilde{a}, \tilde{b}) = \text{DIF}(\tilde{b}, \tilde{a})$
6.  $\text{DIF}(\tilde{a}, \text{DIF}(\tilde{b}, \tilde{c})) = \text{DIF}(\text{DIF}(\tilde{a}, \tilde{b}), \tilde{c})$
7.  $\text{COM}(\tilde{a}, \text{COM}(\tilde{b}, \tilde{c})) = \text{COM}(\text{COM}(\tilde{a}, \tilde{b}), \tilde{c})$
8.  $\text{COM}(\tilde{a}, \text{DIF}(\tilde{b}, \tilde{c})) = \text{DIF}(\text{COM}(\tilde{a}, \tilde{b}), \text{COM}(\tilde{a}, \tilde{c}))$
9.  $\text{COM}(\text{DIF}(\tilde{a}, \tilde{b}), \tilde{c}) = \text{DIF}(\text{COM}(\tilde{a}, \tilde{c}), \text{COM}(\tilde{b}, \tilde{c}))$

In addition to these operations defined by Schäuble we propose the following relation as a reasonable extension of operation 3) above:

$$10. \text{DIF}(\tilde{a}, \tilde{b}) = \varepsilon \iff \tilde{a} = \tilde{b}$$

Schäuble demonstrates that an algebra based on COM and DIF, with the properties given above, satisfies the conditions for a *Boolean ring* (for a definition see e.g. Deskins, 1995, p. 25), and that the ring  $\mathfrak{B} = (X, \text{DIF}, \text{COM}, \varepsilon, \Omega)$  is isomorphic to a ring  $(X, \ominus, \cap, \emptyset, \mathcal{P}(X))$ , where  $X$  is any set and  $\ominus$  denotes the symmetric set difference. Two other interesting observations can be made in connection to this.

1. Since a concept algebra defined as above is isomorphic to a set algebra it is also a reasonable conclusion that concepts can be formalized *as sets*.
2. The concept ring  $\mathfrak{B}$  is isomorphic to a ring  $\mathfrak{L} = (P, \wedge, \vee, 0, 1)$  defined on a *Lindenbaum algebra*  $(P, \wedge, \vee, \neg, 0, 1)$ , where  $P$

consists of equivalence classes of formulas over a formal language, and  $x \vee y :\iff (x \wedge \neg y) \vee (\neg x \wedge y)$ . For a definition of a Lindenbaum algebra see e.g. Hinman (2005, p. 76). The possibility to generate a structural equivalence between a system of formulas over a classification vocabulary and a system of classes (sets of objects) will be further explored in section 4.4.1.

To see the applications of a concept algebra, as outlined above, let us consider the arrangement of classes in a strictly hierarchical classification schedule  $\mathcal{S}$ . In such a schedule we could expect to find any of the following relations for any pair of classification codes  $a, b \in \mathcal{S}$ :

1.  $a$  is *subordinate* to  $b$ . We denote this relation  $a \trianglelefteq b$ .
2.  $a$  is *superordinate* to  $b$ . We denote this relation  $a \trianglerighteq b$ .
3.  $a$  is *properly subordinate* to  $b$ . We denote this relation  $a \triangleleft b$ .
4.  $a$  is *properly superordinate* to  $b$ . We denote this relation  $a \triangleright b$ .
5.  $a$  is *equal* to  $b$ . We denote this relation  $a = b$ .
6. None of the relations 1 – 5 apply to the pair  $a, b$ .

The difference between being *subordinated* and being *properly subordinated* is essentially the same as between the set relations *subset* and *proper subset*. If  $a \trianglelefteq b$  then it is logically possible that also  $a = b$ . However,  $a \triangleleft b$  is incompatible with  $a = b$ .

We now intend to define the relations 1 – 5 above using the concept algebra of Schäuble (1987). Let  $a$  and  $b$  be classification codes,  $\tilde{a}$  the concept denoted by  $a$ , and  $\tilde{b}$  the concept denoted by  $b$ .

1.  $a$  is subordinate to  $b$  iff  $\tilde{b}$  is the concept common to  $\tilde{a}$  and  $\tilde{b}$ .

2.  $a$  is properly subordinate to  $b$  iff  $a$  is subordinate to  $b$ , but  $a$  is not equal to  $b$ .

Using the relations introduced by Schäuble we are ready to give the following definitions:

$$\begin{aligned} a \trianglelefteq b & \text{ iff } \text{COM}(\tilde{a}, \tilde{b}) = \tilde{b} \\ a \triangleleft b & \text{ iff } a \trianglelefteq b \text{ and } a \neq b \end{aligned}$$

**Proposition 4.4.1.** *Let  $a$ ,  $b$ , and  $c$  be classification codes in a hierarchical classification system  $\mathcal{S}$ . Assume that  $a \trianglelefteq b$  and  $b \trianglelefteq c$ . Then it holds that  $a \trianglelefteq c$ .*

*Proof.* Since  $b \trianglelefteq c$  it follows by definition that  $\text{COM}(\tilde{b}, \tilde{c}) = \tilde{c}$ . From the distributive property of the operation  $\text{COM}$  it directly follows that

$$\begin{aligned} \text{COM}(\tilde{a}, \tilde{c}) &= \text{COM}(\tilde{a}, \text{COM}(\tilde{b}, \tilde{c})) \\ &= \text{COM}(\text{COM}(\tilde{a}, \tilde{b}), \tilde{c}) \\ &= \text{COM}(\tilde{b}, \tilde{c}) \\ &= \tilde{c} \end{aligned}$$

Since  $\text{COM}(\tilde{a}, \tilde{c}) = \tilde{c}$  it also holds that  $a \trianglelefteq c$ . □

**Lemma 4.4.1.** *Assume that  $a \neq \varepsilon$  and  $b \neq \varepsilon$ . Then if  $\text{COM}(\tilde{a}, \tilde{b}) = \varepsilon$  then  $\text{DIF}(\tilde{a}, \tilde{b}) \neq \varepsilon$ .*

*Proof.* Assume that  $\text{COM}(\tilde{a}, \tilde{b}) = \varepsilon$  and  $\text{DIF}(\tilde{a}, \tilde{b}) = \varepsilon$ . If  $\text{DIF}(\tilde{a}, \tilde{b}) = \varepsilon$  then  $\tilde{a} = \tilde{b}$ . Since  $\text{COM}(\tilde{a}, \tilde{a}) = \tilde{a}$  and  $\tilde{a} \neq \varepsilon$  it cannot hold that both  $\text{COM}(\tilde{a}, \tilde{b}) = \varepsilon$  and  $\text{DIF}(\tilde{a}, \tilde{b}) = \varepsilon$ . Therefore, if  $\text{COM}(\tilde{a}, \tilde{b}) = \varepsilon$  then  $\text{DIF}(\tilde{a}, \tilde{b}) \neq \varepsilon$ . □

**Proposition 4.4.2.** *Let  $a$ ,  $b$ , and  $c$  be classification codes in a hierarchical classification system  $\mathcal{S}$ . Assume that  $a \triangleleft b$  and  $b \triangleleft c$ . Then it also holds that  $a \triangleleft c$ .*

*Proof.* Since by the assumptions  $a \triangleleft b$  it must also hold that  $a \trianglelefteq b$ , since by the definition of  $a \triangleleft b$  it follows by *modus tollens* that if not  $a \trianglelefteq b$  then it cannot hold that  $a \triangleleft b$ . Conversely, by the assumption  $b \triangleleft c$  it must also hold that  $b \trianglelefteq c$ . Therefore, if  $a \triangleleft b$  and  $b \triangleleft c$  it follows that  $a \trianglelefteq c$ . Since  $a \triangleleft c$  iff  $a \trianglelefteq c$  and  $a \neq c$  it remains to be shown that  $a \neq c$  under the stated assumptions. According to the axioms  $a = c$  iff  $\text{DIF}(\tilde{a}, \tilde{c}) = \varepsilon$  and consequently it must hold that  $a \neq c$  iff  $\text{DIF}(\tilde{a}, \tilde{c}) \neq \varepsilon$ . We now observe the following.

$$\begin{aligned}
\text{COM}(\text{DIF}(\tilde{a}, \tilde{b}), \text{DIF}(\tilde{b}, \tilde{c})) &= \text{DIF}(\text{COM}(\text{DIF}(\tilde{a}, \tilde{b}), \tilde{b}), \text{COM}(\text{DIF}(\tilde{a}, \tilde{b}), \tilde{c})) \\
&= \text{DIF}(\varepsilon, \text{DIF}(\text{COM}(\tilde{a}, \tilde{c}), \text{COM}(\tilde{b}, \tilde{c}))) \\
&= \text{DIF}(\varepsilon, \text{DIF}(\tilde{c}, \tilde{c})) \\
&= \text{DIF}(\varepsilon, \varepsilon) \\
&= \varepsilon
\end{aligned}$$

From the lemma (4.4.1) it directly follows that since

$$\text{COM}(\text{DIF}(\tilde{a}, \tilde{b}), \text{DIF}(\tilde{b}, \tilde{c})) = \varepsilon$$

it also holds that

$$\text{DIF}(\tilde{a}, \tilde{c}) = \text{DIF}(\text{DIF}(\tilde{a}, \tilde{b}), \text{DIF}(\tilde{b}, \tilde{c})) \neq \varepsilon.$$

Therefore,  $a \neq c$ . Now, since  $a \trianglelefteq c$  and  $a \neq c$  we conclude that  $a \triangleleft c$ .  $\square$

A final remark: since  $a \trianglelefteq b$  and  $b \trianglelefteq c$  entails  $a \trianglelefteq c$ , as well as  $a \triangleleft b$  and  $b \triangleleft c$  entails  $a \triangleleft c$ , both  $\trianglelefteq$  and  $\triangleleft$  are *transitive* relations.

#### 4.4.1 Semantics and syntax: a model-theoretic perspective

In section 3.3 it was observed that a documentary language (DL), such as a classification schedule, can not function as its own metalanguage. As a consequence, it is possible to make statements about the subject

of a document in a DL, but it is not possible *within* the DL to evaluate whether or not a sentence formulated in the DL is true. According to Alfred Tarski (1956, p. 155) the truth value of a sentence  $s$  in a formal language  $L_t$  has to be defined in a language distinct from  $L_t$ ; a *metalanguage*  $L_m$ . Tarski's condition is succinctly summarized in the following rule. Let  $s$  be a sentence in a metalanguage and  $S$  a sentence expressing  $s$  in an *object language* (for instance English). Then the following holds:

$$\text{"}S\text{" is true iff } s \quad (4.5)$$

*Model theory* is a branch of mathematical logic dealing with the systematic study of first-order sentences and the precise conditions under which a sentence is *true* (Marker, 2002). By generating a model we will have access to a structure of sentences in a metalanguage (first-order logic) by which we can decide whether a proposition in an object language is true. Let  $V$  be a set of symbols (a *vocabulary*),  $D$  a *domain* of objects and  $\pi : V \rightarrow \mathcal{P}(D^n)$  an *interpretation function*. The system  $\mathcal{M} = (D, \pi)$  is called a *model* of  $V$  (Blackburn & Bos, 2005, p. 4). A sentence  $s$  is *true* with respect to  $\mathcal{M}$  if it is valid (syntactically correct) and *provable* in  $\mathcal{M}$ . In concise notation:  $s$  is true in  $\mathcal{M}$  iff  $\mathcal{M} \models s$ .

As an example, let  $V = \{\text{EVEN, ODD, PRIME, LARGER\_THAN}\}$  and  $D = \{1, 2, 3, 4, 5\}$ . Then  $\pi$  reasonably has the following definition:

$$\begin{aligned} \pi(\text{EVEN}) &= \{2, 4\} \\ \pi(\text{ODD}) &= \{1, 3, 5\} \\ \pi(\text{PRIME}) &= \{2, 3, 5\} \\ \pi(\text{LARGER\_THAN}) &= \\ &\{(2, 1), (3, 2), (3, 1), (4, 1), (4, 2), (4, 3), (5, 1), (5, 2), (5, 3), (5, 4)\} \end{aligned}$$

In this model we can see that the sentence "there exists a number such that it is prime and odd" is true. This is written, using the notation of predicate logic:

$$\mathcal{M} \models (\exists x : \text{PRIME}(x) \wedge \text{ODD}(x))$$

On the other hand, the sentence "all prime numbers are odd" is false in  $\mathcal{M}$ , since 2 is prime, but even. This is written:

$$\mathcal{M} \not\models (\forall x : \text{PRIME}(x) \rightarrow \text{ODD}(x))$$

In fact, the sentence "7 is an odd number" is false in  $\mathcal{M}$  since it is not provable in the model.

Turning to subject classification in a library context, we noted in section 3.1.3 that this procedure typically involves the assignment of classification codes to documents. In this situation there is actually an interplay between three categories of objects: the classification codes, the subjects denoted by these codes, and the classes of documents induced by the assignment of classification codes. This procedure can be expressed in model-theoretic terms by generating a model  $\mathcal{M}$  from the components of a classifier  $\varphi : D \times \Omega \rightarrow \{T, F\}$ . Let the classification schedule  $\Omega$  be the *vocabulary* of the model, the document set  $D$  the domain of the model and  $\pi : \Omega \rightarrow \mathcal{P}(D)$  the interpretation function. We define the *class*  $C_\omega$ , over the class predicate  $\omega$ , as the set

$$C_\omega := \pi(\omega) \tag{4.6}$$

Expressed in words,  $C_\omega$  consists of all documents that are elements of the interpretation (which is a set) of  $\omega$ . From a documentary language perspective  $C_\omega$  is tantamount to the *semantic extension* of  $\omega$ . To be able to appreciate the versatility of this model-theoretic approach we

expand the collection of classes to involve classes defined in terms of *well-formed formulas* on  $\Omega$ . The system  $\Gamma_{\mathcal{M}}$  of well-formed formulas over  $\mathcal{M}$  is defined iteratively as follows (see Blackburn & Bos, 2005, p. 7):

1. For every class predicate  $\omega$  and every document term  $d$ , the statement  $\omega(d)$  is a well-formed (atomic) formula.
2. If  $\phi$  and  $\psi$  are well-formed formulas then also  $\neg\phi$ ,  $(\phi \wedge \psi)$ ,  $(\phi \vee \psi)$ , and  $(\phi \rightarrow \psi)$  are well-formed formulas.
3. If  $\phi$  is a well-formed formula and  $x$  is a variable over  $D$ , then also  $\exists x(\phi)$  and  $\forall x(\phi)$  are well-formed formulas.
4. Nothing else is a well-formed formula.

For every well-formed formula  $\phi_d$  in  $\Gamma_{\mathcal{M}}$  over a document term  $d$  we define the class  $C_{\phi_d}$  as

$$C_{\phi_d} = \{d \in D : \mathcal{M} \models \phi_d\} \quad (4.7)$$

In other words, a document is an element of the class  $C_{\phi_d}$  iff  $\phi_d$  is a *semantic consequence* of (or *semantically true* in)  $\mathcal{M}$ .

As an example of how such a system of classes may appear, consider the class predicates HISTORY and MEDICINE. Applied to a document term  $d$  the statement HISTORY( $d$ ) expresses the notion that  $d$  is about history. Similarly, the statement MEDICINE( $d$ ) expresses the notion that  $d$  is about medicine. From these atomary formulas we can proceed to formulate well-formed formulas such as

$\neg$ MEDICINE( $d$ )	:	the document is <i>not</i> about medicine
HISTORY( $d$ ) $\wedge$ MEDICINE( $d$ )	:	the document is <i>both</i> about history and medicine
$\exists x$ (MEDICINE( $x$ ))	:	there exists a document about medicine

As will be shown in section 4.7.5, the set  $\Gamma_{\mathcal{M}}$  of well-formed formulas over  $\mathcal{M}$  is homomorphic to a *classification topology* on  $D$ . In other words, the structure imposed on the document collection by classification is strictly dependent on the logical structure of sentences in the documentary language.

In the framework developed above we have defined classes in terms of formulas based on specific *interpretations*, i.e. specific classifications on  $D$ . Generally speaking there will be three categories of sentences in  $\Gamma_{\mathcal{M}}$ :

- sentences that are *unsatisfiable*, i.e. sentences that are false under *all* interpretations of the classification codes,
- sentences that are *satisfiable*, i.e. sentences that are true under at least *one* interpretation of the classification codes,
- sentences that are *logically valid*, i.e. sentences that are true under *all* interpretations of the the classification codes.

Since the unsatisfiable and the logically valid sentences will yield classes on  $D$  (the empty set and  $D$  respectively) in a way that is independent of any particular interpretation of  $\Omega$ , these types of sentences are not particularly interesting from a classificatory perspective.

#### 4.4.2 First-order languages and binary classifiers

In the empirical part of this thesis we are utilizing a machine learning algorithm that in its basic formulation is a *binary* classifier, i.e. for any object it will yield one of two possible class labels (for instance in the range  $\{0, 1\}$ ). With regard to the discussion in the previous section it is a pertinent question whether it is possible construct a combination of binary classifiers to handle the classification task entailed in an arbitrary sentence  $S$  from a first-order classificatory language.

Since each single binary classifier can be trained to (with some degree of error) return the value 1 for a positive (true) example of a certain class and 0 for a negative (false) example of the same class the following procedure is generally applicable, which is inspired by the formal treatment of the Boolean information retrieval model in (Baeza-Yates & Ribeiro-Neto, 2011, p. 64-66)

1. Every logical formula can be converted to a *disjunctive normal form* (DNF), which is a disjunction of conjuncts. For instance,

$$S = (a \vee b) \wedge \neg c \quad (4.8)$$

can be converted to the equivalent formula

$$S_{\text{DNF}} = (a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge \neg c) \quad (4.9)$$

2. We convert the conjuncts in the DNF to binary vectors according to a tuple of atomary class predicates. In the sentence  $S$  in (4.8) the corresponding tuple of class predicates is  $(a, b, c)$ . The sentence  $S_{\text{DNF}}$  in (4.9) thereby obtains the binary vector

$$\vec{S}_{\text{DNF}} = (1, 1, 0) \vee (1, 0, 0) \vee (0, 1, 0) \quad (4.10)$$

3. For each atomary class predicate  $\phi_i$  in the sentence  $S$  we train a binary classifier  $\psi_i$  in such way that  $\psi_i$  should yield 1 if the object belongs to the corresponding class  $C_{\phi_i}$ , and 0 otherwise.
4. A document is assigned to the class  $C_S$  if the vector of outputs from the binary classifiers equals one of the vectors in  $\vec{S}_{\text{DNF}}$ . For instance, if the outputs for a certain document  $d$  is  $\psi_a = 0$ ,  $\psi_b = 1$ , and  $\psi_c = 0$ , i.e. the binary vector of outputs is  $(0, 1, 0)$ , then  $d$  will be assigned to  $C_S$  since the binary output vector is

equal to the third vector in (4.10).

## 4.5 Order theory

As we have observed above, sets are fundamental mathematical entities for representing collections of objects. Their use is, however, limited because of their inherent lack of structure. In contexts where information is managed and structured the concept of *order* is central. For instance, books on a library bookshelf are ordered alphabetically within each subject class. The participants in a sprint race are ordered according to the time needed to complete the race. In the mathematical area of *order theory* the concept of ordering on sets is the object of study, where concrete relations like "less than" and "precedes" are investigated and characterized. In this framework each ordering of any complexity is broken down into binary relations and grouped together with other orders having the same fundamental properties. An overview of the theory of ordered sets can be found in e.g. Roman (2008).

### 4.5.1 Basic terminology of order theory

Let  $X$  be a set. A binary relation  $\preceq$  is a *weak preorder* on  $X$  if the following conditions hold (Roman, 2008, p. 4):

1.  $a \preceq a$  for all  $a \in X$  [reflexivity];
2.  $a \preceq b$  and  $b \preceq c$  implies  $a \preceq c$  for all  $a, b, c \in X$  [transitivity].

Conversely, a binary relation  $\prec$  is a *strict preorder* on  $X$  if

1.  $a \prec a$  does not hold for any  $a \in X$  [irreflexivity];
2.  $a \prec b$  and  $b \preceq c$  implies  $a \prec c$  for all  $a, b, c \in X$  [transitivity].

If a preorder  $\preceq$  also satisfies the *antisymmetry* condition, i.e.

$$a \preceq b \text{ and } b \preceq a \text{ iff } a = b \text{ for all } a, b \in X$$

then  $\preceq$  is called a (weak or strict, depending on whether the relation is reflexive or irreflexive) *partial order* (Roman, 2008, p. 2). A set  $X$  together with a partial order  $\preceq$  on  $X$  is called a *partially ordered set*, or *poset* for short. If every pair of elements in  $X$  can be "compared" using  $\preceq$ , i.e. for every pair  $a, b \in X$  it holds that  $a \preceq b$  or  $b \preceq a$ , then  $\preceq$  is called a *total order* or *chain*.

Let  $\mathcal{P} = (X, \preceq)$  be a partial order. The *least* element of a subset  $Y \subseteq X$  is an element  $x \in Y$  such that  $x \preceq y$  for all  $y \in Y$ . Conversely, the *greatest* element of  $Y$  is an element  $x$  such that  $y \preceq x$  for all  $y \in Y$ . Further, the *infimum* of  $Y$ , denoted  $\inf(Y)$ , is defined as the *greatest lower bound* of  $Y$ , i.e. the greatest element  $x \in X$  such that for all  $y \in Y$  it holds that  $x \preceq y$ . Finally, the *supremum* of  $Y$ , denoted  $\sup(Y)$ , is defined as the *least upper bound* of  $Y$ , i.e. the least element  $x \in X$  such that for all  $y \in Y$  it holds that  $y \preceq x$ .

As an example, the power set  $\mathcal{P}(X)$  of a non-empty set  $X$  together with the binary relation  $\subset$  (subset) form a strict partial order since the elements of  $\mathcal{P}(X)$  can be partially ordered by inclusion. Let  $X$  be the set  $\{1, 2, 3\}$ . The elements of  $\mathcal{P}(X)$  can then be listed in the following sequences, using  $\subseteq$  as the ordering relation.

$$\begin{aligned} \emptyset &\subset \{1\} \subset \{1, 2\} \subset \{1, 2, 3\} \\ \emptyset &\subset \{1\} \subset \{1, 3\} \subset \{1, 2, 3\} \\ \emptyset &\subset \{2\} \subset \{1, 2\} \subset \{1, 2, 3\} \\ \emptyset &\subset \{2\} \subset \{2, 3\} \subset \{1, 2, 3\} \\ \emptyset &\subset \{3\} \subset \{1, 3\} \subset \{1, 2, 3\} \\ \emptyset &\subset \{3\} \subset \{2, 3\} \subset \{1, 2, 3\} \end{aligned}$$

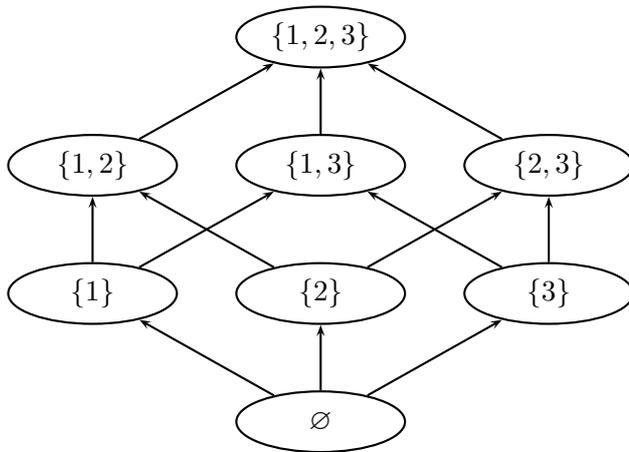
As we can see, the strict partial order defined by  $\subset$  results in six chains for this particular set. Let  $Y = \{\{1, 2\}, \{1, 3\}\}$ . Then  $\inf(Y) = \{1\}$  and  $\sup(Y) = \{1, 2, 3\}$ .

### 4.5.2 Hasse diagram

Let  $\mathcal{P} = (X, <)$  be a strict poset. The *covering relation*  $\tilde{\mathcal{P}} \subset \mathcal{P}$  is a binary relation such that for each pair  $x, y \in X$  it holds that  $(x, y) \in \tilde{\mathcal{P}}$  iff  $x < y$  and there is no element  $z \in X$  such that  $x < z < y$  (Roman, 2008, p. 4). In other words,  $\tilde{\mathcal{P}}$  is the smallest (in the sense of cardinality) binary relation on  $X$  such that it is possible to "restore"  $\mathcal{P}$  from  $\tilde{\mathcal{P}}$ . For instance, the covering relation of the set  $X = \{1, 2, 3, 4\}$  ordered by the numeric order  $<$  is  $\{(1, 2), (2, 3), (3, 4)\}$ . To restore  $\mathcal{P}$  from  $\tilde{\mathcal{P}}$  the following generative rules are exhaustively applied.

1.  $\tilde{\mathcal{P}} \subset \mathcal{P}$ .
2. If  $(x, y) \in \mathcal{P}$  and  $(y, z) \in \mathcal{P}$  then also  $(x, z) \in \mathcal{P}$ .

A partial order  $\mathcal{P}$  with a reasonably limited number of elements can be visualized in a diagram structure called *Hasse diagram* (Roman, 2008, p. 4). The Hasse diagram is formally a visualization of a *directed graph* (see section 4.6.1) in which each edge represents an element of the *covering relation*  $\tilde{\mathcal{P}}$  of  $\mathcal{P}$ . In fact, the underlying graph *equals*  $\tilde{\mathcal{P}}$ . We will below use the expression *Hasse graph* to denote the graph corresponding to the covering relation of a partial order (in contexts where the visualization of the relation is of little or no relevance) and *Hasse tree* if the covering relation is a tree (see section 4.6.1). A Hasse diagram for  $(\mathcal{P}(X), \subset)$ , with  $X = \{1, 2, 3\}$  is depicted in figure 4.16.



**Figure 4.16.** A Hasse diagram of the power set of  $\{1, 2, 3\}$  ordered by the relation  $\subset$ .

### 4.5.3 Lattice

The structure visualized in figure 4.16 above is in fact an example of a subtype of posets called *lattice*. Let  $\mathcal{P} = (X, \prec)$  be a poset. We define two operations on  $\mathcal{P}$ , *meet* (denoted by  $\wedge$ ) and *join* (denoted by  $\vee$ ), as follows. Let  $a, b$  be any two elements in  $\mathcal{P}$ . Then (Roman, 2008, p. 7)

$$a \wedge b = \inf\{a, b\}, \text{ called the } \textit{meet} \text{ of } a \text{ and } b$$

$$a \vee b = \sup\{a, b\}, \text{ called the } \textit{join} \text{ of } a \text{ and } b$$

If every pair of elements in  $\mathcal{P}$  has a meet and a join then  $\mathcal{P}$  is called a *lattice* (Roman, 2008, p. 53).

#### 4.5.4 Order theory and classification

In a physical library documents are arranged by two organization principles working in tandem: by subject category and in alphabetical order. The subject categories are typically arranged in a hierarchical order.

**Lemma 4.5.1.** *Let  $\mathcal{S} = (\mathcal{C}, \trianglelefteq)$  be a hierarchical classification schedule. The relation  $\trianglelefteq$  forms a partial order on  $\mathcal{S}$ .*

*Proof.* We have already shown that  $\trianglelefteq$  is *transitive*. By definition we have that  $a \trianglelefteq b$  iff  $\text{COM}(\tilde{a}, \tilde{b}) = \tilde{b}$ . Since  $\text{COM}(\tilde{a}, \tilde{a}) = \tilde{a}$  it follows that  $a \trianglelefteq a$ , which means that  $\trianglelefteq$  is also *reflexive*. Further, assume that  $a \trianglelefteq b$  and  $b \trianglelefteq a$ . Then  $\text{COM}(\tilde{a}, \tilde{b}) = \tilde{b}$  and  $\text{COM}(\tilde{b}, \tilde{a}) = \tilde{a}$ . But since  $\text{COM}(\tilde{a}, \tilde{b}) = \text{COM}(\tilde{b}, \tilde{a})$  it immediately follows that  $a = b$ . We have thereby established that  $\trianglelefteq$  is *antisymmetric*. Since  $\trianglelefteq$  is reflexive, antisymmetric and transitive it is also a partial order on  $\mathcal{S}$ .  $\square$

**Proposition 4.5.1.** *Let  $\mathcal{S} = (\mathcal{C}, \trianglelefteq)$  be a hierarchical classification schedule. The relation  $\trianglelefteq$  can always be employed to generate a lattice over  $\mathcal{S}$ .*

*Proof.* To turn  $(\mathcal{C}, \trianglelefteq)$  into a lattice we augment  $\mathcal{C}$  with two class variables,  $\varepsilon$  denoting the empty class, and  $\Omega$  denoting the universal class. For every class  $c_i \in \mathcal{C}$  it then holds that  $\varepsilon \trianglelefteq c_i$  and  $c_i \trianglelefteq \Omega$ . It immediately follows that for every element  $c_i \in \mathcal{C}$  there exists at least one element  $x \in \mathcal{C}$  such that  $x \trianglelefteq c_i$ , namely  $\varepsilon$ . Conversely, for every element  $c_i \in \mathcal{C}$  there exists at least one element  $x \in \mathcal{C}$  such that  $x \trianglerighteq c_i$ , namely  $\Omega$ . Let  $C_i^\sharp \subseteq \mathcal{C}$  be the collection of all elements such that for each  $c_j \in C_i^\sharp$  it holds that  $c_i \trianglelefteq c_j$ . Further, let  $C_i^\flat \subseteq \mathcal{C}$  be the collection of all elements such that for each  $c_j \in C_i^\flat$  it holds that  $c_i \trianglerighteq c_j$ . The set  $C_i^\sharp$  contains at least one element, namely  $\Omega$ . The set  $C_i^\flat$  also contains at least one element, namely  $\varepsilon$ . Now, let  $c_i$  and  $c_j$  be any two

classes in  $\mathcal{C}$  and  $C_i^{\sharp} := C_i^{\#} \cup C_i^{\flat}$ . If  $c_j \in C_i^{\sharp}$  then  $c_i \wedge c_j = c_j$  if  $c_j \triangleleft c_i$ , or else  $c_i \wedge c_j = c_i$  if  $c_i \triangleleft c_j$ . If  $c_j \notin C_i^{\sharp}$  then  $c_i \wedge c_j = \varepsilon$ . Conversely, if  $c_j \in C_i^{\sharp}$  then  $c_i \vee c_j = c_j$  if  $c_j \triangleright c_i$ , or else  $c_i \vee c_j = c_i$  if  $c_i \triangleleft c_j$ . If  $c_j \notin C_i^{\sharp}$  then  $c_i \vee c_j = \Omega$ . We conclude that  $(\mathcal{C} \cup \{\varepsilon, \Omega\}, \trianglelefteq, \wedge, \vee)$  is a lattice.  $\square$

There is an interesting parallel between proposition (4.5.1) and the lattice structure of *concepts* derived in *formal concept analysis* (Wille, 2005). Let  $E$  be a set of *entities* and  $A$  a set of *attributes*. We let the notation  $e \leftarrow a$ , where  $e \in E$  and  $a \in A$ , mean "e has the attribute a". Further, let  $X \subseteq E$  and  $Y \subseteq A$ . The derivation operator  $g$  has the following symmetric definition:

$$\begin{aligned} g(X) &= \{a \in A : \forall e \in X (e \leftarrow a)\} \\ g(Y) &= \{e \in E : \forall a \in Y (e \leftarrow a)\} \end{aligned} \quad (4.11)$$

In other words,  $g(X)$  consists of the subset of all attributes in  $A$  such that every element in  $X$  has every attribute in the subset, and  $g(Y)$  consists of the subset of all elements in  $E$  such that every element in the subset has every attribute in  $Y$ . A *formal concept* is defined as a pair  $(X, Y)$ , where  $X \subseteq E$  and  $Y \subseteq A$ , such that  $g(X) = Y$  and  $g(Y) = X$ . We define a partial relation  $\lesssim$  such that  $(X_1, Y_1) \lesssim (X_2, Y_2)$  means that  $(X_1, Y_1)$  is a *subconcept* of  $(X_2, Y_2)$ . In formal concept analysis this relation has the definition

$$\begin{aligned} (X_1, Y_1) \lesssim (X_2, Y_2) &\text{ iff } X_1 \subseteq X_2 \text{ or equivalently} \\ (X_1, Y_1) \lesssim (X_2, Y_2) &\text{ iff } Y_1 \supseteq Y_2 \end{aligned} \quad (4.12)$$

It is straightforward to show, but outside the scope of this work, that  $\lesssim$  induces a lattice structure.

A special case of the hierarchical order is the *linear order* in which each element (except one) has precisely one immediately preceding el-

ement. This occurs for instance in alphabetic orderings of documents. Such an ordering corresponds to a special subcategory of partial orders, called *total orders*, mentioned in section 4.5.1.

## 4.6 Graph theory

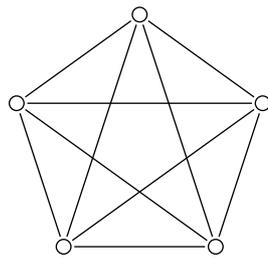
A mathematical structure especially apt for the task of representing a collection of binary (pairwise) relations is called a *graph*. Informally a graph can be conceived as a diagram of nodes with lines connecting the nodes in a certain fashion. *Graph theory* is the mathematical study of graphs and has a certain affinity with a larger mathematical field concerned with the general study of the structure of sets, called *topology* (presented in the next section).

### 4.6.1 Basic concepts of graph theory

The concepts defined in this section can be found in any modern textbook on graph theory, for instance Bondy & Murty (2008). In the terminology of graph theory, the nodes of the graph are called *vertices* (singular: *vertex*) and the lines connecting the nodes are called *edges*. A graph can be *directed*, meaning that the binary relations represented by the graph are not necessarily symmetric and the edges consist of arrows with a certain direction. Conversely, a graph representing binary relations that are consistently symmetric is called *undirected*. Let  $V$  be a set of vertices and let  $v_j, v_k$  be two vertices in  $V$ . Formally, an edge between  $v_j$  and  $v_k$  in a directed graph is the ordered pair  $(v_j, v_k)$  and in an undirected graph the set  $\{v_j, v_k\}$ . The number of edges connected to a vertex  $v_j$  is called the *degree* of  $v_j$ . The set  $V$  together with a set  $E$  of edges, more specifically the ordered pair  $(V, E)$ , is a *graph* over  $V$ . The number of vertices in  $G$  is called the *order* of  $G$ .

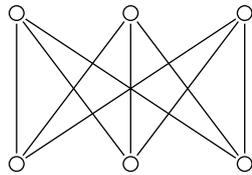
Further, let  $G = (V, E)$  be a graph. A graph  $G' = (V', E')$  is called a *subgraph* of  $G$  (typically written analogously to set notation:  $G' \subseteq G$ ) if  $V' \subseteq V$  and  $E' \subseteq E$ .

A sequence of consecutive edges that together connect two vertices  $v_j$  and  $v_k$  is called a *path* between  $v_j$  and  $v_k$ . A *cycle* is a path from a vertex  $v_j$  leading back to  $v_j$ . The *distance* between two vertices  $v_j, v_k$  in a graph is typically defined as the number of edges on the shortest path connecting  $v_j$  and  $v_k$ . If there is precisely one path between every pair of vertices, the graph is called a *tree*. The tree structure will be of considerable interest to us for the representation of hierarchical configurations of classes. A *complete* graph is a graph such that every pair of vertices in the graph is connected by an edge (or in the terminology of (Bondy & Murty, 2008, p. 4), all vertices are pairwise *adjacent*). A complete graph over  $n$  vertices is typically denoted  $K_n$  (cf. figure 4.17 below).



**Figure 4.17.** The complete graph  $K_5$ .

A *bipartite* graph is a graph  $G = (V, E)$  such that  $V$  can be partitioned into two sets  $X$  and  $Y$  such that every edge in  $G$  connects one vertex in  $X$  and one vertex in  $Y$  (Bondy & Murty, 2008, p. 4). If we let  $m = |X|$  and  $n = |Y|$  a complete bipartite graph is typically denoted  $K_{m,n}$  (cf. figure 4.18).



**Figure 4.18.** The complete bipartite graph  $K_{3,3}$ .

Let  $G$  be a graph and let  $\mathbb{R}^2$  denote the (ordinary) Euclidean plane. If there is a map  $\phi : G \rightarrow \mathbb{R}^2$  such that no edges in the plane cross each other, but only meet at the vertices, the image  $\phi(G)$  is called a *planar embedding* of  $G$ . In general terms, an embedding of  $G$  in  $\mathbb{R}^n$  is the image of a map  $\phi : G \rightarrow \mathbb{R}^n$  such that no edges cross in  $\phi(G)$ .

## 4.6.2 Classification schedules as graphs

We will here consider two arrangements of classification codes – *simple enumerative*, *hierarchical enumerative* – to investigate their properties as graph structures. By “simple enumerative” we refer to an enumerative arrangement of codes with no explicit hierarchy between the codes (only a linear order). We will in the section on topology (section 4.7) use the results obtained here to quantify the dimensionality of the various classification schedule arrangements.

### 4.6.2.1 Simple enumerative schedules



**Figure 4.19.** Linear order in a simple enumerative schedule.

Let  $G_s = (V, E_s)$  be a connected graph with two vertices having degree 1, and the remaining vertices having degree 2. Such a graph is representative of a linear order of objects (see figure 4.19 above), such as an alphabetical ordering of classification codes.

**Proposition 4.6.1.** *There exists a bijective map  $\Phi$  embedding  $G_s$  on the real line  $\mathbb{R}^1$ .*

*Proof.* We will here only give the outline of a strict proof. It is possible to uniquely enumerate (up to equivalence) the vertices in  $G_s$  by beginning with one of the vertices of degree 1 and then traversing through the remaining vertices. Using this enumeration, we can construct a bijective map between the  $|V|$  vertices and  $\mathbb{R}^1$  by mapping the first vertex of the enumeration to 0, the second vertex to 1 and so on. We can similarly construct a bijective map between the edges in  $G_s$  and connected line segments in  $\mathbb{R}^1$ , by mapping the edge between vertex  $j$  and vertex  $j + 1$  onto the interval  $[j - 1, j]$ . The intersection between the line segments in  $\mathbb{R}^1$  are exactly the points associated with the vertices in  $G_s$ . We conclude that there exists a bijective map  $G_s \rightarrow \mathbb{R}^1$ .  $\square$

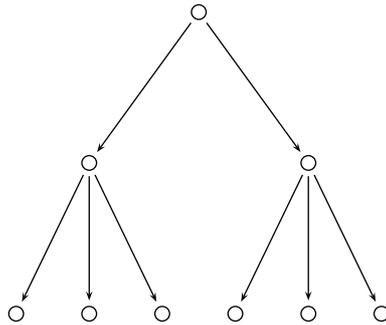
#### 4.6.2.2 Hierarchical enumerative schedules

A hierarchical classification schedule is, as we have discussed in section 4.5.4, determined by a partial order  $\leq$  on a set  $\mathcal{C}$  of classification codes. The relation  $\triangleleft$  is antisymmetric and irreflexive, meaning that given any pair  $c_j, c_k \in \mathcal{C}$  it cannot both hold that  $c_j \triangleleft c_k$  and  $c_k \triangleleft c_j$ . Assuming that for every element in  $\mathcal{C}$ , except for one *root element*, it holds that to every element  $c_k \in \mathcal{C}$  there exists precisely one element  $c_j \in \mathcal{C}$  such that

1.  $c_j \triangleleft c_k$ , and

2. there exists no element  $c_\xi$  such that  $c_j \triangleleft c_\xi \triangleleft c_k$ ,

it is easily verified that the structure of  $\mathcal{C}$  is representable by a *tree*. This follows directly from the fact that each class element has a unique sequence of superordinate class elements up to the root element. A tree has the schematic structure depicted in figure 4.20.



**Figure 4.20.** A schematic graph of a hierarchical enumerative schedule.

We let  $G_h = (V, E_h)$  denote a tree structure of classification codes.

**Proposition 4.6.2.**  $G_h$  is embeddable in  $\mathbb{R}^2$ .

*Proof.* According to *Kuratowski's theorem* (Bondy & Murty, 2008, p. 268) a graph is planar iff it does not contain a subgraph that is isomorphic to a subdivision of  $K_5$  or  $K_{3,3}$ . By (edge) *subdivision* is meant a splitting of existing edges by the insertion of new vertices (see e.g. Bondy & Murty, 2008, p. 55). From the definition of a tree it follows that a tree does not contain any cycles and obviously no subdivision of  $K_5$  or  $K_{3,3}$  is *acyclic*. Therefore, a tree cannot be isomorphic by subdivision to either  $K_5$  or  $K_{3,3}$  and it follows that every tree is a planar graph.  $\square$

## 4.7 Topology

Another mathematical field having the study of structures on sets is the main focus is that of *topology*. On an informal level it can be thought of as an abstraction of geometry where concepts like distance and angle are discarded on behalf of structural properties defined by a collection of subcomponents called *open sets* (Mendelson, 1990, p. 1-2). One of the fundamental properties studied in topology is the problem whether two spaces are equivalent under continuous transformations (so called *homeomorphisms*, see section 4.7.3). As an example, a solid cube and a solid sphere are topologically equivalent because the one shape can be transformed into the other by continuous deformation (twisting and stretching). A joke quite adequately capturing the essence of topology (or more precisely, topologists) is the following (Renteln & Dundes, 2005):

Q: What is a topologist? A: Someone who cannot distinguish between a doughnut and a coffee cup.

The serious fact behind this joke is that a doughnut (i.e. a torus-shaped object) can, theoretically speaking, be transformed into a coffee cup by continuous deformation. In this work make the assumption that topology is a mathematical field with a certain relevance for classification theory, since both fields are concerned with the fundamental *structures* of sets. In order to show how topology applies to classification (both the structure imposed on document collections, as well as structures within classification schedules) we will begin with presenting a few central concepts in topology.

### 4.7.1 Basic definitions in topology

Let  $X$  be a set. A topology  $\tau$  on  $X$  is a set of subsets of  $X$ , such that the empty set  $\emptyset$  and  $X$  are elements in  $\tau$  and following conditions hold for  $\tau$  (see e.g. Munkres, 2000, p. 76)

1. Let  $\mathcal{U}$  be any countable collection of subsets in  $\tau$ . Then  $\bigcup_{A_i \in \mathcal{U}} A_i$  is also an element in  $\tau$ .
2. If  $U \in \tau$  and  $V \in \tau$  then also  $U \cap V \in \tau$ .

This definition of a topology may be surprisingly “abstract” and un-intuitive, but it captures on a basic level the idea that a structure on a set  $X$  should be defined in terms of systems of sets containing each other. The “outermost” set of every topology on  $X$  is  $X$  itself and the “innermost” sets are the atomary building blocks of the structure.

Every element of a topology  $\tau$  is by definition an *open set* in  $\tau$ . Any subset of  $X$  being the *set complement* to an element in  $\tau$  is called *closed*. Let  $U$  be a set in  $\tau$ . The intersection of all closed sets containing  $U$  is called the *closure* of  $U$  (Munkres, 2000, p. 95), which we write  $\text{cl}U$ . Topologies may also contain sets that are simultaneously open and closed. These are commonly (and somewhat awkwardly) called *clopen*. It is easily verified that given any set  $X$  the power set  $\mathcal{P}(X)$  is a topology on  $X$ , and so is the “trivial” topology  $\{\emptyset, X\}$ .

**Example 4.7.1.** *Let  $D$  be a set of documents. Assume that we partition  $D$  into the subsets  $D_f$  of fiction and  $D_n$  of nonfiction.<sup>2</sup> Then the set  $\{\emptyset, D, D_f, D_n\}$  is a topology on  $D$ .*

<sup>2</sup>To keep this example simple we assume that every document is assigned to precisely one of these categories.

### 4.7.2 Basis and subbasis

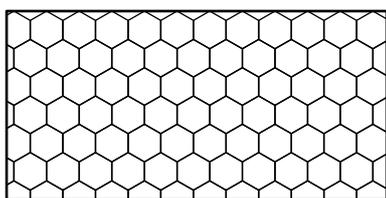
As we can see from the conditions above a topology can be conceived as *generated* from a collection of basic sets by repeated application of the set operations union and intersection. This leads us to the concept of *basis* for a topology. In many cases it is cumbersome or virtually impossible to define an entire topology by explicitly listing its open sets (cf. the similar problem of defining large sets) and in such cases we need a basis to generate a topology. A basis for a topology  $\tau$  defined on a set  $X$  is formally a collection  $\mathcal{B} = \{B_1, B_2, \dots\}$  of subsets (called *basis elements*) of  $X$  such that (p. 78 Munkres, 2000)

1. Every element  $x \in X$  is contained in at least one basis element in  $\mathcal{B}$ .
2. Let  $B_1$  and  $B_2$  be basis elements. If  $x \in X$  is contained in  $B_1 \cap B_2$  then there exists a basis element  $B_3$  such that  $x \in B_3$  and  $B_3 \subseteq B_1 \cap B_2$ .

The conditions above are sufficient to guarantee that the union of all basis elements is  $\tau$  and that every open set in  $\tau$  can be formed by a countable union of basis elements. In other words, the sets in the basis *cover* the topological space (cf. figure 4.21 below). A *subbasis*  $\tilde{\mathcal{B}}$  is a collection of subsets of  $X$  that can be *turned into* a basis by intersecting the elements of  $\tilde{\mathcal{B}}$ .

**Example 4.7.2.** A partition topology is a topology generated by the subsets of a partition of a set. Connecting to the example 4.7.1 above we first notice, given the stated assumptions, that  $\{D_f, D_n\}$  is a partition on  $D$ . It is also a topological basis since

1. every element  $d \in D$  is also in one of  $D_f$  and  $D_n$ , and



**Figure 4.21.** The basis sets can be thought of as tiles fully covering a surface.

2. if  $d \in D_f$  (conversely for  $D_n$ ) then also  $d \in D$ . But it (almost trivially) follows that  $d \in D_f \cap D = D_f$ , which shows that also the second condition for a basis is satisfied.

### 4.7.3 Neighborhood and homeomorphism

Let  $\mathcal{T} = (X, \tau)$  be a topological space and  $p$  a point in  $\mathcal{T}$ . A *neighborhood* to  $p$  is a set  $V \in \tau$  containing an open set  $U$  such that  $p \in U \subseteq V$  (Munkres, 2000, p. 96). It follows that every open set containing  $p$  is also a neighborhood of  $p$ . However, the neighborhood itself is not necessarily open. The collection of neighborhoods around  $p$  is called the *neighborhood system* around  $p$ . The neighborhood system of  $p$  can be intuitively understood as a collection of sets of points "close to"  $p$ . Though this concept is perhaps easier to visualize in a geometric context it is on a general level an important component in the formalization of structures based on (set) containment.

Let  $\mathcal{T}_1 = (X, \tau_1)$  and  $\mathcal{T}_2 = (Y, \tau_2)$  be two topological spaces, and  $f$  a function  $f : X \rightarrow Y$ . Further, let  $p$  be a point in  $\mathcal{T}_1$ . Then  $f$  is said to be *continuous at  $p$*  if, given any neighborhood  $V$  of  $f(p)$ , there exists a neighborhood  $U$  of  $p$  such that  $f(U) \subseteq V$ . The essence of the notion of continuous mappings between topological spaces is

that such mappings preserve the structure of neighborhoods around the points in the preimage and yield structurally equivalent spaces. If a map  $f$  between two topological spaces is continuous "in both directions" it is called a *homeomorphism*, defined formally as follows. With the notation used as before, a function  $f : \mathcal{T}_1 \rightarrow \mathcal{T}_2$  is a *homeomorphism* if (Munkres, 2000, p. 105)

1.  $f$  is *continuous*,
2.  $f$  is bijective,
3. the inverse map  $f^{-1}$  is continuous.

A homeomorphism between two topological spaces can more succinctly be defined as a *structural isomorphism* between two spaces, meaning that such spaces are topologically equivalent. To denote that  $\mathcal{T}_1$  and  $\mathcal{T}_2$  are *homeomorphic* we write  $\mathcal{T}_1 \cong \mathcal{T}_2$ .

**Example 4.7.3.** *A solid sphere and a solid cube are homeomorphic, whereas a solid sphere and a solid torus are not homeomorphic.*

#### 4.7.4 Distinguishability and connectedness

An important property of certain topological spaces which divides them into distinct families is the one of *topological distinguishability*. Loosely speaking, two points in a topological space are distinguishable if it is possible to separate them by a condition on the open sets in the topological space. Any topological space that has the property of distinguishability is called a  $T_0$  space (Willard, 2004, p. 85). A topological space  $\mathcal{T} = (X, \tau)$  is  $T_0$  if for any two elements  $x, y \in X$  there exists an open set  $U \in \tau$  such that  $x \in U$  and  $y \notin U$  or  $x \notin U$  and  $y \in U$ . A common family of  $T_0$  spaces is the  $T_2$  spaces, also called *Hausdorff spaces* (Willard, 2004, p. 86). A topological space

$\mathcal{T} = (X, \tau)$  is  $T_2$  if for each pair of points  $x, y \in X$  there exist two disjoint open sets  $U, V \in \tau$  such that  $x \in U$  and  $y \in V$ , but  $x \notin V$  and  $y \notin U$ . Any two distinct points in a  $T_2$  space therefore have disjoint *neighborhoods*. The oft-encountered Euclidean spaces are examples of  $T_2$  spaces.

A topological space  $\mathcal{T} = (X, \tau)$  is *connected* if it, informally speaking, is not divided into isolated parts. The formal definition is as follows (Munkres, 2000, p. 148).  $\mathcal{T}$  is connected iff there are no sets  $U, V \subseteq X$  such that

1.  $U$  and  $V$  are open, disjoint sets in  $\mathcal{T}$  and
2. the set  $\{U, V\}$  is a partition of  $X$ .

Munkres (2000, p. 148) provides an alternative definition of connectedness that relates in an interesting fashion to the definition of induction dimension (section 4.7.6.2). A topological space  $\mathcal{T} = (X, \tau)$  is *connected* iff the only clopen sets in  $\mathcal{T}$  are  $X$  and  $\emptyset$ .

**Example 4.7.4.** *The topological space of documents introduced in example 4.7.1 is disconnected, since  $D_f$  and  $D_n$  form a partition on  $D$  and are open in the topology.*

## 4.7.5 Subject classification and topology

### 4.7.5.1 Topological properties of document classifications

We are now prepared to go into some detail in characterizing subject classification by means of topological concepts. Subject classification is said to induce a *structure* on a given document collection, but how can we characterize this structure from a mathematical perspective? It is important to point out that there are two related, but still distinct, structures to consider when discussing the topology of classification:

1. the structure imposed on a set of documents as a consequence of classification, and
2. the structure of classification codes in a classification schedule.

We have previously stated that the classification of documents can, given a set  $D$  of documents and a set  $C$  of classification codes, be formalized as a function  $\varphi : D \times C \rightarrow \{T, F\}$ . In the case of a single class  $c_i \in C$  we similarly define a binary classifier  $\varphi_i : D \rightarrow \{T, F\}$ . We have also presented a model-theoretic formalism in which classes of documents are defined by sentences in a formal language by letting the classification schedule be a vocabulary and letting the documents be constants. It is now our endeavor to present how such a class structure generates a topology on  $D$ , but before doing so we will show how the logical operators of the formal language correspond to set-theoretic operations in  $D$ .

**Proposition 4.7.1.** *Let  $L$  be a classificatory language,  $\mathcal{M}$  a model over  $L$ , and  $\{\phi, \psi\}$  two sentences in  $L$ . Further, let  $C_\phi$  and  $C_\psi$  be the classes of documents induced by  $\phi$  and  $\psi$  respectively. Then the following classes are induced by the logical operators (letting  $\rightarrow$  denote class induction):*

$$\begin{aligned} \phi \wedge \psi &\rightarrow C_\phi \cap C_\psi \\ \phi \vee \psi &\rightarrow C_\phi \cup C_\psi \\ \neg\phi &\rightarrow D \setminus C_\phi \text{ (the set complement of } C_\phi\text{)}. \end{aligned}$$

*Proof.* These relations follow immediately from the definitions of the set operations involved. The sentence  $\phi \wedge \psi$  entails the condition that a given document is an element of both  $C_\phi$  and  $C_\psi$ , or in other words the intersection  $C_\phi \cap C_\psi$ . Conversely, the sentence  $\phi \vee \psi$  entails the condition that a given document is an element of (possibly both) of

$C_\phi$  and  $C_\psi$ , which is tantamount to the union  $C_\phi \cup C_\psi$ . Finally,  $\neg\phi$  entails the condition that a document is not an element of  $C_\phi$ , which is tantamount to the set complement of  $C_\phi$ .  $\square$

**Proposition 4.7.2.** *Any classificatory language  $L$  together with a model  $\mathcal{M}$  over  $L$  induces a topology on  $D$ .*

*Proof.* Although this is a general proposition, its proof is remarkably simple. Let  $\tau$  be the collection of classes induced by the sentences in  $L$ . We will show that  $\tau$  contains precisely the required components for a topology, and therefore is a topology.

$\emptyset$  is in  $\tau$  since a subset of the sentences in  $L$  are unsatisfiable (for instance all sentences on the form  $\phi \wedge \neg\phi$ ). Conversely,  $D$  is in  $\tau$  since a subset of the sentences in  $L$  are logically valid (for instance all sentences on the form  $\phi \vee \neg\phi$ ). Let  $\{\phi, \psi\}$  be sentences in  $L$  and  $\{C_\phi, C_\psi\}$  be the classes they induce. Then  $C_\phi \cap C_\psi$  is in  $\tau$  since  $\phi \wedge \psi$  is in  $L$ . Conversely,  $C_\phi \cup C_\psi$  is in  $\tau$  since  $\phi \vee \psi$  is in  $L$ . We have thereby demonstrated that  $\tau$  satisfies the conditions for a topology.  $\square$

In fact, the classes induced from the sentences in  $L$  form the subbasis of a topology on  $D$ . As a consequence it is possible to study classification of a document collection in terms of topological concepts such as containment, coarseness, and various mappings between topologies.

Another property that arises in the class-induced document topology is *measurability*. To any set equipped with a structure called a  $\sigma$ -algebra (sigma-algebra) it is possible to formulate a function called a *measure*, that intuitively speaking, yields values denoting relative *sizes* of various subsets to the original set. An example of a measure is the length of a line segment. To explain how the class-induced topology is measurable we need to first introduce the notion of a  $\sigma$ -algebra (see e.g. Cohn, 2013, p. 2). Let  $X$  be any set. A  $\sigma$ -algebra  $\Sigma$  on  $X$  is a non-empty collection of sets such that if  $A$  is in  $\Sigma$ , then also  $X \setminus A$  is in  $\Sigma$ . Further, the union of any number of sets in  $\Sigma$  is also in  $\Sigma$ . The  $\sigma$ -algebra generated by all the open sets of a topological space  $(X, \tau)$  is called a *Borel  $\sigma$ -algebra* on  $X$  (Cohn, 2013, p. 189). Finally, a set  $X$  together with a  $\sigma$ -algebra on  $X$  is a *measurable space* (Cohn, 2013, p. 8).

**Proposition 4.7.3.** *Every non-empty classification space  $(D, \tau)$  is a measurable space.*

*Proof.* Since  $\tau$  is assumed to be a topology on  $D$  it is also non-empty. From the condition on topologies it follows that  $\tau$  contains all unions of elements in  $\tau$ . Finally, let  $\phi$  be any formula in  $L$  and  $C_\phi$  the induced class. Then also  $\neg\phi$  is in  $L$  and consequently  $D \setminus C_\phi$  is in  $\tau$ . We find that  $\tau$  is a Borel  $\sigma$ -algebra and  $(D, \tau)$  is consequently a measurable space.  $\square$

Any set  $D$  equipped with a  $\sigma$ -algebra  $\Sigma$  on  $D$  together with measure  $\mu$  on  $(D, \Sigma)$  is called a *measure space*. If  $\mu(D) = 1$  then the corresponding measure space is called a *probability space* with  $\mu$  as a

*probability measure* on  $(D, \Sigma)$ . Since the classification space  $(D, \tau)$  is a measurable space we can turn it into a probability space by constructing a suitably defined probability measure, and as a consequence study classifications on  $D$  by the use of probabilities (for instance, the probability of observing documents having certain properties in the topological structure).

#### 4.7.5.2 Topological properties of classification schedules

What we have discussed in analytical terms so far are the topological properties of document classifications induced by a first-order classificatory language. We will now change our focus to hierarchically ordered classification schedules. In what way can we describe such structures using topological concepts? Since hierarchical structures contain partial non-cyclic orders we will approach this problem by using sets defined by order relations.

Let  $A$  be a set and  $\preceq$  a preorder on  $A$ . The *upper set* of an element  $x \in A$  is defined

$$\uparrow x = \{y \in A : x \preceq y\}$$

and the *lower set* of  $x$  is defined

$$\downarrow x = \{y \in A : y \preceq x\}.$$

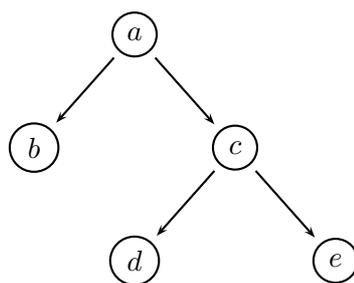
The topological space  $(A, \tau)$  defined using the order relation  $\preceq$ , in which the open sets of  $\tau$  are the *upper sets* of the elements in  $A$ , is an example of an *Alexandrov space* (Arenas, 1999). As a consequence of the definition of  $(A, \tau)$ , the closed sets of  $\tau$  are the *lower sets* of the elements in  $A$ . It is possible to "retrieve" the preorder  $\preceq$  from  $\tau$  by

the defining relation

$$x \preceq y \text{ iff } \text{cl}\{x\} \subseteq \text{cl}\{y\} \quad (4.13)$$

where  $\text{cl}$  denotes *topological closure*. Considering that the preorder uniquely defines an Alexandrov space and the Alexandrov space uniquely specifies a preorder these categories form an interesting duality between an order theoretic and a topological perspective on the structure of a set.

Since both linear orders as well as hierarchical orders can be defined in terms of *partial orders* (which are antisymmetric preorders) the notion of Alexandrov spaces provides an adequate framework for the formalization of the structures of classification schedules in terms of topological spaces. As an example, consider the partial order defined on the set  $C = \{a, b, c, d, e\}$ . The relations displayed in figure



**Figure 4.22.** A Hasse diagram of a tree over 5 classification codes.

4.22 generate the following upper sets on  $C$ :

$$\begin{aligned}\uparrow a &= \{a, b, c, d, e\} \\ \uparrow b &= \{b\} \\ \uparrow c &= \{c, d, e\} \\ \uparrow d &= \{d\} \\ \uparrow e &= \{e\}\end{aligned}$$

Let  $\mathcal{C} = (C, \tau)$  be an Alexandrov space and  $\delta : C \times C \rightarrow \mathbb{Z}_0^+$  a function with the definition

$$\delta(x, y) = |\text{cl}\{x\} \ominus \text{cl}\{y\}| \quad \text{where } \ominus \text{ denotes symmetric set difference.} \quad (4.14)$$

**Proposition 4.7.4.**  $\delta$  is a metric on  $\mathcal{C}$ .

*Proof.* It clearly holds that  $\delta(x, y) \geq 0$  and  $\delta(x, y) = \delta(y, x)$  for all  $x, y \in C$ . Further, assume that there exists a pair  $x, y \in C$  such that  $\delta(x, y) = 0$ . This holds iff  $|\text{cl}(x) \cup \text{cl}(y)| = |\text{cl}(x) \cap \text{cl}(y)|$ . But then it must also be the case that  $\text{cl}(x) = \text{cl}(y)$ , which in turn implies  $x = y$ . A proof showing that the triangle inequality holds for functions on the form  $\delta(A, B) = |A \ominus B|$ , where  $A$  and  $B$  are sets, is found in (Restle, 1959).  $\square$

**Proposition 4.7.5.** Let  $C$  be a set and  $\preceq$  an acyclic partial order on  $C$ . Further, let  $\mathfrak{h} = (C, E)$  be the Hasse tree and  $\mathcal{A} = (C, \tau)$  the Alexandrov space on  $(C, \preceq)$ . We define a metric  $\delta_H$  on  $\mathfrak{h}$  as the geodesic distance between vertices in  $\mathfrak{h}$  and a metric  $\delta_A$  on  $\mathcal{A}$  as in (4.14). Then there exists an isometry  $\phi : \mathfrak{h} \leftrightarrow \mathcal{A}$ .

*Proof.* Let  $r$  be the root vertex of  $\mathfrak{h}$ . From every vertex  $x \in C$  there is a unique path  $P_x = (C_x, E_x)$  from  $x$  to  $r$ . The geodesic distance

between  $x$  and  $r$  is the length of  $P_x$ , which is  $\ell(P_x) = |E_x|$ . Let  $x, y \in C$  be two vertices and  $P_{xy} = (C_{xy}, E_{xy})$  denote the unique path from  $x$  to  $y$ . It generally holds that  $E_{xy} = (E_x \cup E_y) \setminus (E_x \cap E_y) = E_x \ominus E_y$ . Hence,  $\ell(P_{xy}) = |E_x \ominus E_y| = |C_x \ominus C_y|$ . Since  $\text{cl}\{x\} = \downarrow x = \{y \in C : y \preceq x\}$  it is easily verified that  $C_x = \text{cl}\{x\}$  for all  $x \in C$ . We conclude that  $|C_x \ominus C_y| = |\text{cl}\{x\} \ominus \text{cl}\{y\}|$  and therefore  $\delta_H(x, y) = \delta_A(x, y)$  for all  $x, y \in C$ .  $\square$

An interesting consequence of propositions 4.7.4 and 4.7.5 is that it is possible to turn the fairly abstract Alexandrov space into a metric space closely corresponding to a metric space defined on the graph. As we noted in proposition 4.6.2 every tree is embeddable in the Euclidean plane. In essence, both the order-theoretic as well as the topological formalization of abstract hierarchical structures discussed above have natural links to *geometrical* representations.

## 4.7.6 Dimension

The notion of *dimensionality* of a topological space is informally the number of independent values (coordinates) needed to describe a point in the space. For instance, a coordinate system printed on paper has 2 dimensions since every point in the system is specified by an  $x$  and a  $y$  coordinate respectively.

### 4.7.6.1 Lebesgue covering dimension

A more precise definition of the dimensionality of a topological space is given by the *Lebesgue covering dimension* (also called *topological dimension*). In order to present this measure we first need to define two other topological concepts: cover and refinement.

Let  $X$  be a set,  $\mathcal{T} = (X, \tau)$  a topological space on  $X$ , and  $Y \subseteq X$  a subset of  $X$ . A collection of sets  $C = \{U_\alpha : \alpha \in A\}$  is called an

open cover of  $Y$  if  $U_\alpha \in \tau$  for each  $\alpha \in A$  and

$$Y \subseteq \bigcup_{\alpha \in A} U_\alpha \quad (4.15)$$

A collection  $D = \{V_\beta : \beta \in B\}$  of sets is said to be a *refinement* of  $C$  if  $D$  is also a cover of  $Y$  and for every set  $V_\beta \in D$  there exists a set  $U_\alpha \in C$  such that  $V_\beta \subseteq U_\alpha$ . The *order* of a cover (and any collection of sets in general)  $C$  is an integer  $n$  that any  $n + 2$  sets in  $C$  has an empty intersection (Edgar, 2008, p. 91).

The *Lebesgue covering dimension* of a topological space  $\mathcal{T} = (X, \tau)$ , denoted  $\text{Cov } \mathcal{T}$ , is defined as an integer  $d$  such that for every open cover  $C$  of  $\mathcal{T}$  the order of any refinement of  $C$  is  $\leq d$  but not  $\leq d - 1$  (Edgar, 2008, p. 92). This rather abstract definition needs a clarification (and an example). The trivial topology  $\{X, \emptyset\}$  is the *coarsest* topology on  $X$ , having the least possible "detail" level, since all points are contained in the same open set. The *finest* possible topology on  $X$  is given by the power set of  $X$ , i.e. the set of *all* subsets of  $X$ . We say that a topology  $\tau_2$  is *finer* than a topology  $\tau_1$  if  $\tau_1 \subseteq \tau_2$ , i.e. every element of  $\tau_1$  is also an element of  $\tau_2$ . Now consider a sequence  $\tau_1, \dots, \tau_n$  of topologies on  $X$  such that  $\tau_1 \subseteq \tau_2 \subseteq \dots \subseteq \tau_n$ . An open cover of any set  $X$  based on a topology  $\tau_j$  will be a refinement of an open cover based on  $\tau_i$  if  $\tau_i \subseteq \tau_j$ . For every possible refinement  $D$  and for every point  $p \in X$  we check the maximal number  $n$  of sets in  $D$  containing any point  $p \in L$ . Assume that the maximal value for  $n$  can be established for any cover of  $X$ . This means that for every point  $p \in X$  any intersection of cover elements, such that the intersection contains  $p$ , is based on  $\leq n$  elements in the refinement. The order of any refinement containing  $p$  is therefore  $n - 1$  and consequently  $\text{Cov } \mathcal{T} = n - 1$ .

As an example, consider the open line segment  $L = (0, 1)$  of real

numbers. Let  $C$  be a collection of open intervals  $(a, b)$  such that  $C$  is a cover of  $L$ . Then for every refinement  $D$  of  $C$  it must clearly be the case that there are points  $p \in L$  contained in at least 2 open intervals in  $D$ . The order of any refinement of  $C$  is therefore  $2 - 1 = 1$  and thus  $\text{Cov } L = 1$ .

#### 4.7.6.2 Inductive dimension

The small and large *inductive dimensions* are, as the name indicates, defined inductively. We will here only focus on the small inductive dimension,  $\text{ind } X$ , since it is sufficient for our purposes. Initially we state by definition that  $\text{ind } \emptyset = -1$ . Further we state that  $\text{ind } X \leq n$  if for a basis  $\mathcal{B} = \{B_1, B_2, \dots\}$  the inductive dimension of the *boundary* of any basis element  $B_i$ , written  $\text{ind } \partial B_i$ , is  $\leq n - 1$ . Finally, we state that  $\text{ind } X = n$  if  $\text{ind } X \leq n$  but not  $\text{ind } X \leq n - 1$  (Edgar, 2008, p. 104). To get an intuitive picture of this definition, and using an informal example, consider an orange. The orange has the inductive dimension 3 since its boundary (its peel) is in principle a surface (a manifold of a 2-dimensional plane) with inductive dimension 2.

If two topological spaces  $\mathcal{S}$  and  $\mathcal{T}$  are homeomorphic, then  $\text{ind } \mathcal{S} = \text{ind } \mathcal{T}$  (Edgar, 2008, p. 104). This means that in order to find the small inductive dimension of a topological space  $\mathcal{T}$  it is sufficient to show that a space *homeomorphic* to  $\mathcal{T}$  has a certain inductive dimension. Further, for any compact metric space  $\mathcal{T}$  (such as any compact subset of the Euclidean spaces) it holds that  $\text{ind } \mathcal{T} = \text{Cov } \mathcal{T}$  (Edgar, 2008, p. 112). We will utilize these theorems in the reasoning below.

### 4.7.6.3 Hausdorff-Besicovitch dimension

The Lebesgue covering dimension is adequate for "regular" topological spaces but may yield counterintuitive results for more irregular sets like *fractals*. A measure that generalizes the notion of dimensionality to any metric space is called the *Hausdorff-Besicovitch dimension*. Let  $\mathcal{D} = (X, d)$  be a metric space and  $E$  a subset of  $\mathcal{D}$ . The *diameter* of a set  $U \subseteq E$  is defined

$$\text{diam } U := \sup\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in U\}$$

Consider an open cover  $C_\delta$  of  $E$  such that for each  $U \in C_\delta$  the constraint  $\text{diam } U \leq \delta$  holds. We call  $C_\delta$  a  $\delta$ -cover of  $E$ . The *Hausdorff  $s$ -dimensional measure* is defined

$$H^s(E) := \lim_{\delta \rightarrow 0} \left( \inf_{C_\delta} \sum_{U \in C_\delta} (\text{diam } U)^s \right) \quad (4.16)$$

An important property of the Hausdorff measure relates to *scaling*. Let  $\psi_\lambda(E)$  be a *similarity transformation* of  $E$  with scale factor  $\lambda$ , such that for every pair of points  $\mathbf{x}, \mathbf{y} \in E$  it holds that  $d(\psi_\lambda(\mathbf{x}), \psi_\lambda(\mathbf{y})) = \lambda d(\mathbf{x}, \mathbf{y})$ . Then the following relationship holds (Falconer, 2003, p. 29):

$$H^s(\psi_\lambda(E)) = \lambda^s H^s(E)$$

The measure  $H^s(E)$  is a *step function* with  $\{0, \infty\}$  as its range. The *Hausdorff dimension* of  $E$  is defined (Falconer, 2003, p. 31) as the smallest value of  $s$  such that  $H^s(E) = 0$ , or formally

$$\dim_H(E) := \inf\{s \in [0, \infty) : H^s(E) = 0\} \quad (4.17)$$

### 4.7.7 Dimensionality of classification

In an ordinary Euclidean  $n$ -space the *dimensionality* of the space is a quantity indicating how many independent values that are needed to specify a point in that space. For instance, in the Euclidean plane ( $\mathbb{R}^2$ ) a point typically needs to be specified by two values, which in the presence of a Cartesian coordinate system often are called the  $x$  and the  $y$  coordinates respectively of the point. Viewed from the perspective of *applied* mathematics the dimensionality of the space is a value indicating the *representation capacity* of the space since, again, it shows how many independent values are contained in the representation of a point in that space. In section 5.6.2 we investigate a property of machine classifiers called *Vapnik-Chervonenkis dimension* which, informally, indicates how many data points that are guaranteed to be classified correctly by at least one element in a set  $\Psi$  of classifiers. Again, the term *dimension* is used to indicate capacity.

It is commonly experienced that the subject division of physical (i.e. non-digital) documents in a library implies a geometrical distribution of the documents along three reciprocally perpendicular directions (left–right, forward–backward, up–down). To illustrate how the concept of dimensionality can be applied to analyze spatial the properties of seemingly abstract phenomena such as classification and classification schedules we will utilize the method of *embedding* objects in topological spaces. An important application area of a dimension analysis is to identify the conditions for the *visualization* of the structure involved. We will also apply the previously mentioned measures of dimensionality: topological (covering) dimension, and Hausdorff-Besicovitch dimension.

In the case of linear and hierarchical orderings of classification codes the proper formalizations are, as we have argued in section 4.5.4

total and partial orders respectively. Since a partial order  $\preceq$  is by definition transitive it follows that  $\preceq$  is its own *transitive closure* (Roman, 2008, p. 20). The minimal representation of a finite partial order  $\preceq$  is its *covering relation*.

#### 4.7.7.1 Dimensionality of linear orderings

As we stated in proposition 4.6.1 the Hasse graph  $H_1 = (V, E_1)$  of a linear order can be embedded into the real line  $\mathbb{R}^1$ , which is in fact the lowest-dimensional Euclidean space in which  $H_1$  is embeddable without self-intersection. We let  $\tilde{\mathcal{E}}_1$  denote the embedding of  $\mathcal{E}_1$  in  $\mathbb{R}^1$  and say that the *embedding dimension* of  $\tilde{\mathcal{E}}_1$  is 1 since the covering dimension of any compact subset of  $\mathbb{R}^1$  is 1 (cf. proposition 4.7.6 below). Since the set  $V$  of vertices is assumed to be finite the embedding  $\tilde{\mathcal{V}}$  of the *vertex space*  $\mathcal{V}$  is totally disconnected with a basis of singleton sets. It follows that every cover of  $\tilde{\mathcal{V}}$  has a finite subcover of singleton sets. The covering dimension  $\text{Cov } \tilde{\mathcal{V}}$  is therefore 0 and this result carries over to all partial orders on  $V$ .

**Proposition 4.7.6.** *The covering dimension  $\text{Cov } \tilde{\mathcal{E}}_1 = 1$ .*

*Proof.* We assume that  $\tilde{\mathcal{E}}_1$  is homeomorphic to the bounded interval  $[a, b] \subset \mathbb{R}^1$ . Any basis  $\mathcal{B}$  of  $\tilde{\mathcal{E}}_1$  consists of open intervals covering  $[a, b]$ . Clearly, the only clopen sets in  $\tilde{\mathcal{E}}_1$  are  $\emptyset$  and  $[a, b]$ . Therefore it must be the case that  $\text{ind } \tilde{\mathcal{E}}_1 > 0$  (Edgar, 2008, p. 87). Moreover, the boundary of an open interval  $(c, d) \in \mathcal{B}$  is  $\{c, d\}$ , which has the small inductive dimension 0. Therefore,  $\text{ind}[a, b] = 1$ . Since  $\mathbb{R}^1$  (and in fact, every Euclidean space  $\mathbb{R}^n$ ) is a metric space and  $[a, b]$  is a compact subset of  $\mathbb{R}^1$  it follows that  $\text{Cov}[a, b] = \text{ind}[a, b] = 1$ .  $\square$

### 4.7.7.2 Dimensionality of hierarchical orderings

Let  $H_2 = (V, E_2)$  be the Hasse graph of a hierarchical ordering,  $\mathcal{E}_2$  the edge space of  $E_2$ . We have seen that  $H_2$  is always embeddable in  $\mathbb{R}^2$  and conclude that the embedding dimension of  $H_2$  is 2, since the covering dimension of  $\mathbb{R}^2$  is 2 (Edgar, 2008, p. 101). The following proposition may therefore, at least initially, appear counterintuitive.

**Proposition 4.7.7.** *Let  $\tilde{\mathcal{E}}_2$  be the embedding of  $\mathcal{E}_2$  in  $\mathbb{R}^2$ . The covering dimension  $\text{Cov } \tilde{\mathcal{E}}_2 = 1$ .*

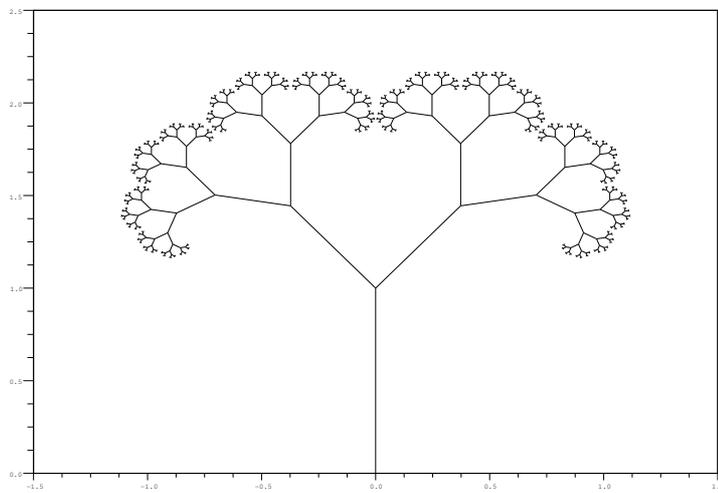
*Proof.* The embedding  $\tilde{\mathcal{E}}_2$  is a finite union of line segments

$$\bigcup_{i=1}^{|E_2|} \ell_i$$

such that  $\ell_i \cong [0, 1]$  for all  $i \in \{1, \dots, |E_2|\}$ . Furthermore,  $\tilde{\mathcal{E}}_2$  is connected. The covering dimension of each line segment  $\ell_i$  is 1 and hence  $\tilde{\mathcal{E}}_2 = 1$ .  $\square$

We see that there is a discrepancy between the embedding dimension and the covering dimension, which is due to the fact that the embedding  $\tilde{\mathcal{E}}_2$  is, informally speaking, locally resembling a line segment but on a global level has an extension in both dimensions of the plane. This phenomenon is mathematically formalized in the theory of *topological manifolds*.

For the visualization of large hierarchies Koike (1995) and Ong et al. (2005) propose the use of *fractal trees*, which is a family of tree structures generated by iterative addition of branches scaled by a dilation factor  $r < 1$  and with a branching angle  $\theta$ . Because of the iterative contraction of the branches the area the generation of the tree converges to a bounded region of  $\mathbb{R}^2$ .



**Figure 4.23.** A binary fractal tree generated using the parameters  $r = 0.58$  and  $\theta = 40^\circ$ .

**Proposition 4.7.8.** *A symmetric fractal tree  $\mathfrak{K}(n, r)$  generated with a dilation value of  $r = 0.5$ , and such that each branch has  $n$  sub-branches, has a Hausdorff dimension  $H(\mathfrak{K}(n, r)) = 1$  for  $n = 2$  and  $H(\mathfrak{K}(n, r)) > 1$  for  $n > 2$ .*

Since the symmetric fractal tree converges set that it is contained in a subset of  $\mathbb{R}^2$  (Mandelbrot & Green, 1999) the proposition can be obtained by studying the convergence of the computationally feasible *box-counting dimension* (for a definition see Edgar, 2008, p. 213) of the fractal tree for successively smaller intervals. This follows from the fact that the box-counting dimension *equals* the Hausdorff dimension for a topological space  $\mathcal{T}$  if  $\mathcal{T}$  can be described as the union  $U$  of disjoint sets and  $U$  is contained in an open subset of  $\mathbb{R}^n$  (Mattila, 1995, p. 67, 81). Since the stem of the tree can, without loss of gener-

alization, be assumed to have unit length, the branches attached to the stem have length  $r$ , the next level of subbranches have length  $r^2$  and so on. The entire length of the tree is therefore the infinite sum

$$\ell(\mathfrak{K}(n, r)) = 1 + nr + n^2r^2 + n^3r^3 + \dots = \sum_{k=0}^{\infty} n^k r^k \quad (4.18)$$

A more straightforward approach to calculating the Hausdorff dimension of an object like  $\mathfrak{K}(n, r)$  is described in (Rellick et al., 1991). This method is based on the observation that the object in question is *self-similar*, i.e. consisting of elements having the same fundamental structure as the object itself. This is clearly the case with symmetric fractal trees; each branch together with its subordinate branches is a tree with the same structure as the entire tree. To compute the Hausdorff dimension of a self-similar structure Rellick et al. (1991) propose a method consisting of two steps:

1. Generate the *Mauldin-Williams graph* for the object in question.
2. Compute the Hausdorff dimension of the object as the value  $s$ , being the solution to the equation system

$$q_i^s = \sum_{j=V; e \in \mathcal{E}_{i,j}} r^s(e) q_j^s \quad (4.19)$$

The Mauldin-Williams graph of  $\mathfrak{K}(n, r)$  is a weighted graph with only one vertex and  $n$  loops, each loop having the weight  $r$ , which results in the equation

$$q^s = \sum_{i=1}^n r^s q^s \quad (4.20)$$

We notice that  $q^s$  can be eliminated from (4.20), reducing the equation

to<sup>3</sup>

$$nr^s = 1 \quad (4.21)$$

which has the general solution for  $s$

$$s = -\frac{\log n}{\log r} = \frac{\log n}{\log 1/r} \quad (4.22)$$

From (4.22) it immediately follows that  $s \geq 1$  iff  $n \geq 1/r$ . For instance, a *ternary* tree (each branch having 3 subbranches) generated with a dilation value  $r = 0.5$  has the Hausdorff dimension

$$H(\mathfrak{K}(n, r)) = -\frac{\log 3}{\log 0.5} = \frac{\log 3}{\log 2} \approx 1.58$$

As this value is higher than the topological dimension 1 of the same tree, and at the same time smaller than the topological dimension of a plane (i.e. 2) it can be stated on an intuitive level that the fractal tree partially fills its embedding space. Similarly, large hierarchical structures of classes can be said to have fractal-like properties, yielding complex self-similar trees.

## 4.8 Concluding remarks

In this chapter we have presented some components of a formal theory of subject classification. We have endeavored to demonstrate that there is a natural link between the semantic and model-theoretic aspects of subject classification as well as its structural and even geometrical properties. The treatment given in this chapter is quite sketchy but the intention of this presentation has been to provide some direc-

---

<sup>3</sup>There is an interesting similarity between equation (4.21) and *Zipf's law* (see section 5.3.1.1).

tions for a more exhaustive theory tying traditional library classification theory with the recent development in automatic document classification based on machine learning. Beside the more common set-theoretic approach we have also employed an even more general and abstract framework, that of *category theory*. On the abstraction level offered by this latter approach the processes of classification can be formally described even without explicitly assuming a certain structure among the documents or the codes in the classification language.

Within the framework of theoretical information retrieval (IR) research topology has been used to formally characterize and compare IR models (Everett & Cater, 1992; Egghe & Rousseau, 1998), based on the similarity measures inherent in those models. Dominich (2000) has also demonstrated that there is a mutual relationship between the pseudometric induced by the similarity measure of a given IR model and the corresponding pseudometric topological space. More specifically, Dominich shows that a topological space generated by a pseudometric basis in fact can be regarded as an IR model. This could be compared to our discussion about the relationship between classification and topology.

A proposition that underlies this chapter is that classification of documents can be regarded as language acts, more specifically statements in a classificatory language, that result in a structure on the documents. We have attempted to state in some detail what is meant by *structure* in this context. Interestingly, these structures can in several ways be described with a terminology that has a clear connection to geometrical/spatial notions, such as measures, metrics, and dimensionality. As we have seen these geometrical notions can be derived from, and therefore are inherent in, the semantic relations between and the logical operations on the codes in the classificatory language.

## **Part II**

# **Automatic text categorization in theory and practice**

## Chapter 5

# Automatic text categorization

### 5.1 Overview

In this chapter we will present the basic concepts and procedures involved in automatic text categorization. Automatic text categorization is typically, but not necessarily, an application of *machine learning*. The *task* involved in the text categorization process is to partition a set of documents into subsets called either *classes* or *clusters*. If machine learning is involved this process is said to be *supervised* if the system performs the classification task by means of a classifier generated from a labeled (precategorized) set of data. In other words, if supervised learning is used the system performs the classification task by means of a statistically induced approximation of a human-produced classification. If the system is designed to perform the partitioning without any information obtained from a precategorized set of documents the task is called *unsupervised* and the resulting groups of documents are typically termed *clusters*.

In order to facilitate machine-based categorization of text documents the documents have to be translated to a computable form,

called *document representations* (Baeza-Yates & Ribeiro-Neto, 2011, p. 58). A *document representation* is an entity that constitutes a formal (usually mathematical) representation of the document. If the document representation is aimed to be used for information retrieval or automatic text categorization it is typically a representation of the *content* of the document. The configuration of these representations will have a considerable impact on the quality of the output from the machine-based classifier (Joachims, 2002, p. 12). We will below describe the translation process from documents to document representations, which typically involves the following steps:

1. tokenization,
2. morphological normalization,
3. feature selection, and
4. generation of feature vectors.

## 5.2 Tokenization and normalization

In this work we regard a text as a *string* (or sequence) of *characters* from a specific *character set*, ignoring the presence of illustrations, typography and other visual aspects of the text. We further assume that such a sequence is readily available for machine-based analysis. Some of the characters form larger units called *words*, whereas other characters are employed to demarcate boundaries between words, clauses, sentences, and paragraphs. The set of characters used to form words is called an *alphabet*.

*Tokenization* is the process of translating a text into a sequence of tokens (for instance word units), whereby word and sentence delim-

iters (such as whitespace, comma, and period) are discarded (Manning et al., 2008, p. 18). For example:

”From the peak of high Olympus”  $\longrightarrow$  (From, the, peak,  
of, high, Olympus)

The general purpose of *morphological normalization* is to exhaustively map related word forms onto the same *lexeme*, i.e. the same lexical element. Formally, let  $V$  be a set of words. Then a morphological normalization process is a surjective map  $f : V \rightarrow V$  such that  $f$  induces an equivalence relation  $\simeq$  over  $V$ , having the definition

$$w_j \simeq w_k \text{ if and only if } f(w_j) = f(w_k) \quad \text{for all } w_j, w_k \in V.$$

The equivalence relation will induce a partition on  $V$  into subsets called *equivalence classes* (Manning et al., 2008, p. 28). All words in the same equivalence class are typically treated as different forms of the same *lexeme*.

One common text normalization method is called *stemming* (Manning et al., 2008, p. 32). Stemming is the process of removing a suffix from a word, according to a set of rules associated with a certain language. For example, using the Porter stemming algorithm (see Porter, 1980) the words *bicycle*, *bicycles*, and *bicycling* are mapped onto the stem *bicycl* whereas *rose*, *roses*, and *rosing* are mapped onto the stem *ros*. Connecting the formalism above to these examples we say that all words having the property that their stems (obtained by a certain stemming algorithm) are pairwise equal, are also members of the same equivalence class. Thus {bicycle, bicycles, bicycling} form an equivalence class; {rose, roses, rosing} another.

Another approach, called *lemmatization* (Manning et al., 2008, p. 32), involves deeper knowledge (which can be augmented by machine learning) about the target language and has the objective to restore each word to its basic lexical form, called *lemma*. Examples of morphological transformations that will not be successfully be addressed by stemming, but possibly by lemmatization, are *thought* → *think*, *went* → *go*, and *rang* → *ring*.

### 5.3 Feature selection and frequency laws

For text categorization tasks it is common procedure to reduce the set of features so that only features that are sufficiently discriminative are retained. *Feature selection* (see e.g. Joachims, 2002, p. 16) is the procedure of selecting those features that are estimated to be have high discriminative capacity down to a certain threshold (or equivalently, removing those features that are estimated to be low discrimination capacity up to a certain threshold). One of the commonly applied methods to this end is *stop word removal*. There are two alternative ways of defining what constitutes a stop word in a collection (Baeza-Yates & Ribeiro-Neto, 2011, p. 220, 226); words with little semantic content or words with a high *df* value (document frequency). The former definition is operationalized by using manually created lists of stop words and the latter by computing a *D*-contextualized *df* value for each term and filtering out terms with a *df* value above a certain threshold (e.g.  $0.75 \cdot |D|$ ).

Another method for feature selection is to rank the features according to feature relevance measures such as information gain or  $\chi^2$  dependency tests (Joachims, 2002, p. 17, 18). The *information gain* measure for a set of classes *C* and a feature *w* (such as the presence

of a certain word) has the definition

$$IG(w, C) := H(C) - H(C|w) - H(C|\neg w) \quad (5.1)$$

where  $H(\cdot)$  denotes *entropy* and  $\neg w$  denotes that the feature  $w$  is not observed (Baeza-Yates & Ribeiro-Neto, 2011, p. 323). The information gain measure is in other words a measure of how much the entropy (probabilistic uncertainty) is reduced by information about a certain feature. The *chi-square* measure quantifies the statistical dependence between a certain class  $c$  and a feature  $w$  (in other words the extent to which the class and the feature are observed together) in a given document collection  $D$  and has the following definition (Baeza-Yates & Ribeiro-Neto, 2011, p. 324).

$$\chi^2(w, c_i) := |D| \cdot \frac{(P(w, c)P(\neg w, \neg c) - P(w, \neg c)P(\neg w, c))^2}{P(w)P(\neg w)P(c)P(\neg c)} \quad (5.2)$$

When a collection of documents is tokenized and translated into a bag of words (see e.g. Baeza-Yates & Ribeiro-Neto, 2011, p. 62) it is commonly observed that the frequencies of words tend to follow certain distribution patterns, termed *frequency laws*. Of these, two will be described in some detail below: *Heaps' law* describing the relationship between the number of documents in the collection and its constituent vocabulary, and *Zipf's law* describing the collection frequency of the words as a function of their rank in a list ordered in descending order by their collection frequencies.

### 5.3.1 Heaps' law

Let  $D$  be a set of documents and  $S$  a sequence of words generated by tokenization of  $D$ . Further, let  $V$  be a vocabulary that iteratively accumulates the unique words from  $S$ , i.e.  $V$  grows for each new word

observed in  $S$ . Heaps' law is an estimation of how  $V$  grows with the size  $n$  of observed words in  $S$  and has the formulation (Heaps, 1978, p. 207):

$$|V| = f(n) = kn^\beta \quad 0 < \beta < 1 \quad (5.3)$$

Experiments have shown that Heaps' law is a very accurate approximation of observed frequencies (Aratljó et al., 1997), and that  $k$  typically takes a value in the interval  $[10, 100]$  and  $\beta$  a value in the interval  $[0.4, 0.6]$  (Baeza-Yates & Ribeiro-Neto, 2011, p. 221). Since

$$\begin{aligned} f'(n) &= \beta kn^{\beta-1} \\ f''(n) &= (\beta^2 - \beta)kn^{\beta-2} \end{aligned}$$

we notice that  $f'(n) > 0$  with  $\lim_{n \rightarrow \infty} f'(n) = 0$  and  $f''(n) < 0$  for all  $n > 0$ . This means that  $f$  is monotonically increasing with  $n$ , but the growth of  $f$  is monotonically *decreasing*. This is congruent with the intuition that most of the new words are observed at the beginning of the word sequence. We can also use Heaps' law to estimate the vocabulary size of a document set. Assume that a document collection  $D$  consists of 1000 documents having an average *length* (number of words) of 200 words. Further assume that  $k = 25$  and  $\beta = 0.5$ . Then according to Heaps' law  $|V| = 25 \cdot 200000^{0.5} \approx 11180$ . Assuming that Heaps' law provides a fairly accurate approximation of the vocabulary size of  $V$  it follows that the vector representations of the documents in  $D$  will be *sparse*, i.e. a large fraction of the vector values will be 0.

### 5.3.1.1 Zipf's law

Zipf's law is a theoretical approximation of the relationship between word frequency and word rank in a collection of texts (Baeza-Yates & Ribeiro-Neto, 2011, p. 71). Among the interesting observations that can be made from this relationship is that only a small fraction of the words in the collection vocabulary correspond to the majority of word occurrences (or tokens) in the documents.

Formally, let  $D$  be a collection of documents and  $V$  a vocabulary (set of terms). Assume that we generate an ordered list  $L$  on the elements in  $V$  based on the number of times each term occurs in  $D$ ; if a term  $t_j$  has a higher number of occurrences than a term  $t_k$  it will have a higher rank in  $L$  (and vice versa). For a given term  $t$  we let  $f(t)$  denote the *frequency* (number of occurrences) of  $t$  in  $D$  and  $r(t)$  the *rank index* of  $t$  in  $L$ . Zipf's law states the following approximate relationship between  $f$  and  $r$ :

$$f(t) = C \cdot r^{-\alpha}(t) \quad \text{where } C, \alpha > 0 \quad (5.4)$$

Equivalently, Zipf's law can be expressed as a linear relationship by logarithmizing the terms in equation (5.4).

$$\log f(t) = \log C - \alpha \cdot \log r(t) \quad \text{where } C, \alpha > 0 \quad (5.5)$$

We will now endeavor to present Zipf's law in even more formal terms. Let  $D$  be a set of documents and  $V$  be a vocabulary (set of words) associated with  $D$ . Further, let  $S$  be a set of  $n$  word occurrences over  $V$ , where a *word occurrence* is defined as an ordered triple  $s = (w, d, i)$  in which  $w \in V$ ,  $d \in D$  and  $i$  denotes the position of a specific instance of  $w$  in  $d$ . We define a surjective mapping  $\tau : S \rightarrow V$  according to  $\tau((w, d, i)) := w$ . Using this mapping, we define the

frequency  $f(w)$  of the word  $w$  as the cardinality (number of elements) of the preimage of  $w$  over  $\tau$ , that is

$$\tau^{\leftarrow}(w) := \{s \in S : \tau(s) = w\}$$

$$f(w) := |\tau^{\leftarrow}(w)|$$

We now assign a *total order*  $\preceq$  on  $V$  as follows

$$w_i \preceq w_j \iff f(w_i) \geq f(w_j) \text{ for all } w_i, w_j \in V.$$

Observe that  $\preceq$  is the mathematical equivalent of an arrangement of the words in *descending order* according to their frequency in  $S$ . Let  $r : V \rightarrow \mathbb{Z}^+$ , where  $\mathbb{Z}^+$  denotes the set of positive integers, be a *ranking function* defined over  $\preceq$ , such that  $r(\inf_{\preceq} S) = 1$  and  $r(w_i) \leq r(w_j)$  iff  $w_i \preceq w_j$ . Further, let  $g : \mathbb{Z}^+ \rightarrow \mathbb{R}$  be a *frequency-by-ranking* function, such that  $g(k)$  yields the frequency of the word having the  $k$ th position in the order induced by  $\preceq$ . Assume that the value for  $g(1)$ , i.e. the highest-ranked term, has been observed and can be used as a constant. Zipf's law states that there exist constants  $C, \beta > 0$  such that

$$g(k) = C \cdot k^{-\beta} \quad k = 1, 2, 3, \dots \quad (5.6)$$

It follows directly from equation (5.6) that  $C = g(1)$ . We reformulate the equation accordingly:

$$g(k) = g(1) \cdot k^{-\beta} \quad k = 2, 3, 4, \dots \quad (5.7)$$

Assuming that a document  $d$  contains  $n$  word occurrences over  $v = |V|$  words we can write  $n$  as a sum of  $g$  for all values of  $k$  up to the

size of the vocabulary:

$$\sum_{k=1}^v g(k) = n$$

Therefore, according to Zipf's law,

$$\sum_{k=1}^v g(k) = \sum_{k=1}^v g(1) \cdot k^{-\beta} = g(1) \cdot \sum_{k=1}^v k^{-\beta} = n \quad (5.8)$$

We let  $H_\beta(v)$  denote the *generalized harmonic number* of order  $v$  of  $\beta$ , defined by

$$H_\beta(v) := \sum_{k=1}^v k^{-\beta}$$

and obtain the following expression for  $g(1)$  by reformulation of equation (5.8):

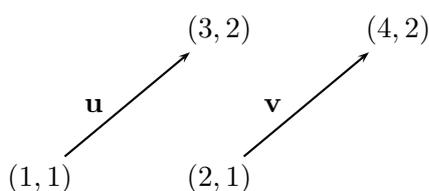
$$g(1) = \frac{n}{H_\beta(v)} \quad (5.9)$$

By plugging (5.9) into (5.7) we finally obtain the following general approximation for the frequency of the  $k$ th highest ranked word in a document containing  $n$  word occurrences, and given a vocabulary of size  $v$ :

$$g(k) = \frac{n}{k^\beta H_\beta(v)} \quad (5.10)$$

## 5.4 Document representation

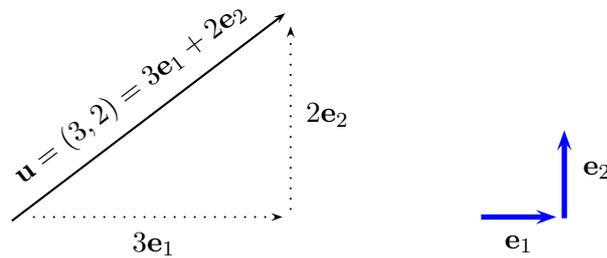
A commonly used model for generating document representations is the *vector space model* (Baeza-Yates & Ribeiro-Neto, 2011, p. 77). The vector space model was presented by Salton et al. (1975) as a statistical framework for content representation in a (Euclidean) vector space. The underlying idea is that each document can be represented by a sequence of quantitative *feature weights*, which in turn corresponds to a *vector* in a suitable vector space. Vectors can intuitively be understood as arrows in a space denoting displacement (or movement) along a straight path from one point to another in the space. What characterizes vectors is therefore the *size* and the *direction* of the displacement. However, vectors do not have a specific *location* in the vector space. For example, in figure 5.1 we notice that the displacement from (1, 1) to (3, 2) has the same size and direction as the displacement from (2, 1) to (4, 2). Therefore  $\mathbf{u} = \mathbf{v}$ .



**Figure 5.1.** Two equal vectors  $\mathbf{u}$  and  $\mathbf{v}$  in the Euclidean plane.

In order to represent vectors in a vector space it is common to define a *coordinate system* on a set of axes pointing in different directions of the space. Given that the coordinate system is rectangular and not curvilinear, these axes can in turn be globally (“everywhere” in the space) represented by the use of a collection of vectors called *basis vectors*.

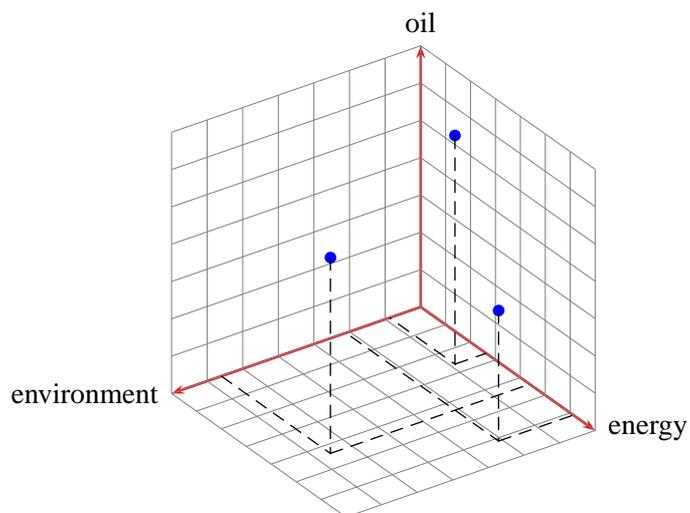
For example, in figure 5.2 we see that a vector denoted as (3, 2) corresponds to a displacement 3 units (3 times the “length” of the



**Figure 5.2.** A vector represented as a linear combination of globally defined basis vectors.

corresponding basis vector) in the direction of the first basis vector (which we can call  $e_1$ ) and 2 units in the direction of the second basis vector (which we can call  $e_2$ ). The resulting displacement is obtained by *adding* the component vectors. This operation, i.e. expressing a vector as a sum of basis vectors scaled by coefficients, is called a *linear combination* of the basis vectors.

In the vector space model the basis vectors correspond to the various features used to represent the documents, for instance features based on statistics related to term frequencies (so called *term weights*, see section 5.4.1 below). Consequently, document vectors are linear combinations of the selected feature vectors, which in turn may correspond to different words in a vocabulary. Assuming that the document vectors denote a displacement from the origin of the coordinate system, then the vectors also “point” to a position in space and are consequently called *position vectors*. In figure 5.3 the document vectors are therefore visualized as dots in a term space. The underlying logic of this framework is that geometric measures such as distance and angle correspond to similarity (or more precisely: *dissimilarity*) between the documents.



**Figure 5.3.** A vector space defined over the words *oil*, *environment*, and *energy*.

A useful measure of document-document similarity in the vector space model is the *cosine* of the angle between the representation vectors (Baeza-Yates & Ribeiro-Neto, 2011, p. 78; Manning et al., 2008, p. 111). Let  $\mathbf{d}_j$  and  $\mathbf{d}_k$  denote two representation vectors in a document space  $\mathcal{F}$  and the notation  $\angle(\mathbf{d}_j, \mathbf{d}_k)$  represent the angle between  $\mathbf{d}_j$  and  $\mathbf{d}_k$ . Then

$$\cos \theta = \frac{\mathbf{d}_j \cdot \mathbf{d}_k}{\|\mathbf{d}_j\| \times \|\mathbf{d}_k\|} \quad \text{where } \theta = \angle(\mathbf{d}_j, \mathbf{d}_k)$$

For computational convenience we may length-normalize each document representation vector so that  $\|\mathbf{d}_j\| = 1$  for every  $\mathbf{d}_j \in \mathcal{F}$ , which effectively turns the cosine into the dot product of the vectors:

$$\cos \theta = \mathbf{d}_j \cdot \mathbf{d}_k$$

### 5.4.1 Term weighting by tf-idf

In order to facilitate the computation of the degree of similarity between documents for an automatic text categorization task, or documents and queries in an information retrieval setting, it is common practice to generate *document representations* using measures based on term frequencies. Intuitively, it is a reasonable assumption that the frequency of a term in a particular document is an indicator of the degree as to which that document “is about” the concept that the term denotes. It should be noted that we will in this section not take into consideration the fact that many terms are homonyms or polysemes (i.e. that they denote different concepts).

In other words, it would be desirable to quantify how much information each term provides about the (content of) a particular document and use this data to make documents comparable to each other or to user queries for classification, clustering, and information retrieval

purposes, by means of formal computations. In machine classification and information retrieval this is typically accomplished by assigning non-negative values to the terms, so called *term weights* (Baeza-Yates & Ribeiro-Neto, 2011, p. 66-67). More specifically, given a document collection  $D$  (a set of documents) and a vocabulary  $V$  (a set of terms) term weighting can be formalized by means of a function

$$w : D \times V \rightarrow \mathbb{R}_{\geq 0}$$

The logic behind the use of term weights is simply that the higher the value of the term weight, the more “important” the term is as a descriptor of the content of a particular document. In order to present the common term weighting schemes we begin by recalling that a tokenized document  $d_j$  can be represented as a list of tokens or equivalently as a function

$$d_j : \mathbb{N} \rightarrow V$$

i.e., an indexed family of words. For instance, a document  $d_j$  beginning with the words “this article presents” would be represented by the function

$$\tilde{d}_j = \{(1, \text{this}), (2, \text{article}), (3, \text{presents}), \dots\}$$

Using this formalization we can proceed to generate equivalence classes over the terms in  $d_j$ . This can be achieved by using equivalence relations such as string equality without token normalization (edit distance 0) or string equality after token normalization (Manning et al., 2008, p. 26).

### 5.4.1.1 The local term weighting scheme *tf*

Term weighting schemes defined over *local* (within-document) frequencies are commonly denoted *term frequency (tf)* schemes (Salton & Buckley, 1988, p. 516; Baeza-Yates & Ribeiro-Neto, 2011, p. 68). Conversely, an important family of term weighting schemes based on *global* (within-collection) frequencies are the *inverse document frequency (idf)* schemes. Below we present a few common definitions of these weighting schemes. A list of alternative definitions of the *tf* and the *idf* weighting schemes is presented by Salton & Buckley (1988).

Let  $d_j$  be any document in a collection  $D$  and let  $E$  denote an equivalence relation, defined as string equality, over the terms in  $d_j$ . The quotient set  $\tilde{d}_j/E$  will then be a partition of  $\tilde{d}_j$  over the equivalence relation. We can define a simple term weighting measure on term  $k_i$  over  $\tilde{d}_j/E$  as the number of equivalence classes in which  $k_i$  is an element, or formally the cardinality of the set  $\{X \in \tilde{d}_j/E : k_i \in X\}$ . Since every term can appear in at most one equivalence class we find that such a measure returns 0 or 1. This weighting method is known as the *binary tf* weighting scheme (Baeza-Yates & Ribeiro-Neto, 2011, p. 73). Formally:

$$f(d_j, k_i) := \begin{cases} 1 & \text{if } k_i \text{ is present in } d_j \\ 0 & \text{otherwise} \end{cases}$$

A potential drawback of this weighting scheme is that it does not take term frequency into account. A different approach is to consider the cardinality of each equivalence class in  $\tilde{d}_j/E$ , which corresponds to the number of occurrences (tokens) of each term. The idea to use local term frequencies to represent document content was proposed already by Luhn (1957). This leads us to a common definition of the

*tf* weighting scheme, namely

$$f(d_j, k_i) := f_{ij}$$

where  $f_{ij}$  denotes the local, unnormalized, frequency of  $k_i$ . In this term weighting scheme the significance of the term in a document is proportional to its frequency. Expressed differently: the weight of the term becomes a linear function of its frequency. It is, however, questionable whether a term that has the local frequency  $f_{ij} = 10$  in a document is 10 times as “important” as a term that occurs only once. For that reason a version of *tf* based on *sublinear scaling* has been devised, having the following definition (see e.g. Manning et al., 2008, p. 116):

$$f(d_j, k_i) := \begin{cases} 1 + \log f_i & \text{if } f_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

The expression *sublinear* indicates that the resulting function yields a value that is constantly less than or equal to the unnormalized, linear weighting scheme.

#### 5.4.1.2 The global term weighting scheme *idf*

Spärck Jones (1972) suggested that terms should not only be weighted according to their local document frequency, but also according to their *specificity*, which is quantified as a value that is monotonically decreasing with their collection frequency *df*, i.e. the number of documents  $d_j \in D$  such that  $k_i$  occurs in  $d_j$ . The magnitude of the specificity is defined by defining a function yielding an output value that is

constrained by an inequality as follows:

$$f(n) := m \quad m \in \mathbb{Z}, m \text{ satisfies the following} \quad (5.11)$$

$$\text{inequality: } 2^{m-1} < n \leq 2^m$$

It is easily verified that  $m = \lceil \log_2(n) \rceil$  is a solution to the inequality in (5.11). Spärck Jones proceeds to define the weight of term  $k_i$  as the sum

$$w(k_i) = f(N) - f(df_i) + 1 = \lceil \log_2(N) \rceil - \lceil \log_2(df_i) \rceil + 1$$

This weighting scheme is today known as the *idf* measure and is typically defined as follows (see e.g. Manning et al., 2008, p. 108):

$$\text{idf}(k_i) := \log\left(\frac{N}{df_i}\right) = \log(N) - \log(df_i) \quad (5.12)$$

The theoretical motivation behind the definition of the *idf* measure can be derived from Zipf's law (Baeza-Yates & Ribeiro-Neto, 2011, p. 71). A presentation of this frequency law is given in section 5.3.1.1. A variant of Zipf's law states that the document frequency  $n(r)$  of the  $r$ th most frequent term in a document collection (i.e. the term having rank  $r$  if we rank the terms in descending order by their total number of occurrences in the document collection) is given by the power law

$$n(r) \approx Cr^{-\alpha} \quad \alpha > 0$$

Taking logarithms and letting  $\alpha = 1$  we get

$$\log(n(r)) \approx \log(C) - \log(r)$$

from which it immediately follows

$$\log(r) \approx \log(C) - \log(n(r)) = \log\left(\frac{C}{n(r)}\right)$$

Making the approximation  $C = N$  (i.e. we assume that the most frequent term occurs in every document in the collection) and assuming that term  $k_i$  has the  $r$ th highest frequency we get:

$$\log(r) \approx \log\left(\frac{N}{df_i}\right)$$

which is the definition of the idf measure. As we can see, the idf weight of a term  $k_i$  is in this derivation interpreted as the logarithm of its rank with respect to a vocabulary ranked in descending order according to their total frequency in the collection.

The *idf* measure can also be approximated and interpreted using an information-theoretic approach (Robertson, 2004). Assume that term  $k_i$  has been observed in  $df_i$  documents in  $D$ . Then the probability of randomly selecting a document  $d_j \in D$  containing  $k_i$  is

$$P(k_i|d_j) = \frac{df_i + 1}{N + 2}$$

where we have applied the Laplace's *rule of succession* to the probability calculations (see Amati & Van Rijsbergen, 2002). The *self-information* (see e.g. Katona & Nemetz, 1976) of the event of selecting a document containing  $k_i$  is thereby

$$\text{Inf}(k_i) = -\log(P(k_i)) = \log\left(\frac{1}{P(k_i)}\right) = \log\left(\frac{N + 2}{df_i + 1}\right)$$

Thus, the idf measure applied to a term  $k_i$  could be interpreted as the quantity of information “contained” in  $k_i$ , but Robertson (2004) ar-

gues that this interpretation has a somewhat shaky foundation, mainly because it is difficult to determine the appropriate event space against which this interpretation could be made in an information retrieval setting.

#### 5.4.1.3 The joint term weighting scheme *tf-idf*

The *tf* and the *idf* weighting schemes are often combined into a joint term-weighting scheme called *tf-idf*. This is performed by simply multiplying the *tf* and the *idf* values, i.e.

$$tfidf(d_j, k_i) := f(d_j, k_i) \cdot idf(k_i)$$

This weighting scheme will consequently favor terms that have a high local term frequency but a low collection frequency, of which the latter indicates a high specificity or discrimination capacity (Salton & Buckley, 1988, p. 516)

Amati & Van Rijsbergen (2002) show that the *tf-idf* weighting scheme can be derived directly through probabilistic reasoning by assuming that the probability of randomly selecting a term  $k_i$  that occurs in  $df_i$  out of  $N$  documents is (using additive smoothing for the probability calculation):

$$P(k_i) = \frac{df_i + 0.5}{N + 1}$$

Assuming that the occurrences of  $k_i$  appear independently from each other, the probability of finding  $f_i$  instances of term  $k_i$  is

$$P(f_i) = \left( \frac{df_i + 0.5}{N + 1} \right)^{f_i}$$

The self-information associated with the event of randomly selecting a document containing  $f_i$  instances of  $k_i$  is thereby

$$\text{Inf}(f_i) = f_i \log_2 \left( \frac{N + 1}{df_i + 0.5} \right) \quad (5.13)$$

As we can see, the second factor in equation (5.13) is very similar to the common definition of the *idf* weight in equation (5.12).

#### 5.4.2 Term weighting by divergence from randomness

A probabilistic language model for term weighting was proposed by Amati & Van Rijsbergen (2002) and is based on the assumption that the significance of a term  $k_i$  in a document  $d$  is proportional to the degree to which its local frequency  $f_i$  differs from the frequency that would have been observed under a random distribution of the terms among the documents. For this reason this term weighting framework is named *divergence from randomness* (Baeza-Yates & Ribeiro-Neto, 2011, p. 113). The divergence from randomness framework appears to have been exclusively used as an information retrieval model, but the underlying term weighting principles are not constrained to information retrieval and could reasonably be applied to automatic classification as well. Two kinds of probabilities are considered in this framework:

- The probability  $\text{Prob}_1$  of observing the local term frequency  $f_i$  in a document under a randomness model  $M$ .
- The probability  $\text{Prob}_2$  of observing  $k_i$  within its elite set.

Here, the *elite set* of a term  $k_i$  is defined as the subset of the document collection containing  $k_i$ . For the construction of the weighting scheme the information content (self-information) associated with  $\text{Prob}_1$  is

used together with a *risk* factor defined as  $\text{Risk} := 1 - \text{Prob}_2$ . The rationale behind using the latter factor is that the risk of  $k_i$  being a non-informative term is high when its local frequency is low (compare the reasoning behind the *tf* weighting scheme). Further, the local  $f_i$  frequency ought to be normalized according to the length of the document, since longer documents tend to have higher *tf* values (and vice versa). The joint weighting function is defined:

$$w(d_j, k_i) := \text{Inf}_1 \cdot \text{Risk} = (-\log_2(\text{Prob}_1)) \cdot (1 - \text{Prob}_2)$$

#### 5.4.2.1 Models of randomness

Two models of randomness are considered by Amati & Van Rijsbergen (2002): the *Bernoulli* model and the *Bose-Einstein* model. The *Bernoulli model* assumes that the term frequencies have a binomial distribution with probability  $p = 1/N$  for the event of observing a term appearing in a document and probability  $q = (N - 1)/N$  for the event of not observing  $k_i$ . Such a binomial distribution has the probability mass function

$$P(f_i) = \binom{F_i}{f_i} p^{f_i} q^{F_i - f_i}$$

where  $F_i$  denotes the *collection frequency* of  $k_i$ , i.e. the total number of occurrences of the term in the collection. Since this formula is computationally intractable Amati & Van Rijsbergen propose the approximation of the binomial distribution in terms of a Poisson distribution with  $\lambda = p \cdot f_i$ . They further apply Stirling's formula to approximate the factorial operation, yielding the following expression for the information content  $\text{Inf}_1$ :

$$\text{Inf}_1(f_i) \approx f_i \cdot D(\phi, p) + 0.5 \log_2(2\pi f (1 - \phi))$$

where  $\phi := F_i/f_i$ ,  $p := 1/N$ , and  $D(\phi, p) := \phi \cdot \log_2(\phi/p) + (1 - \phi) \cdot \log_2((1 - \phi)/(1 - p))$ .

The Bose-Einstein model is based on a somewhat different probability distribution, pertaining to the collection frequency of a term being the sum of its within-document frequencies. As above, we let the collection frequency of a word  $k_i$  be denoted by  $f$ . This value acts as a constraint on the sum

$$F_i = f_{i,1} + f_{i,2} + \dots + f_{i,N} \quad (5.14)$$

where  $f_{i,j}$  denotes the term frequency of  $k_i$  in document  $d_j$ . Following Amati & Van Rijsbergen (2002, p. 365) we let  $s_1$  denote the number of possible solutions to the equation (5.14), treating  $f_{i,1}, f_{i,2}, \dots$  as free (but non-negative) variables. Now, consider the sum

$$F_i - f_{i,k} = f_{i,1} + f_{i,2} + \dots + f_{i,k-1} + f_{i,k+1} + \dots + f_{i,N} \quad (5.15)$$

i.e. the modified sum formed by subtracting the local frequency of term  $f_i$  in document  $d_k$ . Let  $s_2$  denote the number of possible solutions to the equation (5.15). In Bose-Einstein statistics the probability of observing  $f_i$  occurrences of a term  $k_i$  is given by the quotient  $s_2/s_1$ . Amati & Van Rijsbergen (2002, p. 366) approximate this probability by a geometric distribution as follows:

$$P(f_i) \approx \left( \frac{1}{1 + \lambda} \right) \left( \frac{\lambda}{1 + \lambda} \right)^{f_i}$$

where  $\lambda := F_i/N$ , and hence,

$$\begin{aligned} \text{Inf}_1(f_i) &= -\log_2 \left\{ \left( \frac{1}{1+\lambda} \right) \left( \frac{\lambda}{1+\lambda} \right)^{f_i} \right\} \\ &= \log_2(1+\lambda) - f_i(\log_2(\lambda) - \log_2(1+\lambda)) \\ &= (1+f_i) \cdot \log_2(1+\lambda) - f_i \log_2(\lambda) \end{aligned}$$

The probability  $\text{Prob}_2$  of finding a term  $k_i$  in an elite document is calculated in two different ways in the DFR model. The *Laplace* model (Amati & Van Rijsbergen, 2002, p. 365) is derived by considering the probability of observing  $f_i$  occurrences of a term, given that  $f_i - 1$  occurrences have already been observed. This probability is computed, according to Laplace's rule of succession as

$$\text{Prob}_2 = \frac{f_i}{f_i + 1}$$

Consequently, the risk of  $k_i$  not being a useful descriptor of  $d$  is given by

$$\text{Risk} = 1 - \text{Prob}_2 = 1 - \frac{f_i}{f_i + 1} = \frac{f_i + 1 - f_i}{f_i + 1} = \frac{1}{f_i + 1} \quad (5.16)$$

The *Bernoulli* model is derived by considering a Bernoulli trial  $B(F_i, p, q)$ , where  $p$  denotes the probability of observing a term  $k_i$  in a document  $d$  in the elite set (and conversely  $q = 1 - p$  the probability of *not* observing  $k_i$ ). Assume that the total number of occurrences of  $k_i$  in the elite set is  $F_i$ . Then the probability of observing  $f_i$  occurrences of the term  $k_i$  in  $d$  is

$$P(F_i, f_i, p, q) = \binom{F_i}{f_i} p^{f_i} q^{F_i - f_i} \quad (5.17)$$

Assume further that one more occurrence of  $k_i$  is added to the elite

set. The probability of now observing  $f_i + 1$  occurrences of  $k_i$  in  $d$  is given by

$$P(F_i + 1, f_i + 1, p, q) = \binom{F_i + 1}{n + 1} p^{f_i + 1} q^{F_i - f_i} \quad (5.18)$$

The *incremental rate* between the probabilities in (5.18) and (5.17) is defined in Amati & Van Rijsbergen (2002) as

$$\alpha = 1 - \frac{P(p, q, F_i + 1, f_i + 1)}{P(p, q, F_i, f_i)} = 1 - \frac{F_i + 1}{f_i + 1} \cdot p \quad (5.19)$$

Letting  $n$  denote the number of documents in the elite set and estimating  $p = 1/n$  we get

$$\alpha = 1 - \frac{F_i + 1}{f_i + 1} \cdot \frac{1}{n} = 1 - \frac{F_i + 1}{n(f_i + 1)} \quad (5.20)$$

The risk of  $k_i$  not being a useful descriptor of  $d$  is consequently according to this model

$$\text{Risk} = 1 - \alpha = \frac{F_i + 1}{n(f_i + 1)} \quad (5.21)$$

Since the local frequencies  $f_i$  are dependent on the length  $\ell(d)$  of the document, Amati & Van Rijsbergen also perform a normalization of the local frequencies using the ratio  $sl/\ell(d)$ , where  $sl$  denotes the average length of the documents in the collection. This normalization is defined on a general form according to

$$fn_i := f_i \cdot \log_2 \left( 1 + \frac{sl}{\ell(d)} \right) \quad (5.22)$$

For the practical calculations of the probabilities described above the normalized frequencies  $fn_i$  are used instead of  $f_i$ .

## 5.5 Supervised and unsupervised classification

As we noted in the introduction to this chapter, automatic text categorization tasks performed by means of machine learning fall into one of two categories of learning algorithms: methods using supervised and unsupervised classification respectively. Since this work is mainly concerned with an algorithm for supervised classification, the support vector machine, we will begin with summarily presenting the principles of unsupervised classification and then go into more detail with respect to supervised classification.

### 5.5.1 Unsupervised classification (Clustering)

*Clustering* or *cluster analysis* is the name of a family of methods aimed at exploring the structure of a data set without any preparatory induction of category patterns, as in supervised learning. Similar to supervised classification the result will be a partition of objects into groups (called *clusters*) but in the case of clustering these groups are induced from the data itself. Clustering methods are generally divided into *partitional* (also called *flat*) and *hierarchical* methods. Partitional clustering methods will only divide the data into clusters with no particular structure between the clusters, whereas hierarchical methods generate hierarchical layers of clusters. Cluster analysis is an important technique in *data mining*.

### 5.5.2 $k$ -means clustering

One of the most commonly used methods for partitional cluster analysis is *k-means* clustering (see e.g. Manning et al., 2008, p. 331). A somewhat peculiar detail about this method is that the number  $k$  of clusters has to be specified before the analysis starts, which means

that several sessions with different values for  $k$  may be needed to find an optimal clustering.  $k$ -means clustering is typically defined in a Euclidean space  $\mathbb{R}^n$ .

When the clustering procedure starts  $k$  cluster points are distributed randomly in the representation space. The overall distance (or deviation) from the cluster centers is measured by a *residual sum of squares* (RSS) function. The aim of the clustering procedure is to find a configuration of the cluster points such that the RSS function is minimized. Let  $\mathbf{u}$  be a cluster point and  $\omega_{\mathbf{u}}$  the cluster of documents assigned to  $\mathbf{u}$ . Then the RSS function has the definition

$$\rho(\mathbf{u}) := \sum_{\mathbf{d} \in \omega_{\mathbf{u}}} (\mathbf{d} - \mathbf{u})^2 \quad (5.23)$$

The partial derivative of  $\rho$  with respect to a single cluster centroid coordinate  $u_i$  is

$$\frac{\partial \rho}{\partial u_i} = \sum_{\mathbf{d} \in \omega_{\mathbf{u}}} 2(d_i - u_i) \quad (5.24)$$

$\nabla \rho = \mathbf{0}$  when  $\mathbf{u}$  is the centroid of all  $\mathbf{d} \in \omega_{\mathbf{u}}$ . Therefore, by setting  $\mathbf{u}$  to the centroid of all  $\mathbf{d} \in \omega_{\mathbf{u}}$  the reconfiguration of cluster points will iteratively make  $\rho$  converge to a minimum. One problem that has to be considered when using the  $k$ -means algorithm is its sensitivity to *outliers* (single objects with large dissimilarity to the other objects in the set). If such objects are chosen as seed elements for the clustering the procedure may end up with singleton clusters. Another problem that occasionally affects the  $k$ -means method is that the number of non-empty clusters generated by the procedure is less than the stipulated value  $k$ , yielding empty clusters (Manning et al., 2008, pp. 363, 364).

### 5.5.3 Hierarchical clustering

Hierarchical clustering methods work, as mentioned above, by generating a hierarchical structure of clusters. The visual representation of the resulting structure is typically a *dendrogram* (Manning et al., 2008, pp. 377, 378). It is possible to regard a hierarchical cluster structure as a collection of *flat* clusterings ordered by inclusion, in such way that a flat clustering  $C_i$  is included in a flat clustering  $C_j$  if every element in  $C_i$  is a proper subset of an element in  $C_j$ . Hierarchical clustering algorithms can be divided into two families according to their direction of work: *bottom-up* (agglomerative) and *top-down* (divisive).

Let  $X$  be a set of objects that are subjected to a clustering algorithm. A bottom-up clustering is performed by initially defining a cluster structure  $C^0$  such that each element in  $X$  is a *singleton* cluster in  $C^0$ . The clustering algorithm then proceeds to generate a sequence of clusterings  $C^1, C^2, \dots$  by successively merging clusters from the previous clusterings. To select which clusters that should be merged in each successive step a combination of an object (for instance, document) similarity measure and a cluster similarity measure is used. *Single-link* algorithms define the similarity between two clusters  $C_i$  and  $C_j$  as the *largest* object similarity  $\text{sim}(d_a, d_b)$ , such that  $d_a \in C_i$  and  $d_b \in C_j$ . In contrast, *complete-link* algorithms define cluster similarity as the *smallest* object similarity between the clusters. Single-link methods typically produce *elongated* clusters, i.e. clusters with a large maximal distance between objects within the same cluster, whereas complete-link methods tend to generate clusters with a small diameter (Manning et al., 2008, p. 382). There also exist algorithms utilizing the average object similarity between clusters (*average-link* algorithms) as well as the vector *centroid* (if applicable) of each cluster (Manning et al., 2008, pp. 389, 391).

### 5.5.4 Supervised classification

The aim of *supervised* classification is to assign documents to a specific set of classes, using information induced from a set of labeled training documents. We can summarize the steps that typically are involved in supervised classification as follows.

1. **Initialization.** A classification algorithm is selected for the classification task and configured. In the case of the SVM algorithm this involves the choice of a learning kernel (see section 6.6) as well as an initial selection of *hyperparameters*.
2. **Induction.** The classification system is provided with a *training set* of categorized documents, from which it induces an estimated target function, called *classifier*, that optimally expresses the relationship between document content and document category.
3. **Intermediate evaluation.** The induced classifier  $\psi$  is applied to a *development set*, which may be obtained from the training set, to evaluate the classification performance of  $\psi$ .
4. **Adjustment.** Hyperparameters pertaining to the applied classification algorithms are adjusted to induce a new classifier.
5. **Repetition** of the process from step 2, if desired.
6. **Final evaluation.** The classification system is evaluated against a *test set* distinct from the training set and the development set.

In the presentation of classification methods below, the following symbols will be consistently used. We let  $D$  be a set of documents,  $\mathcal{D}$  a set of document representations and  $\mathcal{C}$  a set of categories.

### 5.5.5 $k$ -nearest neighbor classification

The intuitive idea behind *k-nearest neighbor* classification is that there is a correspondence between the conceptual closeness of certain physical documents in a collection and the measurable closeness of the corresponding document representations. This idea is akin to the cluster hypothesis (Van Rijsbergen, 1979)

This hypothesis may be simply stated as follows: *closely associated documents tend to be relevant to the same requests.*

Though the cluster hypothesis is expressed in IR-theoretic terms, using the notion of *relevance*, the same reasoning is applicable to classification – closely associated documents tend to belong to the same category. By “closely associated documents” van Rijsbergen refers to documents having similar *representations*, whether they be in the form of similar keyword sets or similar feature vectors.

Let  $\mathcal{D}$  be embedded in a metric space (typically  $\mathbb{R}^n$ ) and let  $\mathbf{d}$  denote a single point (a representation of a document  $d$ ) in  $\mathcal{D}$ . The system of *neighborhoods* (see section 4.7.3) around  $\mathbf{d}$  is the collection of open balls  $\{\mathbf{p} \in \mathcal{D} : \delta(\mathbf{d}, \mathbf{p}) < r\}$  where  $r$  is a real number. Assume that  $\mathbf{d}$  represents an unclassified document and that there exists a neighborhood  $\mathcal{N}_k(\mathbf{d})$  of  $\mathbf{d}$  containing precisely  $k$  representations of classified documents. The primary decision rule in  $k$ -NN classification is to assign  $d$  to the class that is most frequent in  $\mathcal{N}_k(\mathbf{d})$  (Baeza-Yates & Ribeiro-Neto, 2011, p. 299). If this decision rule is not sufficient a *tie-breaking* rule has to be applied, for instance to use another value of  $k$  or to measure a score based on the similarity between  $\mathbf{d}$  and each category in  $\mathcal{N}_k(\mathbf{d})$  (Manning et al., 2008, p. 275).

### 5.5.6 Naïve Bayesian inference

The *Bayesian* approach to document classification is, as the name suggests, based on probability theory (Mitchell, 1997, p. 154). Given a document  $d$  and available evidence  $E$ , the general principle behind probabilistic classification is to assign the class  $c$  to  $d$  that maximizes the probability  $P(c|d, E)$ . Probabilistic classification models are generally easy to update with new evidence, which together with their theoretical simplicity make them popular in environments where the evidence landscape is continually changing, for instance in *spam filtering* software.

Since the probability  $P(c|d)$  is unfeasible to compute directly it is normally transformed into a computational form by means of *Bayes' theorem* (hence the name of this method):

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \propto P(d|c)P(c) \quad (5.25)$$

Since the probability  $P(d)$  is constant over all classes it is normally left out from the inference process and the remaining decision rule thereby gets the formulation

$$\psi(d) = \arg \max_{c \in \mathcal{C}} P(d|c)P(c) \quad (5.26)$$

It may not be immediately obvious why  $P(d|c)$  is more feasible to compute than  $P(c|d)$ , but the important difference lies in the fact that  $P(d|c)$  can be "decomposed" in terms of the *features* making up the document representation. Assuming that  $d$  is a text document and the features are related to the vocabulary  $V$  of  $D$  it should be possible to express  $P(d|c)$  as a function of the probabilities  $P(t_i|c)$ , where  $t_i$  represents a single term in the vocabulary. The expression  $P(t_i|c)$  will then have the interpretation "the probability of finding term  $t_i$  in

a document assigned to class  $c$ ".

If we make another important assumption, namely that the probabilities  $P(t_i|c)$  are *independent* from each other (which is a naïve assumption, in a statistical sense) we can apply the *multiplication theorem* of probability theory:

$$P(d|c) = P(t_1|c) \times P(t_2|c) \times \cdots \times P(t_{|V|}|c) = \prod_{t_i \in d} P(t_i|c) \quad (5.27)$$

By these assumptions the decision rule is then formulated

$$\psi(d) = \arg \max_{c \in \mathcal{C}} P(c) \prod_{t_i \in V(d)} P(t_i|c) \quad (5.28)$$

To avoid computational *underflow* (the condition that appears when values become so small that they cannot be stored with sufficient precision in the computer's primary memory) the probabilities are normally logarithmized:

$$\psi(d) = \arg \max_{c \in \mathcal{C}} \log P(c) + \sum_{t_i \in V(d)} \log P(t_i|c) \quad (5.29)$$

### 5.5.7 Perceptrons and feedforward neural networks

*Perceptrons* belong to the category of *linear* classifiers and are the basic units of *neural networks* (Mitchell, 1997). By supervised learning the perceptron induces a weight vector  $\mathbf{w}$  which is used to yield the following output, given an input vector  $\mathbf{x}$  (which could for instance be a document representation vector):

$$\psi(\mathbf{w}, \mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ -1 & \text{otherwise} \end{cases} \quad (5.30)$$

The function in (5.30) is called an *activation function*. Since the equation  $\mathbf{w} \cdot \mathbf{x} + b = 0$  defines a hyperplane in  $\mathbb{R}^n$  we notice that perceptrons perform classification by means of separating hyperplanes, comparable to the support vector machine algorithm (see ch. 6). The important difference between these two methods lies in how  $\mathbf{w}$  is induced. One disadvantage with the definition in (5.30) is that  $\psi$  is a step function and therefore not continuously differentiable. Further, it may not perform well in situations where the decision boundary between categories is nonlinear. A viable alternative in situations where the linear activation function is inadequate is the logistic (sigmoid) function

$$\psi_\sigma(\mathbf{w}, \mathbf{x}) = (1 + \exp(\mathbf{w} \cdot \mathbf{x} + b))^{-1} \quad (5.31)$$

A learning rule for the perceptron can be constructed by a *gradient descent* (see e.g. Bertsekas, 1999, p. 24f) approach to minimizing an error function. Let  $\mathcal{X} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell)\}$ , where  $\mathbf{x}_i \in \mathbb{R}^n$  and  $y_i \in \{-1, +1\}$  be a set of representation vectors together with associated class labels. A suitable error function is the sum of the squared errors over all data points in  $\mathcal{X}$ , i.e.

$$e(\mathbf{w}, \mathbf{x}) = \frac{1}{2} \sum_{i=1}^{\ell} (y_i - \psi(\mathbf{w}, \mathbf{x}_i))^2 \quad (5.32)$$

If  $\psi$  were differentiable we could now localize the minimum of  $e$  by applying a gradient descent algorithm. We notice that  $\psi$  can be con-

ceived as a *composition* of two functions  $f, g$  such that  $\psi = g \circ f$ :

$$\begin{aligned}
 f(\mathbf{w}, \mathbf{x}) &= \mathbf{w} \cdot \mathbf{x} + b \\
 g(x) &= \begin{cases} +1 & \text{if } x > 0 \\ -1 & \text{otherwise} \end{cases}
 \end{aligned} \tag{5.33}$$

We define a modified error function using  $f$  instead of  $\psi$ :

$$\hat{e}(\mathbf{w}, \mathbf{x}) = \frac{1}{2} \sum_{i=1}^{\ell} (y_i - f(\mathbf{w}, \mathbf{x}_i) - b)^2 = \frac{1}{2} \sum_{i=1}^{\ell} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 \tag{5.34}$$

Observe that we have cancelled the threshold  $b$  in the formulation of  $\hat{e}$ , since  $b$  will not affect the solution to the problem of finding an optimal configuration of  $\mathbf{w}$ . Clearly,  $e$  is minimized if  $\hat{e}$  is minimized, which occurs when  $\hat{e}(\mathbf{w}, \mathbf{x}) = 0$ . The partial derivative of  $\hat{e}$  with respect to a single weight  $w_k$  is

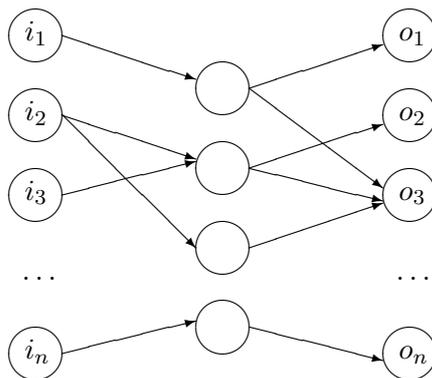
$$\frac{\partial \hat{e}}{\partial w_k} = \sum_{i=1}^{\ell} [\mathbf{x}_i]_k \mathbf{w} \cdot \mathbf{x}_i - [\mathbf{x}_i]_k y_i = \sum_{i=1}^{\ell} (\mathbf{w} \cdot \mathbf{x}_i - y_i) [\mathbf{x}_i]_k \tag{5.35}$$

The gradient descent rule states that  $w_k$  should be modified in the *negative* direction of  $\frac{\partial \hat{e}}{\partial w_k}$  by a learning rate  $\eta$ ; thus the learning rule for  $w_k$  becomes

$$w_k \leftarrow w_k - \eta \sum_{i=1}^{\ell} (\mathbf{w} \cdot \mathbf{x}_i - y_i) [\mathbf{x}_i]_k \tag{5.36}$$

A *feedforward neural network* consists of a set of perceptrons arranged as an acyclic directed graph (see figure 5.4 below) with at least three node layers: an input layer, a hidden layer, and an output layer. Each edge in the graph corresponds to a connection (“synapse”) be-

tween perceptrons. The activation of the network is performed by initiating the input nodes and iteratively propagating the values as signals to the connecting perceptrons followed by generating output from the perceptrons by their activation functions, until the layer of output nodes is reached.



**Figure 5.4.** A feedforward neural network

For the training of the network a *backpropagation algorithm*, which is a generalization of the perceptron learning rule, is used. Theoretically there is a risk that learning methods obtained by gradient descent become "trapped" at a local minimum, but according to Mitchell (1997, p. 98) this approach to learning in a feedforward neural network still performs well for many "real" problems. Neural networks are suitable in situations where the data consists of many attribute-value pairs and is noisy, i.e. contains irregularities (Mitchell, 1997, p. 85), which makes them useful for text classification.

## 5.6 Elements of statistical learning theory

Supervised machine learning is based on the principle that a system learns to perform a task by means of set of labeled training examples (Joachims, 2002, p. 7). A common application of supervised machine learning is *supervised classification*, which entails the learning of an optimal classifier. In the case of pattern recognition the training set contains both positive and negative examples (i.e. non-examples) of a target pattern. In this section we will present the principles of statistical learning underlying the theory of supervised machine learning in general and the *support vector machine* algorithm (see chapter 6) in particular.

### 5.6.1 Empirical risk minimization

Let  $X = \{x_1, x_2, \dots, x_k\}$  be a set of categorizable objects, for instance text documents. We assume that there exists a set  $\mathcal{C}$  of classification codes and a *classifier*  $\varphi : X \rightarrow \mathcal{C}$ . We further assume the existence of a function  $\rho : X \rightarrow \mathcal{F}$  mapping each element in  $X$  onto a representation in a representation space  $\mathcal{F}$  (for instance a vector space). Let  $\mathcal{X} = \{(\mathbf{x}_i, c_i)\}_{i=1}^{\ell}$ , where  $\mathbf{x}_i = \rho(x_i)$  and  $c_i = \varphi(x_i)$ , be the set of labeled representations for the elements in  $X$ . As we have stated in section 5, the general purpose of machine learning for classification is to find a (statistical) classifier  $\psi : \mathcal{F} \rightarrow \mathcal{C}$  such that  $\psi$  approximates  $\varphi$ .

Let  $J$  be an index set used to enumerate a collection  $\Psi$  of machine classifiers. We let  $\psi_\alpha$  denote the classifier in  $\Psi$  having index  $\alpha \in J$ . To measure the overall error of a machine classifier  $\psi_\alpha$  we introduce a *loss function*  $L : \mathcal{F} \times \mathcal{C} \times J \rightarrow \{0, 1\}$  (Vapnik, 1998, p. 25) with

the following definition.

$$L(\mathbf{x}, c, \alpha) = \begin{cases} 0 & \text{if } \psi_\alpha(\mathbf{x}) = c \\ 1 & \text{if } \psi_\alpha(\mathbf{x}) \neq c \end{cases} \quad (5.37)$$

where  $(\mathbf{x}, c)$  denotes a labeled representation. Assuming the probability distribution function  $P(c|\mathbf{x})$  we define the overall loss as the *expectation value* of  $L(x, \alpha)$  over the support  $\mathcal{X}$ , which is given by the Lebesgue-Stieltjes integral

$$R(\alpha) = \int_{\mathcal{X}} L(\mathbf{x}, c, \alpha) P(c|\mathbf{x}) \quad (5.38)$$

We call  $R(\alpha)$  the *statistical risk* for the problem at hand. If  $P(c|\mathbf{x})$  would be fully known the objective of machine learning would be to minimize the expression in (5.38). For most practical problems, however, the probability distribution is unknown and the statistical risk has to be estimated. This can be accomplished by using a sample  $S \subset X$  of precategorized elements and calculating the arithmetic mean of the loss function over  $S$ :

$$R_{\text{emp}}(\alpha) = \frac{1}{\ell} \sum_{i=1}^{\ell} L(\mathbf{x}_i, c_i, \alpha) \quad \ell = |S|, c_i = \varphi(x_i) \quad (5.39)$$

The expression in (5.39) is called the *empirical risk* for the problem at hand over the sample  $S$ . The working principle of selecting a classifier  $\psi_\alpha$ , based on finding the minimum of  $R_{\text{emp}}(\alpha)$  by sampling  $X$ , is called *empirical risk minimization* (ERM) (Vapnik, 1998, p. 32).

Assume a set of machine classifiers indexed by the variable  $\alpha \in J$ . Let  $\alpha_\ell$  be the index of the classifier minimizing the statistical or empirical risk (depending on context) for the sample size  $\ell$ . The ERM principle is said to be *consistent* if the following conditions hold (Vapnik,

1998, p. 80).

$$R(\alpha_\ell) \xrightarrow[\ell \rightarrow \infty]{P} \inf_{\alpha \in J} R(\alpha) \tag{5.40}$$

$$R_{\text{emp}}(\alpha_\ell) \xrightarrow[\ell \rightarrow \infty]{P} \inf_{\alpha \in J} R(\alpha) \tag{5.41}$$

The notation  $\xrightarrow{P}$  denotes *convergence in probability*. A sequence  $(x_1, x_2, \dots)$  converges in probability towards  $y$  if given any  $\varepsilon > 0$  the limit  $\lim_{n \rightarrow \infty} P(|x_n - y| > \varepsilon) = 0$ . Condition (5.40) states that the statistical risk computed over the entire domain of labeled representations converges in probability towards the minimal statistical risk when the sample size increases. Condition (5.41) states that also the empirical risk, which is an estimation of the statistical risk, converges towards the minimal statistical risk.

Vapnik (1998, p. 119) shows that the ERM principle is guaranteed to be consistent if it holds that

$$\lim_{\ell \rightarrow \infty} \frac{G^\Lambda(\ell)}{\ell} = 0 \tag{5.42}$$

where  $G^\Lambda(\ell) = \ln N^\Lambda(\mathbf{x}_1, \dots, \mathbf{x}_\ell)$  and  $N^\Lambda(\mathbf{x}_1, \dots, \mathbf{x}_\ell)$  denotes the number of classifications on a set  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_\ell\}$  that can be produced by a set  $\Psi$  of functions (classifiers). If  $N^\Lambda(\ell) = 2^\ell$  the set of functions can generate any classification on  $X$ , which in practical terms means that at least one classifier in  $\Psi$  will produce a desired classification. However, the condition in (5.42) does not provide any computable bounds to estimate the rate of convergence of  $R_{\text{emp}} \rightarrow R$  as a function of  $\ell$ . If the training is not based on sufficiently large sets of training examples it is quite possible that the empirical risk is not a good estimate of the statistical risk and that training evaluation may yield overly optimistic bounds on the statisti-

cal risk (Vapnik, 1998, p. 220). This is due to the fact that the statistical risk is probabilistically bounded not only by the empirical risk but also by the *capacity* of the candidate classifiers in  $\Psi$ . To proceed with formulating bounds for the statistical risk in relation to both empirical risk and classification capacity we will first define a core concept of Vapnik's statistical learning theory: the *Vapnik-Chervonenkis dimension*.

### 5.6.2 Vapnik-Chervonenkis dimension

The *Vapnik-Chervonenkis dimension* (or *VC dimension* for short) of a set  $\Psi$  of machine classifiers is informally the largest number of points in a representation space that will be correctly classified by at least one classifier in  $\Psi$ , irrespective of how the points are configured in the space.

Let  $X$  be a finite set with cardinality  $k$ . Consider the task of partitioning  $X$  into exactly 2 subsets. This procedure can be described formally as follows. Let  $\mathcal{F}$  be a collection of functions  $X \rightarrow \{0, 1\}$  and  $\simeq_f$  an equivalence relation over a function  $f \in \mathcal{F}$  with the definition

$$x \simeq_f y \text{ iff } f(x) = f(y)$$

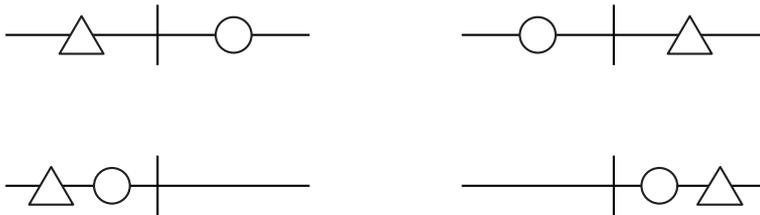
We further let

$$[x] = \{y \in X : x \simeq_f y\}$$

be the equivalence class of  $x$  over  $\simeq_f$  and the quotient set  $X / \simeq_f$  the set of all equivalence classes in  $X$  by  $\simeq_f$ . Each quotient set defined in this way is a partition of into 2 subsets. Since there can be at most  $2^k$  different functions in  $\mathcal{F}$  we find that there are at most  $2^k$  different

ways to partition  $X$  into exactly 2 subsets. We say that  $\mathcal{F}$  *shatters*  $X$  if the functions in  $\mathcal{F}$  can partition  $X$  in all  $2^k$  different ways.

Let  $\Psi$  be a collection of classifiers  $\mathbb{R}^n \rightarrow C$ . The VC dimension of  $\Psi$ , often denoted with a single  $h$ , is defined as the *largest* number of vectors in  $\mathbb{R}^n$  that can be shattered by  $\Psi$  (Vapnik, 1998, p. 147).



**Figure 5.5.** The largest number of vectors that can be shattered by a linear function in  $\mathbb{R}^1$  is 2.

**Proposition 5.6.1.** Let  $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ , where  $\mathbf{x}_i \in \mathbb{R}^n$  and  $y_i \in \mathbb{Z}$ , be a set of labeled points and  $\Psi : \mathbb{R}^n \rightarrow \mathbb{Z}$  a set of classifiers. Let the VC dimension of  $\Psi$  be denoted by  $h$ . Assume that  $h < \ell$ . Then the empirical risk  $R_{\text{emp}}(\alpha^*)$  of the most successful classifier in  $\psi_{\alpha^*}(\mathbf{x}) \in \Psi$  is

$$R_{\text{emp}}(\alpha^*) \begin{cases} = 0 & \text{if } \ell \leq h \\ \leq 1 - h/\ell & \text{otherwise} \end{cases}$$

*Proof.* The following bounds hold for the number of clusters  $N(\mathcal{X})$  that can be generated by a set of indicator functions having VC di-

mension  $h$  (Vapnik, 1998, p. 147):

$$N(\mathcal{X}) \begin{cases} = 2^\ell & \text{if } \ell \leq h \\ \leq \sum_{i=0}^h C_\ell^i = \sum_{i=0}^h \binom{\ell}{i} & \text{otherwise} \end{cases} \quad (5.43)$$

From the binomial expansion

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$$

it follows that

$$\sum_{k=0}^n \binom{n}{k} = \sum_{k=0}^n \binom{n}{k} 1^{n-k} 1^k = (1 + 1)^n = 2^n.$$

From (5.43) we therefore conclude that if  $\ell > h$  the strict bound  $N(\mathcal{X}) < 2^\ell$  must hold. The VC dimension is therefore a bound on the guaranteed number of data points that will be correctly classified by the functions in  $\Psi$  and in the worst case  $\ell - h$  data points will be incorrectly classified by the best classifier in  $\Psi$ , in which case the loss will be

$$\frac{1}{\ell}(\ell - h) = 1 - \frac{h}{\ell}.$$

□

**Proposition 5.6.2.** *The VC dimension of a linear classifier (i.e. a separating hyperplane) in  $\mathbb{R}^n$  is  $n + 1$ .*

For the reasoning leading to this proposition, albeit not a strict proof, see Burges (1998, p. 125).

### 5.6.3 Structural risk minimization and regularization

In contrast to the ERM principle, which is only focused on the minimization of the empirical risk, the *structural risk minimization* (SRM) principle stipulates that also the complexity or capacity of the classifier should be considered (Vapnik, 1998, p. 219). What complicates the picture is that there is also a dependency between capacity and empirical risk: the higher the capacity of the classifier the smaller the anticipated empirical risk (and vice versa).

To approach this problem in an orderly fashion the SRM principle states that the set  $\Psi$  of candidate classifier sets should be assigned a structure (hence the name of this principle) of nested subsets  $\Psi_1 \subset \Psi_2 \subset \dots \subset \Psi_k$ , such that each element  $\Psi_i$  has a finite VC dimension  $h_i$  and  $\Psi_j \subset \Psi_k$  if  $h_j < h_k$  (Vapnik, 1998, pp. 221, 222; Burges, 1998, p. 128). Vapnik shows that it is possible use the VC dimension to formulate a constructive (computable) bound on the relationship between the statistical risk  $R(\alpha)$  and the empirical risk  $R_{\text{emp}}(\alpha)$ . Let  $\psi_\alpha$  be a classifier having the finite VC dimension  $h$ . We further let  $\ell$  denote the number of elements in the training set and  $\varepsilon$  a positive constant. Then the following probabilistic bound holds on the difference between the statistical and the empirical risk (Vapnik, 1998, p. 148):

$$P \left\{ \sup_{\alpha \in J} \left| \int_{\mathcal{X}} L(x, \alpha) P(c|\mathbf{x}) - \frac{1}{\ell} \sum_{i=1}^{\ell} L(x, \alpha) \right| > \varepsilon \right\} < 4 \exp \left\{ \left( \frac{h(1 + \ln(2\ell/h))}{\ell} - \left( \varepsilon - \frac{\ell}{2} \right)^2 \right) \ell \right\}$$

$$\begin{aligned}
 & P\{\sup_{\alpha \in J} |R(\alpha) - R_{\text{emp}}(\alpha)| > \varepsilon\} < \\
 & 4 \exp \left\{ \left( \frac{h(1 + \ln(2\ell/h))}{\ell} - \left(\varepsilon - \frac{\ell}{2}\right)^2 \right) \ell \right\} \quad (5.44)
 \end{aligned}$$

We let  $\eta$  denote the left-hand probability in (5.44) and proceed to solve the inequality for  $\varepsilon$ .

$$\begin{aligned}
 \eta & < 4 \exp \left\{ \left( \frac{h(1 + \ln(2\ell/h))}{\ell} - \left(\varepsilon - \frac{\ell}{2}\right)^2 \right) \ell \right\} \\
 \frac{\ln \frac{\eta}{4}}{\ell} & < \frac{h(1 + \ln(2\ell/h))}{\ell} - \left(\varepsilon - \frac{1}{\ell}\right)^2 \\
 \left(\varepsilon - \frac{1}{\ell}\right)^2 & < \frac{h(1 + \ln(2\ell/h))}{\ell} - \frac{\ln \frac{\eta}{4}}{\ell} \\
 \varepsilon & < \sqrt{\frac{h(\ln \frac{2\ell}{h} + 1) - \ln \frac{\eta}{4}}{\ell}} + \frac{1}{\ell}
 \end{aligned}$$

We conclude that with probability  $1 - \eta$  it holds that

$$R(\alpha) - R_{\text{emp}}(\alpha) \leq \varepsilon < \sqrt{\frac{h(\ln \frac{2\ell}{h} + 1) - \ln \frac{\eta}{4}}{\ell}} + \frac{1}{\ell} \quad (5.45)$$

and therefore with probability  $1 - \eta$ :

$$R(\alpha) < R_{\text{emp}}(\alpha) + \sqrt{\frac{h(\ln \frac{2\ell}{h} + 1) - \ln \frac{\eta}{4}}{\ell}} + \frac{1}{\ell} \quad (5.46)$$

Interestingly, the probabilistic bound in (5.46) on the statistical risk  $R(\alpha)$  is (cf. the remarks in (Burges, 1998, p. 124)):

1. independent of  $P(c|\mathbf{x})$ ,
2. decreasing with the sample size  $\ell$ ,

3. increasing with the empirical risk, and
4. increasing with the VC dimension.

The SRM principle states that the optimal selection of a machine classifier minimizes the probabilistically bounded risk in (5.46) and thereby strikes a balance between training error (as measured by the empirical risk) and *capacity* (as expressed by the VC dimension).

The SRM principle is akin to the idea of *regularization*. Schölkopf & Smola (2002, p. 87) state that empirical risk minimization may not guarantee good generalization behavior of the classifier and as a countermeasure it is proposed (Schölkopf & Smola, 2002, p. 89) that the set of possible classifiers should be limited by formulating a *regularized risk functional* as

$$R_{\text{reg}}(\alpha) := R_{\text{emp}}(\alpha) + \lambda\Omega(\alpha) \quad (5.47)$$

where  $\Omega$  is a *regularization* function (for instance vector norm) and  $\lambda$  is a constant regulating the trade-off between empirical risk and the classification capacity of  $\psi_\alpha$ . The modified objective is then to minimize  $R_{\text{reg}}(\alpha)$  instead of  $R_{\text{emp}}(\alpha)$ . An instructive example with relevance to SVM, stated by Schölkopf & Smola (2002, p. 89), is to let  $\psi = \mathbf{w}$  (the vector defining the separating hyperplane) and  $\Omega = \frac{1}{2}\|\cdot\|^2$  (the squared vector norm) whereby the objective is to minimize

$$R_{\text{reg}} := R_{\text{emp}} + \frac{\lambda}{2}\|\mathbf{w}\|^2 \quad (5.48)$$

For a treatment of the role of the vector  $\mathbf{w}$  in SVM classification and the minimization of  $\frac{1}{2}\|\mathbf{w}\|^2$ , see section 6.4 and onward. For a developed application of the idea of regularization, see section 6.5.1 on the C-SVM classifier. In the following chapter we will look at the details of the SVM algorithm and see how it builds on the SRM principle.

## Chapter 6

# Support vector machines

### 6.1 Introduction

*Support vector machines* (SVMs) are a family of machine learning methods based on Vladimir Vapnik's principle of *structural risk minimization* (Joachims, 2002, p. 35). Originally published as a method for binary classification (Boser et al., 1992; Cortes & Vapnik, 1995) there are today extensions for multiclass problems (Crammer et al., 2001), hierarchical classification (Cai & Hofmann, 2004), problems with structured output such as trees and graphs (Tsochantaridis et al., 2005), cluster analysis (Ben-Hur et al., 2001) as well as regression problems (Drucker et al., 1997).

### 6.2 Comparative performance

Joachims (2002, p. 46-49) states that an automatic text categorization task is typically characterized by the following properties:

1. *A high-dimensional feature space.* In the section on Heaps' law (section 5.3.1) we observed that the vocabulary size of a docu-

ment collection grows monotonically with the number of word occurrences, which typically results in a high number of feature words.

2. *Sparse document vectors.* Whereas the feature space is typically high-dimensional, the number of unique words in each document is typically much lower than the size of the collection vocabulary.
3. *Heterogeneous use of terms.* Because of the semantic diversity in natural languages each document category can be expressed by many different words, which makes it important to retain a large portion of the vocabulary to guarantee a high degree of category recognition. Another option is to use techniques for dimension reduction of the feature space, while retaining most of the important information of the original feature space.
4. *High level of redundancy.* In each document there are typically many words indicating its subject category, which makes the feature set of the document redundant with respect to automatic classification.
5. *Zipf's law.* As discussed in section 5.3.1.1 words are typically distributed by their frequency in document collections as well as single document in such a fashion that only a small fraction of the vocabulary accounts for a majority of the actual word occurrences.

Joachims (2002, p. 72) argues on the basis of the theoretical properties of the SVM algorithm that it can be justified as an appropriate choice for automatic text categorization.

In a comprehensive study comparing the SVM algorithm to 16 other classification methods and 9 other regression methods Meyer et

al. (2003) found that SVM generally performed well on classification tasks, but were slightly outperformed by other methods such as neural networks and random forests, on regression problems. In a comparative study of 5 different text categorization methods (SVM,  $k$ -NN, a variant of the vector space model (VSM),  $k$ -NN with LSA, and VSM with LSA) on two collections consisting of electronic messages and Usenet articles respectively Cardoso-Cachopo et al. (2003) found that SVM outperformed the other methods on the *mean ranked reciprocal* evaluation measure, closely followed by  $k$ -NN based on LSA. In an extensive text categorization study performed on a modified subset of the Reuters Corpus Volume I test collection, called RCV1-v2, Lewis et al. (2004) found that both the applied variants of SVM learning outperformed the  $k$ -NN and the Rocchio algorithms.

### 6.3 Quadratic programming

In this section we present the basic notions of *quadratic programming*, which is a subdiscipline of a mathematical area called *optimization*. This section is intended to give a preliminary background for understanding some aspects of the theoretical treatment of the SVM algorithm.

The general objective of *optimization*, or *mathematical programming* as the field is variously called, is to find the optimum (minimum or maximum) of a function  $f(\mathbf{x})$ , where  $\mathbf{x} \in \mathbb{R}^n$  is a vector of variables (Nocedal & Wright, 2006, p. 2). In the treatment that follows, a point (i.e. a configuration of variables) where  $f$  is optimized is denoted by  $\mathbf{x}^*$ . If the arguments in  $\mathbf{x}$  are constrained by a set of equalities and/or inequalities the problem is called *constrained optimization*. Any point, optimum or not, that satisfies the constraints is called a *feasible point* and is denoted with a tilde (for instance  $\tilde{\mathbf{x}}$ ) in

this work. The set of all feasible points is called the *feasible region* of the problem (Fletcher, 1987, p. 140). A *feasible direction* from a feasible point  $\tilde{\mathbf{x}}$  is a vector  $\mathbf{d}$  such that  $\tilde{\mathbf{x}} + \mathbf{d}$  is also a feasible point (Bertsekas, 1999, p. 215).

The function that is sought to be optimized is called the *objective function*. A problem posed as a minimization problem may easily be transformed into a maximization problem (and vice versa) by optimizing  $-f(\mathbf{x})$  instead of  $f(\mathbf{x})$ . We may therefore summarize the goal of constrained optimization as the following general problem (Fletcher, 1987, p. 140). We seek to

$$\begin{aligned} & \text{minimize} && f(\mathbf{x}) && \text{where } \mathbf{x} \in \mathbb{R}^n \\ & \text{subject to} && g_i(\mathbf{x}) = 0 && \text{for all } i \in \mathcal{E} \\ & && h_j(\mathbf{x}) \geq 0 && \text{for all } j \in \mathcal{I} \end{aligned} \tag{6.1}$$

Here,  $\mathcal{E}$  denotes the index set of all equality constraints and  $\mathcal{I}$  the index set of all inequality constraints. A constraint  $g$  is said to be *active* at  $\tilde{\mathbf{x}}$  if  $g(\tilde{\mathbf{x}}) = 0$ . The *active set*  $\mathcal{A}(\tilde{\mathbf{x}})$  is the set of indices  $\mathcal{E} \cup \{j \in \mathcal{I} : h_j(\tilde{\mathbf{x}}) = 0\}$  (Nocedal & Wright, 2006, p. 308). If the objective function consists of a linear polynomial (for instance  $2x_1 + 3x_2 - 5$ ) the problem is categorized as *linear programming*. Consequently, if the objective function is nonlinear the problem is termed *nonlinear programming*. A subtype of nonlinear programming problems, that is particularly pertinent to the theory of support vector machines, is the case where the objective function consists of a second-order polynomial, called *quadratic programming* (QP). The canonical form of the objective function of a QP problem is

$$f(\mathbf{x}) := \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{c}^\top \mathbf{x} \tag{6.2}$$

where  $\mathbf{Q}$  is an  $n \times n$  symmetric matrix (Fletcher, 1987, p. 230).

### 6.3.1 Primal and dual form

For some optimization problems it is possible to find a solution by solving a different problem, called the *dual* of the initial problem (which is conversely called the *primal* problem). In other words, a solution to the dual form of the problem is a solution to its primal form. A trivial optimization example is the problem of finding the minimum of  $f(x)$  has the dual problem of finding the maximum of  $-f(x)$ . A dual formulation with particular relevance to the kind of quadratic programming problems that appear in the context of SVM optimization is called the *Wolfe dual* (Fletcher, 1987, p. 219). Assume that we seek to

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) \\ &\text{subject to} && h_j(\mathbf{x}) \geq 0 \quad \text{for all } j \in \mathcal{I} \end{aligned} \tag{6.3}$$

where  $f$  is a *convex* function and each constraint  $h_j$  is a *concave* function. In the Wolfe dual form of this problem we seek to

$$\begin{aligned} &\text{maximize} && \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) \\ &\text{subject to} && \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0} \\ &&& \lambda_j \geq 0 \quad \text{for all } j \in \mathcal{I} \end{aligned} \tag{6.4}$$

Here  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_{|\mathcal{I}|})$  is a vector of *Lagrange multipliers* and the function  $\mathcal{L}$  is called a *Lagrange function*. The general form for a Lagrange function is given in section 6.3.3 below.

### 6.3.2 Lagrange multipliers

*Lagrange multipliers* are coefficients attached to the constraint functions of an optimization problem. These coefficients often occur in the formulation of necessary and sufficient conditions for optimality. Assume that we seek to

$$\begin{aligned} &\text{minimize } f(\mathbf{x}) \\ &\text{subject to } g(\mathbf{x}) = 0 \end{aligned} \tag{6.5}$$

In other words we search for the minimum of  $f(\mathbf{x})$  under a single equality constraint on  $\mathbf{x}$ . Let  $\mathbf{x}^*$  be a solution to (6.5), i.e. a feasible point and a minimizer of  $f$ . Since  $\mathbf{x}^*$  is a minimizer of  $f$  there cannot exist a feasible direction  $\mathbf{d}$  such that  $f(\mathbf{x}^* + \mathbf{d}) < f(\mathbf{x}^*)$ , or equivalently,  $f(\mathbf{x}^* + \mathbf{d}) - f(\mathbf{x}^*) < 0$ . The *Taylor expansion* up to order  $n$  of an analytic function  $\varphi$  is given by the series

$$\varphi(x) = \varphi(a) + \varphi'(a)(x-a) + \frac{\varphi''(a)}{2!}(x-a)^2 + \dots + \frac{\varphi^{(n)}(a)}{n!}(x-a)^n \tag{6.6}$$

Using (6.6) we find that the Taylor expansion of  $g(\mathbf{x}^* + \mathbf{d})$  up to the first order is  $g(\mathbf{x}^*) + \nabla g(\mathbf{x}^*)^\top \mathbf{d} = \nabla g(\mathbf{x}^*)^\top \mathbf{d}$ . But since  $\mathbf{d}$  is a feasible direction we have

$$\nabla g(\mathbf{x}^*)^\top \mathbf{d} = 0 \tag{6.7}$$

Conversely, the first-order Taylor expansion of  $f(\mathbf{x}^* + \mathbf{d})$  is  $f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^\top \mathbf{d}$ . According to the assumption about  $\mathbf{x}^*$  there does not exist a feasible direction such that

$$0 > f(\mathbf{x}^* + \mathbf{d}) - f(\mathbf{x}^*) \approx \nabla f(\mathbf{x}^*)^\top \mathbf{d} \tag{6.8}$$

Hence,  $\mathbf{x}^*$  is a minimizer of  $f$  iff there exists no feasible direction  $\mathbf{d}$  such that (6.7) and (6.8) are simultaneously satisfied. It is easily shown (see Nocedal & Wright, 2006, p. 309) that a feasible direction  $\mathbf{d}$  satisfying (6.7) and (6.8) is possible to construct precisely when  $\nabla f(\mathbf{x}^*) \nparallel \nabla g(\mathbf{x}^*)$ . Consequently,  $\mathbf{x}^*$  is a minimizer of  $f$  precisely when  $\nabla f(\mathbf{x}^*) \parallel \nabla g(\mathbf{x}^*)$ , or equivalently stated,

$$\nabla f(\mathbf{x}^*) = \lambda \nabla g(\mathbf{x}^*) \quad \lambda \in \mathbb{R} \setminus \{0\} \quad (6.9)$$

This condition can be generalized to the following rule. We form the *Lagrangian function*

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) := f(\mathbf{x}) - \sum_{i \in \mathcal{E}} \lambda_i g_i(\mathbf{x}) \quad \boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_{|\mathcal{E}|}) \quad (6.10)$$

The coefficients in  $\boldsymbol{\lambda}$  are called *Lagrange multipliers*. We can summarize the observations above in the following general rule. A feasible point  $\tilde{\mathbf{x}}$  is a minimizer of  $f$  iff

$$\nabla_{\mathbf{x}} \mathcal{L}(\tilde{\mathbf{x}}, \boldsymbol{\lambda}) = \mathbf{0} \quad (6.11)$$

### 6.3.3 Karush-Kuhn-Tucker conditions

Assume that a feasible point  $\tilde{\mathbf{x}}$  is a solution to a general constrained optimization problem  $\mathcal{P}$  on the form (6.1). Further assume that a *regularity condition* holds at  $\tilde{\mathbf{x}}$  on the constraints. One important regularity condition is called the *linear independence constraint qualification* (LICQ) and holds if the gradients for all active constraints (equality constraints and inequality constraints = 0) are linearly independent (Nocedal & Wright, 2006, p. 320). Another regularity condition that is pertinent for this work holds if the inequality constraints are *linear* (Fletcher, 1987, p. 203, 204). We construct a generalized Lagrange

function as follows (Bertsekas, 1999, p. 316):

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &:= f(\mathbf{x}) - \sum_{i \in \mathcal{E}} \lambda_i g_i(\mathbf{x}) - \sum_{j \in \mathcal{I}} \mu_j h_j(\mathbf{x}) \quad \text{where} \\ \boldsymbol{\lambda} &= (\lambda_1, \dots, \lambda_{|\mathcal{E}|}), \boldsymbol{\mu} = (\mu_1, \dots, \mu_{|\mathcal{I}|}) \end{aligned} \tag{6.12}$$

Under the assumptions above the *Karush-Kuhn-Tucker (KKT) conditions* state that there exists a vector  $\boldsymbol{\lambda}$  and a vector  $\boldsymbol{\mu}$  such that (Nocedal & Wright, 2006, p. 321)

$$\nabla_{\mathbf{x}}^* \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})|_{\mathbf{x}=\mathbf{x}^*} = \mathbf{0} \tag{6.13}$$

$$g_i(\mathbf{x}^*) = 0 \quad \forall i \in \mathcal{E} \tag{6.14}$$

$$h_i(\mathbf{x}^*) \geq 0 \quad \forall i \in \mathcal{I} \tag{6.15}$$

$$\lambda_i \geq 0 \quad \forall i \in \mathcal{E} \tag{6.16}$$

$$\lambda_i g_i(\mathbf{x}^*) = 0 \quad \forall i \in \mathcal{E} \tag{6.17}$$

$$\mu_j h_j(\mathbf{x}^*) = 0 \quad \forall i \in \mathcal{I} \tag{6.18}$$

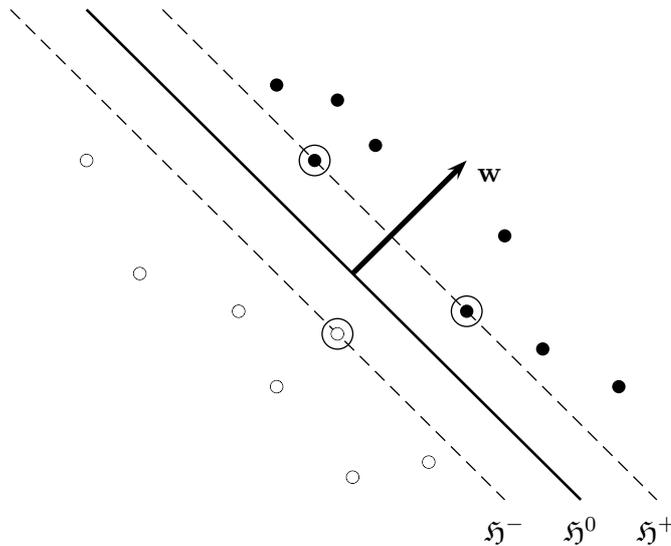
The conditions (6.17) and (6.18) are commonly called the *KKT complementarity conditions* (Nocedal & Wright, 2006, p. 321) and state that for every constraint, either the Lagrange multiplier or the constraint itself has to be equal to zero.

The KKT conditions state the *necessary* criteria for a feasible point  $\tilde{\mathbf{x}}$  to be a solution to  $\mathcal{P}$ . It is naturally desirable to also formulate conditions for determining if  $\tilde{\mathbf{x}}$  is a solution to  $\mathcal{P}$ , i.e. *sufficient* conditions for a solution. Assuming that there are no equality constraints present and that the regularity condition is met for a feasible point  $\tilde{\mathbf{x}}$  then the KKT conditions are sufficient for optimality if the ob-

jective function  $f$  is convex and all constraints are concave functions (Fletcher, 1987, p. 218).

We have thereby presented the fundamentals of quadratic programming, and will now proceed to give an overview of the essential components of the SVM algorithm.

## 6.4 Linear SVM using a hard margin



**Figure 6.1.** Linear separation of vectors in  $\mathbb{R}^n$

Let  $X$  be a set of categorized objects (for instance, documents). Further, let  $\mathbf{X}$  denote a set of feature vectors  $\mathbf{x}_i \in \mathbb{R}^n$  assigned to the elements in  $X$ . We also assume that each vector  $\mathbf{x}_i$  is associated with a class label  $y_i \in \{-1, +1\}$ . Such a binary classification scenario typically arises in *pattern recognition* where the machine learning task is to recognize a target category  $c$  and separate it from other categories (labeled patterns). Given a target class  $c$  we assign to each feature

vector  $\mathbf{x}_i \in X$  a category label  $y_i$  according to the scheme

$$y_i = \begin{cases} +1 & \text{if } \varphi(x_i) = c \\ -1 & \text{if } \varphi(x_i) \neq c \end{cases}$$

and define the set  $\mathcal{X}$  as the collection of all ordered pairs  $(\mathbf{x}_i, y_i)$  over  $X$ . Let  $\mathbf{X}^+$  be the set of all feature vectors  $\mathbf{x}_i$  such that  $y_i = +1$ . These vectors are said to represent the *positive examples* of  $c$ . Conversely, let  $\mathbf{X}^-$  be the set of all feature vectors  $\mathbf{x}_i$  such that  $y_i = -1$ , representing the *negative examples* of  $c$ . Assume that  $\mathbf{X}^+$  and  $\mathbf{X}^-$  are linearly separable, i.e. that it is possible to construct a hyperplane  $\mathfrak{H}^0$  fully separating the vectors in  $\mathbf{X}^+$  from the vectors in  $\mathbf{X}^-$ . Further, let  $\mathfrak{H}^+$  and  $\mathfrak{H}^-$  be hyperplanes, parallel and equidistant to  $\mathfrak{H}^0$ , defining the *margin* between the sets of feature vectors. The distance between  $\mathfrak{H}^+$  and  $\mathfrak{H}^-$  will then be equal to the shortest distance, as measured in the direction normal to  $\mathfrak{H}^0$ , between the vectors in  $\mathbf{X}^+$  and  $\mathbf{X}^-$  respectively (Burges, 1998).

Let  $\mathbf{w}$  be a vector that is *normal* to  $\mathfrak{H}^0$  and  $\mathbf{x}_0$  a fixed point (origin point) in  $\mathfrak{H}^0$ . Then  $\mathfrak{H}^0$  can be defined as the set of all points  $\mathbf{x}$  such that

$$\mathbf{w} \cdot (\mathbf{x} - \mathbf{x}_0) = 0 \tag{6.19}$$

By further letting  $b = -\mathbf{w} \cdot \mathbf{x}_0$  we find that the following condition holds for all points  $\mathbf{x} \in \mathfrak{H}^0$ :

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{6.20}$$

Before we proceed with the derivation of the margin hyperplanes we should consider how a classifier based on  $\mathfrak{H}^0$  can be defined. The principle is simply to consider on which side of the hyperplane a certain

data point falls and assign one of the labels  $\{-1, +1\}$  accordingly (cf. Burges, 1998, p. 134). We state this formally as follows:

$$\psi(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b) \quad (6.21)$$

We further define a fixed point  $\mathbf{x}_0^+ \in \mathfrak{H}^+$  as the *displacement* of  $\mathbf{x}_0$  in the direction of  $\mathbf{w}$  scaled by a real number  $\alpha$ , i.e.  $\mathbf{x}_0^+ = \mathbf{x}_0 + \alpha\mathbf{w}$ . Similarly, we define a fixed point  $\mathbf{x}_0^- \in \mathfrak{H}^-$  as  $\mathbf{x}_0^- = \mathbf{x}_0 - \alpha\mathbf{w}$ . The distance  $d$  between any of the margin hyperplanes and  $\mathfrak{H}_0$  is then

$$d = \|(\mathbf{x}_0 + \alpha\mathbf{w}) - \mathbf{x}_0\| = \alpha\|\mathbf{w}\| \quad (6.22)$$

Conversely, we see that the scalar  $\alpha$  can be expressed in terms of the distance  $d$ :

$$\alpha = \frac{d}{\|\mathbf{w}\|} \quad (6.23)$$

We find now that the hyperplane  $\mathfrak{H}^+$  is the set of points  $\mathbf{x}$  such that

$$\mathbf{w} \cdot (\mathbf{x} - \mathbf{x}_0 - \alpha\mathbf{w}) = \mathbf{w} \cdot \mathbf{x} - \mathbf{w} \cdot \mathbf{x}_0 - \alpha\mathbf{w} \cdot \mathbf{w} = 0 \quad (6.24)$$

Since we have defined  $b = -\mathbf{w} \cdot \mathbf{x}_0$  it follows from (6.23) and (6.24) that

$$\mathfrak{H}^+ = \{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = d\|\mathbf{w}\|\} \quad (6.25)$$

and conversely

$$\mathfrak{H}^- = \{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} - b = -d\|\mathbf{w}\|\} \quad (6.26)$$

By scaling  $\mathbf{w}$  so that

$$d = 1/\|\mathbf{w}\| \quad (6.27)$$

we can simplify the expressions for the hyperplanes even further:

$$\begin{aligned} \mathfrak{H}^+ &= \{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = 1\} \\ \mathfrak{H}^- &= \{\mathbf{x} : \mathbf{w} \cdot \mathbf{x} - b = -1\} \end{aligned} \quad (6.28)$$

Since the fixed point displacement was defined in the direction of  $\mathbf{w}$  it is clear that the sets  $\mathbf{X}^+$  and  $\mathbf{X}^-$  obtain the following bounds:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x} + b &\geq 1 \text{ for all } x \in \mathbf{X}^+ \\ \mathbf{w} \cdot \mathbf{x} - b &\leq -1 \text{ for all } x \in \mathbf{X}^- \end{aligned} \quad (6.29)$$

These bounds can be expressed in a joint rule for all pairs  $(\mathbf{x}_i, y_i) \in \mathcal{X}$ :

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad (6.30)$$

To obtain the general bounds stated in (6.30) we have defined the distance between any of the margin hyperplanes and  $\mathfrak{H}^0$  as  $d = 1/\|\mathbf{w}\|$ . It immediately follows that the entire margin has width  $2/\|\mathbf{w}\|$ . Interestingly, the *shattering* capacity of  $\psi$  is directly related to the distance  $2d$  between the margin hyperplanes. Vapnik (1998, p. 413) states that the following bound holds with respect to the VC dimension  $h$  of a separating hyperplane defined by the normal vector  $\mathbf{w}$  in a subset  $\mathbf{X}^* \subset \mathbb{R}^n$ . Assume that  $\|\mathbf{x}\| < \lambda$  for all  $\mathbf{x} \in \mathbf{X}^*$  (making  $\mathbf{X}^*$  a closed ball in  $\mathbb{R}^n$ ) and  $\|\mathbf{w}\| \leq \nu$ . Then

$$h \leq \min([\lambda^2 \nu^2], n) + 1 \quad (6.31)$$

To minimize the complexity, as measured by the VC dimension, of the classification model induced by the hyperplane defined by  $\mathbf{w}$ , the task is therefore to minimize  $\|\mathbf{w}\|$ . It directly follows from (6.27) that the model complexity is *minimized* when the margin between the vector sets  $\mathbf{X}^+$  and  $\mathbf{X}^-$  is *maximized*. This is the rationale underlying the characterization of SVMs as *maximal-margin classifiers* (Shawe-Taylor & Cristianini, 2004, p. 212). We will now proceed to derive an expression for the computational problem involved in finding the maximally separating hyperplane.

#### 6.4.1 The SVM optimization problem in the primal form

Since the margin between the vector sets is maximized when  $\|\mathbf{w}\|$  is minimized the optimization problem at hand can equivalently be expressed in terms of the squared norm  $\|\mathbf{w}\|^2 = \mathbf{w} \cdot \mathbf{w}$ . We seek to

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & g_i(\mathbf{w}, b) = y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0 \quad i \in \{1, \dots, \ell\} \end{aligned} \tag{6.32}$$

The formulation of the SVM optimization problem in (6.32) is called its *primal form* (Joachims, 2002, p. 37). We observe that the objective function  $f$  has a quadratic form in the single variable  $\mathbf{w}$ , which means that (6.32) is a *quadratic program*. The fraction  $\frac{1}{2}$  in the definition of  $f$  is inserted to make the coefficient vanish from the gradient  $\nabla f$ . To proceed we need to formulate the *Lagrangian function* (see section

6.3.2)

$$\begin{aligned}
\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) &= f(\mathbf{w}) - \sum_{i=1}^{\ell} \alpha_i g_i(\mathbf{w}, b) \\
&= \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{\ell} \alpha_i y_i (\mathbf{w} \cdot \mathbf{x}_i + b) + \sum_{i=1}^{\ell} \alpha_i
\end{aligned} \tag{6.33}$$

where the values in the coefficient vector  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_\ell)$  are Lagrangian multipliers. If the constraints of the SVM optimization problem would consist only of equalities the condition  $\nabla \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{0}$  would be sufficient to guarantee that a minimum of  $f$  has been found. However, since the constraints are inequalities we need additional theorems to state conditions for optimality. We will first perform an analysis of the primal form of the SVM optimization problem with the Karush-Kuhn-Tucker conditions and reformulate the problem in the *Wolfe dual* form (see e.g. Joachims, 2002, p. 38).

#### 6.4.2 The primal form and the KKT conditions

The Karush-Kuhn-Tucker (KKT) conditions (see section 6.3.3) state the *necessary*, and under certain circumstances *sufficient*, conditions for a point  $\mathbf{w}$  to be the solution to a nonlinear programming problem.

**Proposition 6.4.1.** *The objective function in (6.32) is convex.*

*Proof.* Since  $\partial f / \partial w_i = w_i$  is monotonically non-decreasing, it follows that  $f$  is convex in all dimensions of  $\mathbb{R}^n$ .  $\square$

Since  $f$  is convex and the constraints in (6.32) are linear the conditions are met for stating that a feasible point  $\tilde{\mathbf{w}}$  is a minimizer of (6.32) iff the KKT conditions are satisfied at  $\tilde{\mathbf{w}}$ , i.e. the KKT conditions are sufficient for optimality. The KKT conditions applied to the problem

in (6.32) are listed below.

$$\begin{aligned} \nabla_{\mathbf{w}} \mathcal{L} = \mathbf{0} &\implies \mathbf{w}^* - \sum_{i=1}^{\ell} \alpha_i^* y_i \mathbf{x}_i = \mathbf{0} \\ &\implies \mathbf{w}^* = \sum_{i=1}^{\ell} \alpha_i^* y_i \mathbf{x}_i \end{aligned} \quad (6.34)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \implies \sum_{i=1}^{\ell} \alpha_i^* y_i = 0 \quad (6.35)$$

For all  $i \in \{1, \dots, \ell\}$  it further holds that

$$\alpha_i^* \geq 0 \quad (6.36)$$

$$g_i(\mathbf{w}^*, b^*) \geq 0 \implies y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1 \geq 0 \quad (6.37)$$

$$\alpha_i^* g_i(\mathbf{w}^*, b^*) = 0 \implies \alpha_i^* (y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1) = 0 \quad (6.38)$$

The condition on the Lagrangian (eq. 6.34) shows that the vector  $\mathbf{w}^*$  is a *linear combination* of the points in the training set. Further, if there is a point  $\mathbf{x}_i$  such that  $\alpha_i^* > 0$  at the solution  $(\mathbf{w}^*, b^*)$ , then it follows directly from the KKT condition (6.37) that the constraint  $g_i$  is *active*, i.e.  $g_i(\mathbf{w}^*, b^*) = 0$ . That is,  $\mathbf{x}_i$  is located in one of the margin hyperplanes. We conclude that  $\mathbf{w}^*$  is in fact a linear combination of precisely those points  $\mathbf{x}_i$ , called *support vectors*, for which  $\alpha_i^* > 0$  (cf. the encircled points in figure 6.1).

Stated differently, assuming that the linearity condition in (6.35) holds for a configuration  $\alpha$  it follows from conditions (6.37) and (6.38), and the fact that the KKT conditions are sufficient for opti-

mality, that a configuration  $(\mathbf{w}, b, \boldsymbol{\alpha})$  is a minimizer of (6.32) iff for every data point  $\mathbf{x}_i$  and the corresponding multiplier  $\alpha_i$  it holds that

$$\begin{aligned} \text{if } \alpha_i = 0 & \text{ then } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \\ \text{if } \alpha_i > 0 & \text{ then } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 \end{aligned} \quad (6.39)$$

### 6.4.3 The optimization problem in the dual form

We will now look at a dual formulation of the optimization problem, which is equivalent to the problem stated in (6.32), in the sense that the ordered pair  $(\mathbf{w}^*, b^*)$  is a solution to problem in the primal form iff it is a solution to the problem in the dual form. Since the objective function  $f$  is convex and the inequality constraints in (6.32) are linear we can formulate the optimization problem in a dual form, called the *Wolfe dual* (see section 6.3.1). In this form the objective is to *maximize*  $\mathcal{L}$  subject to certain constraints. More specifically, the dual optimization problem is to

$$\begin{aligned} \text{maximize } \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{\ell} \alpha_i y_i (\mathbf{w} \cdot \mathbf{x}_i + b) + \sum_{i=1}^{\ell} \alpha_i \\ \text{subject to } \nabla \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) &= \mathbf{0} \\ \alpha_i &\geq 0 \quad \text{for all } i \in \{1, \dots, \ell\} \end{aligned} \quad (6.40)$$

Since the constraint on the Lagrangian is the same as the KKT condition (6.34) we again obtain the following relations.

$$\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}^*, b^*, \boldsymbol{\alpha}^*) = \mathbf{0} \quad \implies \quad \mathbf{w}^* = \sum_{i=1}^{\ell} \alpha_i^* y_i \mathbf{x}_i \quad (6.41)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \implies \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \quad (6.42)$$

By substituting the right-hand side expressions in (6.41) and (6.42) into (6.40) and reformulating (6.40) into a minimization problem we obtain a new objective function, defined on  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_\ell)$ .

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i=1}^{\ell} \alpha_i \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\ & \alpha_i \geq 0 \quad \text{for all } i \in \{1, \dots, \ell\} \end{aligned} \quad (6.43)$$

We can further transform (6.43) into the canonical form for a quadratic program presented in (6.2). Let  $\mathbf{Q}$  be a symmetric matrix with the definition

$$\mathbf{Q} := \left[ y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \right]_{i,j \in \{1, \dots, \ell\}} \quad (6.44)$$

Since the second order partial derivative of the objective function  $f$  in 6.43 with respect to  $\alpha_i$  and  $\alpha_j$  is

$$\frac{\partial^2 f}{\partial \alpha_i \partial \alpha_j} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (6.45)$$

we find that  $2\mathbf{Q}$  is the *Hessian matrix* (the complete matrix of second-order partial derivatives) of  $f$ . Further, let  $\mathbf{c} = (-1, \dots, -1)^\top$ ,  $\mathbf{I}$  the  $\ell \times \ell$  *identity matrix* and  $\mathbf{e}_i$  the  $i$ th column vector of  $\mathbf{I}$ . Finally, let  $\mathbf{y} = (y_1, \dots, y_\ell)$ . Using this notation we restate the optimization

problem in (6.43) as follows.

$$\begin{aligned}
 \text{minimize} \quad & f(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{G} \boldsymbol{\alpha} + \mathbf{c}^\top \boldsymbol{\alpha} \\
 \text{subject to} \quad & \mathbf{y}^\top \boldsymbol{\alpha} = 0 \\
 & \mathbf{e}_i^\top \boldsymbol{\alpha} \geq 0 \quad \text{for all } i \in \{1, \dots, \ell\}
 \end{aligned} \tag{6.46}$$

## 6.5 Soft-margin SVM

For many classification problems the data points are not as conveniently linearly separable as in the ideal case of the original formulation presented above. An approach that has been suggested for handling non-separable data is to relax the strict linearity of the separating hyperplane by introducing *slack variables*, one per training point, in the constraints of the primal form (6.30) (Vapnik, 1998, p. 411):

$$\begin{aligned}
 y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1 - \xi_i \quad \text{for all } i \in \{1, \dots, \ell\} \\
 \xi_i &\geq 0 \quad \text{for all } i \in \{1, \dots, \ell\}
 \end{aligned} \tag{6.47}$$

### 6.5.1 C-SVM

In Cortes & Vapnik (1995) the authors propose that the influence of the slack variables on the overall constraints should be regulated by a non-negative constant  $C$ . This approach is a direct application of the principles of *structural risk minimization* and *regularization*, which we presented in section 5.6.3. A higher value of  $C$  will allow a higher number of violations of the hyperplane separation (i.e. allow a higher number of data points to be located at a side of the hyperplane that does not correspond to its category label) and yield a lower *empirical risk* (see section 5.6.1) at the cost of a more complex classification

model, whereas a lower value will impose a stricter margin and therefore more training errors will occur (Joachims, 2002, p. 40). The regularized risk functional is formulated as follows. Let  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_\ell)$ . The quadratic programming problem has the formulation

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{\ell} \xi_i \\ \text{subject to} \quad & g_i(\mathbf{w}, b, \boldsymbol{\xi}) = y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \geq 0 \\ & \xi_i \geq 0 \\ \text{for all} \quad & i \in \{1, \dots, \ell\} \end{aligned} \quad (6.48)$$

The corresponding Lagrangian function for the objective function and the constraints in (6.48) is given by

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \\ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{\ell} \xi_i - \sum_{i=1}^{\ell} (\alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i)) - \sum_{i=1}^{\ell} \beta_i \xi_i \end{aligned} \quad (6.49)$$

We notice that the slack variables in  $\boldsymbol{\xi}$  obtain a separate collection  $\beta_1, \dots, \beta_\ell$  of Lagrangian multipliers to enforce the condition  $\xi_i \geq 0$ . The KKT conditions for the modified optimization problem in (6.48) are presented below. As a shorthand for the quite lengthy expression

$\mathcal{L}(\mathbf{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$  we will only write  $\mathcal{L}$  in the KKT equations.

$$\begin{aligned} \nabla_{\mathbf{w}} \mathcal{L} = \mathbf{0} &\implies \mathbf{w}^* - \sum_{i=1}^{\ell} \alpha_i^* y_i \mathbf{x}_i = \mathbf{0} \\ &\implies \mathbf{w}^* = \sum_{i=1}^{\ell} \alpha_i^* y_i \mathbf{x}_i \end{aligned} \quad (6.50)$$

We find out that the inclusion of slack variables has not changed the formulation of the solution vector  $\mathbf{w}^*$ , as compared to the hard-margin case.

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \implies \sum_{i=1}^{\ell} \alpha_i^* y_i = 0 \quad (6.51)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \implies \alpha_i^* + \beta_i^* = C \quad \text{for all } i \in \{1, \dots, \ell\} \quad (6.52)$$

For all  $i \in \{1, \dots, \ell\}$  it further holds that

$$\alpha_i^* \geq 0 \quad (6.53)$$

$$\beta_i^* \geq 0 \quad (6.54)$$

$$g_i(\mathbf{w}^*, b^*, \xi_i^*) \geq 0 \implies y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1 + \xi_i^* \geq 0 \quad (6.55)$$

$$\alpha_i^* g_i(\mathbf{w}^*, b^*, \xi_i^*) = 0 \implies \alpha_i^* (y_i(\mathbf{w}^* \cdot \mathbf{x}_i + b^*) - 1 + \xi_i^*) = 0 \quad (6.56)$$

$$\beta_i^* h_i(\xi_i^*) = 0 \quad \implies \quad \beta_i^* \xi_i^* = 0 \quad (6.57)$$

From condition (6.52) we learn that the Lagrangian multipliers  $\alpha_1, \dots, \alpha_\ell$  are now both lower and upper bounded at the solution. In addition to the constraint  $\alpha_i \geq 0$  we also see, due to the fact that both  $\beta_i$  and  $C$  are non-negative, that  $\alpha_i \leq C$  for all  $i \in \{1, \dots, \ell\}$ . Support vectors  $\mathbf{x}_i$  for which  $\alpha_i = C$  are commonly called *bounded* support vectors (Joachims, 2002, p. 40).

The Wolfe dual of (6.48) has the formulation

$$\begin{aligned} &\text{maximize } \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) && \text{subject to} \\ &\nabla \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = 0 && (6.58) \\ &\alpha_i \geq 0, \beta_i \geq 0 && \text{for all } i \in \{1, \dots, \ell\} \end{aligned}$$

Taking the gradient of  $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$  with respect to  $\mathbf{w}$  yields the same objective function as in (6.43), but the optimization problem will now have stricter bounds on the Lagrangian multipliers.

$$\begin{aligned} &\text{minimize } \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i=1}^{\ell} \alpha_i \\ &\text{subject to } \sum_{i=1}^{\ell} \alpha_i y_i = 0 && (6.59) \\ &0 \leq \alpha_i \leq C && \text{for all } i \in \{1, \dots, \ell\} \end{aligned}$$

The KKT condition (6.52) stipulates that  $\beta_i > 0$  iff  $\alpha_i < C$ . Since the condition (6.57) requires that  $\xi_i = 0$  if  $\beta_i > 0$  it follows that  $\xi_i = 0$  if  $\alpha_i < C$ . We conclude that  $\xi_i$  will vanish from condition (6.56) for all *unbounded* support vectors ( $\alpha_i < C$ ). Given that  $\mathbf{w}^*$  is computed, the bias term  $b^*$  at the solution can therefore be calculated by insertion of

$\mathbf{w}^*$  into (6.56) and computing the average of  $b^*$  over all unbounded support vectors.

In analogy with how we formulated the optimality conditions for hard-margin SVMs we now state the corresponding conditions for soft-margin SVMs. Assuming that the linearity condition in (6.51) holds for a configuration  $\alpha$  it follows from the KKT conditions that a configuration  $(\mathbf{w}, b, \alpha)$  is a minimizer of (6.59) iff for every data point  $\mathbf{x}_i$  and the corresponding multiplier  $\alpha_i$  it holds that

$$\begin{aligned} \text{if } \alpha_i = 0 & \quad \text{then } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \\ \text{if } 0 < \alpha_i < C & \quad \text{then } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 \\ \text{if } \alpha_i = C & \quad \text{then } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \leq 1 \end{aligned} \tag{6.60}$$

### 6.5.2 $\nu$ -SVM

An alternative formulation of a soft-margin classifier using slack variables, called  $\nu$ -SVM (pronounced "nu-SVM"), was proposed by Schölkopf et al. (2000). One important difference between the soft-margin classifier described by Cortes & Vapnik (C-SVM) and  $\nu$ -SVM is that the latter formulation does not use a regularization constant  $C$  controlling the trade-off between training errors and the number of support vectors generated. Instead there is a constant  $\nu$  introduced, stipulating an upper bound on the number of hyperplane violations and a lower bound on the number of support vectors generated (Schölkopf et al., 2000, p. 1226). The  $\nu$ -SVM classifier is posed as the following

optimization problem:

$$\begin{aligned}
 \text{minimize} \quad & f(\mathbf{w}, \boldsymbol{\xi}, \rho) = \frac{1}{2} \|\mathbf{w}\|^2 - \nu \rho + \frac{1}{\ell} \sum_{i=1}^{\ell} \xi_i \\
 \text{subject to} \quad & g_i(\mathbf{w}, b, \boldsymbol{\xi}, \rho) = y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - \rho + \xi_i \geq 0 \\
 & \xi_i \geq 0 \\
 & \rho \geq 0 \\
 \text{for all} \quad & i \in \{1, \dots, \ell\}
 \end{aligned} \tag{6.61}$$

The corresponding Lagrangian function is given by

$$\mathcal{L} = \frac{1}{2} \|\mathbf{w}\|^2 - \nu \rho + \frac{1}{\ell} \sum_{i=1}^{\ell} \xi_i - \sum_{i=1}^{\ell} (\alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - \rho + \xi_i)) - \sum_{i=1}^{\ell} \beta_i \xi_i - \sum_{i=1}^{\ell} \gamma_i \rho \tag{6.62}$$

By rearranging the terms in (6.62) we get

$$\mathcal{L} = \frac{1}{2} \|\mathbf{w}\|^2 - \nu \rho + \frac{1}{\ell} \sum_{i=1}^{\ell} \xi_i - \sum_{i=1}^{\ell} \alpha_i y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - \sum_{i=1}^{\ell} (\alpha_i + \beta_i) \xi_i + \sum_{i=1}^{\ell} (\alpha_i - \gamma_i) \rho \tag{6.63}$$

Taking the partial derivative of  $\mathcal{L}$  with respect to  $\mathbf{w}$ ,  $b$ ,  $\boldsymbol{\xi}$ , and  $\rho$  respectively under the constraint  $\nabla \mathcal{L} = \mathbf{0}$  produces the following set

of equalities and bounds on the Lagrangian multipliers:

$$\begin{aligned}
 \mathbf{w} &= \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i \\
 \sum_{i=1}^{\ell} \alpha_i y_i &= 0 \\
 \alpha_i + \beta_i &= \frac{1}{\ell} \\
 \sum_{i=1}^{\ell} \alpha_i - \gamma &= \nu
 \end{aligned} \tag{6.64}$$

By substituting these relations into (6.63) the Wolfe dual form obtains the formulation

$$\begin{aligned}
 \text{minimize} \quad & \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \mathbf{x}_j \cdot \mathbf{x}_k \\
 \text{subject to} \quad & 0 \leq \alpha_i \leq \frac{1}{\ell} \quad \text{for all } i \in \{1, \dots, \ell\} \\
 & \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\
 & \sum_{i=1}^{\ell} \alpha_i \geq \nu
 \end{aligned} \tag{6.65}$$

## 6.6 Kernel methods for SVM

We have so far seen (e.g. equation 6.34) that the vector  $\mathbf{w}$  determining the orientation of the separating hyperplane of the SVM classifier is a linear combination of the support vectors. In other words, the *decision boundary* between the categories in the feature space is linear. A powerful construct in the design of SVMs for dealing with problems where the categories are not readily linearly separable is a mapping of the feature space called the *kernel trick*. The fundamental idea of this

procedure is to (implicitly) map the feature vectors from the original feature space  $\mathcal{F}$  (which is typically a Euclidean space) into a higher-dimensional space  $\tilde{\mathcal{F}}$  (which may be any Hilbert space) and compute the dot product of the feature vectors in  $\tilde{\mathcal{F}}$  instead of  $\mathcal{F}$ . This will typically yield a *nonlinear* decision boundary in  $\mathcal{F}$  (Schölkopf & Smola, 2002, p. 15).

### 6.6.1 Kernels

The term *kernel* has several distinct definitions and domains of use in mathematics. Purportedly, the term is borrowed from the class of functions constituting the core of *integral transforms* and which are in focus in *Mercer's theorem* (section 6.6.5 below). The definition used below is tailored for the use in machine learning and appears in e.g. Schölkopf & Smola (2002, p. 2, 3) and Shawe-Taylor & Cristianini (2004, p. 34). Let  $S$  be a set and  $\tilde{\mathcal{F}}$  an inner product space with inner product  $\langle \cdot, \cdot \rangle_{\tilde{\mathcal{F}}}$ . As we will see below,  $\tilde{\mathcal{F}}$  is assumed to be a *Hilbert space*, i.e. a vector space that is *complete* with respect to the norm induced by the inner product of the space. We also assume a total map  $\phi : S \rightarrow \tilde{\mathcal{F}}$ . A *kernel* is a function  $\kappa : S \times S \rightarrow \mathbb{R}$  such that

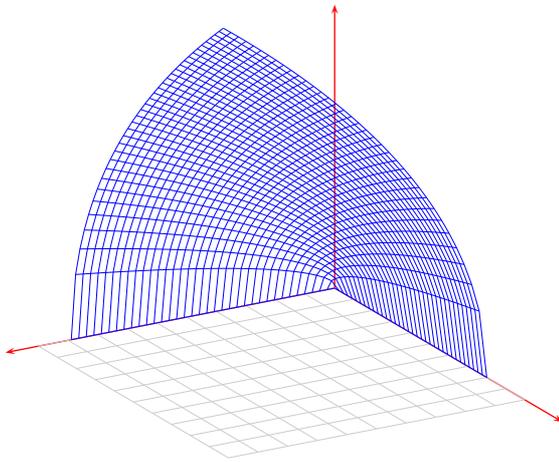
$$\kappa(x, z) = \langle \phi(x), \phi(z) \rangle_{\tilde{\mathcal{F}}} \quad (6.66)$$

It follows from the symmetry of the inner product that also  $\kappa$  has to be symmetric, i.e.  $\kappa(x, z) = \kappa(z, x)$  for all  $x, z \in S$ . Strictly speaking the definition in (6.66) applies to any set  $X$ , but in the treatment that follows, and in congruence with the rest of this chapter, we assume  $S$  to be a Euclidean vector space.

As an example, consider a map  $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$  with the definition

$$(x_1, x_2) \mapsto (x_1^2, x_2^2, \sqrt{2}x_1x_2) \quad (6.67)$$

The map  $\phi$  yields a surface displayed in figure 6.2. Let  $\mathbf{x} = (x_1, x_2)$



**Figure 6.2.** A plot of the map  $(x_1, x_2) \mapsto (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ .

and  $\mathbf{z} = (z_1, z_2)$  be vectors in  $\mathbb{R}^2$ . The dot product of the mapped vectors  $\phi(\mathbf{x})$  and  $\phi(\mathbf{z})$  is given by

$$\phi(\mathbf{x}) \cdot \phi(\mathbf{z}) = (x_1z_1)^2 + (x_2z_2)^2 + 2x_1x_2z_1z_2 = (\mathbf{x} \cdot \mathbf{z})^2 \quad (6.68)$$

It follows that  $\kappa(\mathbf{x}, \mathbf{z}) := (\mathbf{x} \cdot \mathbf{z})^2$  is a kernel with the corresponding map  $\phi$ .

### 6.6.2 The Riesz representation theorem

Let  $\mathcal{H}$  be a Hilbert space over a field  $\mathbb{K}$  and  $\mathcal{H}^*$  the dual space of all continuous linear functionals  $\mathcal{H} \rightarrow \mathbb{K}$ . The *Riesz representation theorem* (see e.g. Akhiezer & Glazman, 1993, p. 33) states that to each element  $f \in \mathcal{H}^*$  there exists a unique element  $\mathbf{f} \in \mathcal{H}$  such that  $f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{f} \rangle$  for every  $\mathbf{x} \in \mathcal{H}$ . In other words, all the information needed to evaluate  $f$  at every point is contained in  $\mathbf{f}$  and therefore  $\mathbf{f}$

adequately represents  $f$ . If we apply this theorem to the definition of weak topology we can state that the neighborhood of a point  $\mathbf{x} \in \mathcal{H}$  can be defined as the set  $\{\mathbf{y} \in \mathcal{H} : \langle \mathbf{x}, \mathbf{z} \rangle - \langle \mathbf{y}, \mathbf{z} \rangle < \delta \text{ for all } \mathbf{z} \in \mathcal{H}\}$  (Lee & Khargonekar, 2009).

**Example 6.6.1.** Consider the Euclidean space  $\mathbb{R}^3$ . Let  $\mathcal{F}$  denote the collection of continuous linear functionals  $\mathbb{R}^3 \rightarrow \mathbb{R}$ . Let  $f$  be any functional in  $\mathcal{F}$ , for instance  $f(\mathbf{x}) = 2x_1 + 3x_2 - 5x_3$ , where  $\mathbf{x} = (x_1, x_2, x_3)$ . Let  $U$  be an open subset of  $\mathbb{R}$ , for instance the open interval  $(0, 1)$ . By the Riesz representation theorem we have that every element  $f \in \mathcal{F}$  can be written in terms of an element  $\mathbf{f} \in \mathbb{R}^3$  such that  $f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{f} \rangle$  for all  $\mathbf{x} \in \mathbb{R}^3$ . Then the preimage  $f^{-1}(U)$  is the set of points  $\mathbf{x} \in \mathbb{R}^3$  such that  $f(\mathbf{x}) \in (0, 1)$ , or equivalently  $\langle \mathbf{x}, \mathbf{f} \rangle \in (0, 1)$ . If we assume the above definition of  $f$ , i.e.  $f(\mathbf{x}) = 2x_1 + 3x_2 - 5x_3$ , then clearly  $\mathbf{f} = (2, 3, -5)$ .

### 6.6.3 Reproducing kernel Hilbert space

Let  $\mathbf{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_\ell\}$  be a set of vectors in  $\mathbb{R}^n$ . Further, let  $\kappa : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$  be a kernel. For each element  $\mathbf{x}_i \in \mathbf{S}$  we define a corresponding function  $\psi_i^\kappa(\cdot) := \kappa(\mathbf{x}_i, \cdot)$ . Let the collection of functions  $\{\psi_i^\kappa\}_i$  so defined be *basis functions* of a vector space  $\mathcal{H}_\kappa$ . Hence,  $\mathcal{H}_\kappa$  is the space of all linear combinations over  $\{\psi_i^\kappa\}_i$ , i.e.

$$\mathcal{H}_\kappa(\mathbf{S}) = \left\{ \sum_{i=1}^{\ell} \alpha_i \psi_i^\kappa(\cdot) \right\}$$

Let  $f$  and  $g$  be functions defined  $f(\sigma) := \sum_{i=1}^r \alpha_i \psi_\sigma^\kappa(\mathbf{x}_i)$  and  $g(\sigma) := \sum_{j=1}^s \beta_j \psi_\sigma^\kappa(\mathbf{x}_j)$ . Following Shawe-Taylor & Cristianini (2004, p. 62)

we define an inner product  $\langle \cdot, \cdot \rangle$  on the elements in  $\mathcal{H}_\kappa(\mathbf{S})$  as

$$\langle f, g \rangle := \sum_{i=1}^r \sum_{j=1}^s \alpha_i \beta_j \kappa(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i=1}^r \sum_{j=1}^s \alpha_i \beta_j \psi_i^\kappa(\mathbf{x}_j) = \sum_{i=1}^r \alpha_i g(\mathbf{x}_i) \quad (6.69)$$

It follows from the definition in (6.69) that

$$\langle f, \psi_\sigma^\kappa(\cdot) \rangle = \sum_{i=1}^r \alpha_i \psi_\sigma^\kappa(\mathbf{x}_i) = f(\sigma) \quad (6.70)$$

Shawe-Taylor & Cristianini (2004, p. 62f) demonstrate that  $\mathcal{H}_\kappa(\mathbf{S})$  is a *Hilbert space* by ascertaining that given any Cauchy sequence of functions  $f_1, f_2, \dots \in \mathcal{H}_\kappa(\mathbf{S})$ , defined so that for each  $\varepsilon > 0$  there exists an  $\omega \in \mathbb{N}$  such that  $\|f_m(\sigma) - f_n(\sigma)\| < \varepsilon$  for each  $m, n > \omega$ , the limit  $g(\sigma) = \lim_{n \rightarrow \infty} f_n(\sigma)$  is also an element in  $\mathcal{H}_\kappa(\mathbf{S})$ . Because of the property in (6.70) the space  $\mathcal{H}_\kappa(\mathbf{S})$  is called a *reproducing kernel Hilbert space* (RKHS), since to each element  $f \in \mathcal{H}_\kappa(\mathbf{S})$  and every point  $\mathbf{x}_i$  there exists a function  $\psi_i^\kappa \in \mathcal{H}_\kappa$  such that the inner product  $\langle f, \psi_i^\kappa \rangle$  evaluates  $f$  at  $\mathbf{x}_i$ . Further, the *Moore-Aronszajn theorem* states that each symmetric, positive definite kernel  $\kappa$  induces a *unique* RKHS (Aronszajn, 1950, p. 344; Schölkopf et al., 1999, p. 71).

Consider the *dual* space  $\mathcal{H}_\kappa^*(\mathbf{S})$  of linear functionals  $\mathcal{H}_\kappa(\mathbf{S}) \rightarrow \mathbb{R}$ . Further, let  $h_\sigma \in \mathcal{H}_\kappa^*(\mathbf{S})$  be a pointwise evaluation function with the definition  $f \mapsto f(\sigma)$ . Then according to (6.70) it follows that there exists an element  $\psi_\sigma^\kappa \in \mathcal{H}_\kappa(\mathbf{S})$  such that  $h_\sigma(f) = \langle f, \psi_\sigma^\kappa \rangle$ . This result is also congruent with the *Riesz representation theorem* (see section 6.6.2 above) which states that to each linear functional  $h : \mathcal{H} \rightarrow \mathbb{F}$ , where  $\mathcal{H}$  is a Hilbert space over the field  $\mathbb{F}$ , there exists an element  $\mathbf{z} \in \mathcal{H}$  such that  $\langle \mathbf{z}, \cdot \rangle$  evaluates  $h$  at every point in  $\mathcal{H}$ .

### 6.6.4 The kernel trick

In section 6.6.1 we defined a kernel  $\kappa$  as a function computing the inner product between vectors mapped into a Hilbert space  $\tilde{\mathcal{F}}$  by a function  $\phi$ , i.e.

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle_{\tilde{\mathcal{F}}}$$

Assuming that  $\tilde{\mathcal{F}}_{\kappa}$  is a RKHS defined on a kernel  $\kappa$ , letting  $\psi_i^{\kappa} := \kappa(\mathbf{x}_i, \cdot)$  and  $\psi_j^{\kappa} := \kappa(\mathbf{x}_j, \cdot)$  we find, using the definition in (6.69), that

$$\langle \psi_i^{\kappa}, \psi_j^{\kappa} \rangle = \kappa(\mathbf{x}_i, \mathbf{x}_j) \quad (6.71)$$

In other words, by letting  $\phi : \mathbb{R}^n \rightarrow \mathcal{H}_{\kappa}$  with the definition  $\mathbf{x} \mapsto \kappa(\mathbf{x}, \cdot)$  we have generated the complete framework for performing the kernel trick. It should be noted that although the map  $\phi$  is unfeasible to compute directly (which is often the case) it is still possible to access the inner product in the RKHS. The ultimate point of the kernel trick is that the kernel can be substituted for the dot product in the formulation of the optimization problems in for instance (6.43) and (6.59), which in practical terms means that it is possible to transform a linear into a nonlinear classification problem. Reformulating (6.59) using a kernel  $\kappa$  we get

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{\ell} \alpha_i \\ \text{subject to} \quad & \sum_{i=1}^{\ell} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \quad \text{for all } i \in \{1, \dots, \ell\} \end{aligned} \quad (6.72)$$

When kernels are used to induce a separating hyperplane the definition of the classifier in (6.21) obtains a slightly different expression. We recall from equation (6.41) that  $\mathbf{w}$  is a linear combination of the support vectors:

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i$$

The classifier defined in (6.21) therefore has the expansion

$$\psi(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b) = \text{sgn} \left( \sum_{i=1}^{\ell} (\alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}) + b \right) \quad (6.73)$$

When using kernels the vector  $\mathbf{w}$  defining the separating hyperplane is replaced by a function in the RKHS defined on the set  $\mathbf{X}$  of training vectors. Let  $\mathcal{H}_\kappa(\mathbf{X})$  be the Hilbert space

$$\mathcal{H}_\kappa(\mathbf{X}) := \left\{ \sum_{i=1}^{\ell} \lambda_i \kappa(\mathbf{x}_i, \cdot) \right\} \quad \lambda_i \in \mathbb{R}$$

and  $w(\mathbf{x})$  an element in  $\mathcal{H}_\kappa(\mathbf{X})$  with the definition

$$w(\mathbf{x}) := \sum_{i=1}^{\ell} \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) \quad (6.74)$$

where  $\alpha_i$  and  $y_i$  have the same denotations as previously in this chapter. We now redefine the SVM classifier in (6.73) using  $w(\mathbf{x})$ :

$$\psi(\mathbf{x}) = \text{sgn}(w(\mathbf{x}) + b) \quad (6.75)$$

It should be observed that the function  $w$  in practical terms will only be defined over the support vectors (i.e. the vectors  $\mathbf{x}_i$  for which

$\alpha_i > 0$ ), and that the linear classifier formulated in (6.73) is a special case of (6.75) with  $\kappa(\mathbf{x}_i, \mathbf{x}) := \mathbf{x}_i \cdot \mathbf{x}$ . Hence, the eventual SVM classifier is not a function of all training points  $(\mathbf{x}_i, y_i)$  but only the set of support vectors.

The corresponding KKT conditions for optimality (cf. eq. 6.60) are given by

$$\begin{aligned} \text{if } \alpha_i = 0 & \quad \text{then } y_i \left( \sum_{j=1}^{\ell} \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) + b \right) \geq 1 \\ \text{if } 0 < \alpha_i < C & \quad \text{then } y_i \left( \sum_{j=1}^{\ell} \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) + b \right) = 1 \\ \text{if } \alpha_i = C & \quad \text{then } y_i \left( \sum_{j=1}^{\ell} \alpha_j y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) + b \right) \leq 1 \end{aligned} \quad (6.76)$$

Any function of the form stated in equation (6.74) is called a *spline* function. The *representer theorem* first formulated in (Kimeldorf & Wahba, 1971) and subsequently adapted for machine learning (see e.g. Schölkopf & Smola, 2002, p. 90) states that given a RKHS  $\mathcal{H}_\kappa$  and a set  $\{y_i\}_{i=1}^{\ell}$  of constants, if a function  $f \in \mathcal{H}_\kappa$  minimizes the expression

$$\sum_{i=1}^{\ell} \sum_{j=1}^{\ell} (N_i \circ f - y_i)(N_j \circ f - y_j) + \int (Lf)^2 d\mu \quad (6.77)$$

where  $\{N_i\}_{i=1}^{\ell}$  are linear functionals and  $L$  a linear differential operator, then  $f$  is a spline function in  $\mathcal{H}_\kappa$ , i.e.

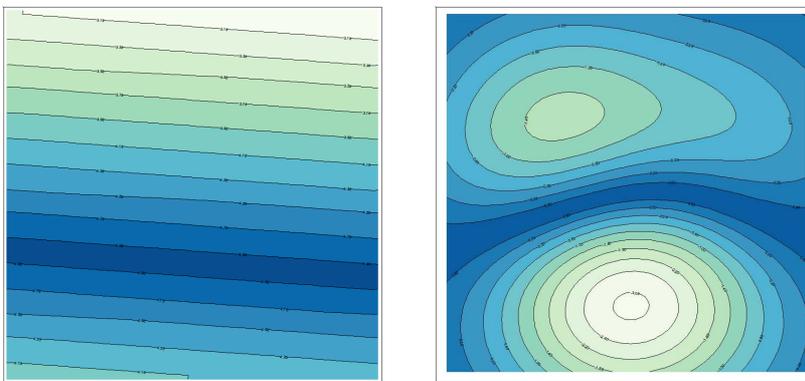
$$f = \sum_{i=1}^{\ell} \alpha_i \kappa(\mathbf{x}_i, \cdot) \quad \alpha_i \in \mathbb{R} \quad (6.78)$$

It can be shown that the regularized risk functional for the SVM optimization problem in equation (5.48) can be written on the form given in (6.77).

The most commonly used non-linear kernels for SVMs are the following (Joachims, 2002, p. 42):

Kernel	Definition
Polynomial	$\kappa(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z} + 1)^d$
Radial basis function	$\kappa(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \ \mathbf{x} - \mathbf{z}\ ^2)$
Sigmoid	$\kappa(\mathbf{x}, \mathbf{z}) = \tanh(a\mathbf{x} \cdot \mathbf{z} + b)$

As a visual example, by plotting the decision surface of the standard linear kernel and a radial basis kernel respectively we can observe that the latter generates a non-linear separation between the categories (figure 6.3).



**Figure 6.3.** Two-dimensional contour plots of the decision surfaces of the linear and a radial basis kernel respectively.

### 6.6.5 Mercer's theorem

In the formulation of the kernel trick above we have assumed the existence of a semipositive definite kernel. For practical reasons it is natural to ask precisely under which conditions a function is a kernel. This is obtained through a condition known as *Mercer's theorem*

(Shawe-Taylor & Cristianini, 2004, p. 65). Let  $\mathbf{S}$  be a compact subset of  $\mathbb{R}^n$  and  $(\mathbf{S}, \mathfrak{A}(\mathbf{S}), \mu)$  a measure space defined over  $\mathbf{S}$ . A *square-integrable* function on  $(\mathbf{S}, \mathfrak{A}(\mathbf{S}), \mu)$  is a function  $f$  such that

$$\int_{\mathbf{S}} |f(\mathbf{x})|^2 d\mu(\mathbf{x}) < \infty$$

By  $\mathcal{L}^2_{\mu}(\mathbf{S})$  we denote the Hilbert space of square-integrable functions defined on  $\mathbf{S}$ . Further, an *integral transform*  $T_{\kappa} : \mathcal{L}^2(\mathbf{S}) \rightarrow \mathcal{L}^2(\mathbf{S})$  of a function  $f$  over a set  $\mathbf{S}$  is defined

$$T_{\kappa}f(\mathbf{x}) := \int_{\mathbf{S}} \kappa(\mathbf{x}, \mathbf{z})f(\mathbf{z}) d\mu(\mathbf{z})$$

One of the most commonly applied integral transforms in data analysis is the *Fourier transform*. The function  $\kappa$  is called the *kernel* of the integral transform. Further,  $\kappa$  is said to be *nonnegative definite* (or *positive semidefinite*) if

$$\int_{\mathbf{S} \times \mathbf{S}} \kappa(\mathbf{x}, \mathbf{z})f(\mathbf{x})f(\mathbf{z}) d\mu(\mathbf{x}) d\mu(\mathbf{z}) \geq 0$$

for all choices of  $f$ . *Mercer's theorem* states that a continuous, symmetric, positive semidefinite kernel  $\kappa$  can be expanded in terms of a set of orthonormal basis functions  $\{\phi_i\}_i$  in  $\mathcal{L}^2_{\mu}(\mathbf{S})$ , according to

$$\kappa(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{\infty} \phi_i(\mathbf{x})\phi_i(\mathbf{z}) < \infty \quad (6.79)$$

which is a dot product in the Hilbert space  $\ell^2$  of square-summable sequences. This theorem is basically an application of the *generalized Fourier series*. Let  $\kappa$  be a kernel and  $\Phi = \{\phi_i\}_i$  the orthonormal basis of the reproducing kernel Hilbert space  $\mathcal{F}$  defined on  $\kappa$ . Further, let  $\langle \cdot, \cdot \rangle$  denote the inner product in  $\mathcal{F}$ . It is easily verified that since by

assumption  $\Phi$  is the basis of  $\mathcal{F}$  it holds that (see also Shawe-Taylor & Cristianini, 2004, p. 52)

$$\sum_{\phi_i \in \Phi} \langle \phi_i, \psi_\sigma \rangle \phi_i(\mathbf{z}) = \psi_\sigma(\mathbf{z}) \quad (6.80)$$

for all  $\psi \in \mathcal{F}$ . It follows that a kernel  $\kappa$  can be expanded according to

$$\kappa(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{\infty} \langle \phi_i(\cdot), \kappa(\mathbf{x}, \cdot) \rangle \phi_i(\mathbf{z}) = \sum_{i=1}^{\infty} \phi_i(\mathbf{x}) \phi_i(\mathbf{z}) \quad (6.81)$$

$\kappa(\mathbf{x}, \mathbf{z})$  converges to a finite value since  $\mathbf{S}$  is assumed to be compact (for a proof see Shawe-Taylor & Cristianini, 2004, p. 65). In summary, Mercer's theorem states that any binary function that is continuous, symmetric, and positive semidefinite is also a kernel in the sense defined in section 6.6.1. Functions that can be shown to satisfy these conditions are called *Mercer kernels*.

## **Part III**

# **Experiments with semantic kernels**

## Chapter 7

# Semantic kernels

As we noted in section 5.4, a restriction inherent in the classical vector space model is that the term vectors forming a basis of the document representation space are stipulated to be pairwise orthogonal, effectively meaning that the terms are regarded as semantically independent of each other (Baeza-Yates & Ribeiro-Neto, 2011, p. 98). One obvious drawback of this model is that the computational similarity between documents for classification (or between documents and queries in an information retrieval setting) is performed on a term level, rather than on a conceptual level. For instance, such a model does not take into account that different terms may belong to the same semantic scope by means of synonymy relations, hyponymy relations and so on.

Wong et al. (1985) proposed a generalization of the vector space model for information retrieval, called the *generalized vector space model*, by introducing term-term correlation values into the vector similarity calculations. The idea behind the use of *semantic kernels* (also called semantic *smoothing* kernels) for SVM classification is likewise to facilitate similarity computations that include information

about semantic similarity between terms. To illustrate the intuition behind this approach, consider two documents – one containing the word *journal* and the other document containing the word *magazine* (but none of the documents containing both these terms). These two words could be described as near synonyms since their senses overlap, yet having mutually exclusive senses as well. In the classical vector space model the relationship between the documents induced by the presence of these terms is not manifest in the document vectors since the terms are distinct from each other.

$$\mathbf{d}_j = (2, 0)$$

$$\mathbf{d}_k = (0, 3)$$

The dot product between  $\mathbf{d}_j$  and  $\mathbf{d}_k$  is 0. In other words: if the documents would be computationally matched on the basis of the words *journal* and *magazine*, the similarity (as measured by the cosine of the angle between the document vectors) would be 0, despite the semantic relationship between these two words.

A semantic kernel is, analogously to the generalized vector space model, a classification kernel for which the orthogonality assumption has been discarded. In general terms, a semantic kernel consists of a square, symmetric matrix  $\mathbf{G}$  made up of pairwise similarity values between the terms in the vocabulary (Bloehdorn et al., 2006). Previous studies of semantic kernels have included term similarity values based on semantic relations contained in WordNet (Siolas & d'Alché Buc, 2000; Basili et al., 2006), Wikipedia (Wang & Domeniconi, 2008) as well as corpus-based statistics obtained by latent semantic analysis (Cristianini et al., 2002). In order to formally explain the theoretical justification for the use of semantic kernels and the document representation spaces induced by these kernels, we will proceed by exploring the vector space model in terms of basic *tensor algebra*. Tensor

algebra can be intuitively understood as a generalization of the concept of vectors (De et al., 2008, p. 1) and facilitates the study of vectors with respect to different vector space bases.

## 7.1 Document vectors and tensor calculus

Let  $\mathbf{x} = (x^1, x^2, \dots, x^n)$  be a vector of a vector space  $X$ . The components of  $\mathbf{x}$  are said to be *contravariant*, which in tensorial notation is shown by using upper indices (see e.g. De et al., 2008, p. 5). Since the vector  $\mathbf{x}$  is indexed with one upper index, but no lower index, we call  $\mathbf{x}$  a (1, 0)-tensor. Further, let  $\mathfrak{B} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$  be a basis of  $X$ , i.e. a set of linearly independent vectors in  $X$  such that every vector in  $X$  can be written as a linear combination of the elements in  $\mathfrak{B}$ . We also assume that the following conditions hold for the basis vectors in  $\mathfrak{B}$ :

$$\begin{aligned} \|\mathbf{e}_\mu\| &= 1 & \forall \mu \in \{1, 2, \dots, n\} \\ \mathbf{e}_\mu \cdot \mathbf{e}_\nu &\in [0, 1] & \forall \mu, \nu \in \{1, 2, \dots, n\} \\ \mathbf{e}_\mu \cdot \mathbf{e}_\nu &= 1 \quad \text{iff} \quad \mu = \nu \end{aligned}$$

Since the basis vectors are assumed to have unit length it follows that  $\mathbf{e}_\mu \cdot \mathbf{e}_\nu = \cos \theta$ , where  $\theta$  denotes the angle between  $\mathbf{e}_\mu$  and  $\mathbf{e}_\nu$ . It should also be observed that the reciprocal dot product between two non-identical basis vectors is not necessarily 0, i.e.  $\mathfrak{B}$  is not necessarily *orthonormal*. Then  $\mathbf{x}$  can then be written as a linear combination of the basis vectors in  $\mathfrak{B}$  as follows:

$$\mathbf{x} = x^1 \mathbf{e}_1 + x^2 \mathbf{e}_2 + \dots + x^n \mathbf{e}_n = \sum_{\mu=1}^n x^\mu \mathbf{e}_\mu$$

Using the *Einstein summation convention* (De et al., 2008, p. 5) we may succinctly write this sum as  $\mathbf{x} = x^\mu \mathbf{e}_\mu$ . We recursively define an *inner product*  $\langle \cdot, \cdot \rangle$  in  $V$  by assuming the following initial condition:

$$\langle \mathbf{e}_\mu, \mathbf{e}_\nu \rangle = \mathbf{e}_\mu \cdot \mathbf{e}_\nu \quad \forall \mu, \nu \in \{1, 2, \dots, n\} \quad (7.1)$$

Since the inner product is by definition *linear* in its arguments, i.e for all vectors  $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$  and for every scalar  $\alpha \in \mathbb{R}$  it holds that

$$\begin{aligned} \langle \alpha \mathbf{x}, \mathbf{y} \rangle &= \alpha \langle \mathbf{x}, \mathbf{y} \rangle \\ \langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle &= \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle, \end{aligned}$$

it follows that the condition in (7.1) is sufficient for deducing the inner product between any pair of vectors in  $X$ .

In order to get a general picture of how the inner product is computed for any vector in  $X$ , let us consider two example vectors

$$\begin{aligned} \mathbf{x} &= (x^1, x^2) = x^1 \mathbf{e}_1 + x^2 \mathbf{e}_2 \\ \mathbf{y} &= (y^1, y^2) = y^1 \mathbf{e}_1 + y^2 \mathbf{e}_2 \end{aligned}$$

From the definition of the inner product  $\langle \cdot, \cdot \rangle$  we get

$$\begin{aligned} \langle \mathbf{x}, \mathbf{y} \rangle &= x^1 y^1 (\mathbf{e}_1 \cdot \mathbf{e}_1) + x^2 y^2 (\mathbf{e}_2 \cdot \mathbf{e}_2) + x^1 y^2 (\mathbf{e}_1 \cdot \mathbf{e}_2) + x^2 y^1 (\mathbf{e}_2 \cdot \mathbf{e}_1) \\ &= \sum_{\mu=1}^2 \sum_{\nu=1}^2 x^\mu x^\nu (\mathbf{e}_\mu \cdot \mathbf{e}_\nu) \end{aligned}$$

It is easily confirmed that the general computational formula for computing the inner product of two  $n$ -dimensional vectors is

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{\mu=1}^n \sum_{\nu=1}^n x^\mu x^\nu (\mathbf{e}_\mu \cdot \mathbf{e}_\nu) = x^\mu x^\nu (\mathbf{e}_\mu \cdot \mathbf{e}_\nu)$$

where the last expression is using the Einstein summation convention.

We may express this operation in terms of matrix products as follows, letting  $g_{\mu\nu} = \mathbf{e}_\mu \cdot \mathbf{e}_\nu$  for notational convenience:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \begin{bmatrix} x^1 & x^2 & \dots & x^n \end{bmatrix} \begin{bmatrix} g_{1,1} & g_{1,2} & \dots & g_{1,n} \\ g_{2,1} & g_{2,2} & \dots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n,1} & g_{n,2} & \dots & g_{n,n} \end{bmatrix} \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix}$$

Letting  $\mathbf{E} = [ \mathbf{e}_1 \ \mathbf{e}_2 \ \dots \ \mathbf{e}_n ]$  be a matrix consisting of all basis vectors as column vectors, and letting  $\mathbf{G} = \mathbf{E}^T \mathbf{E}$ , we may abbreviate the expression above to

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{G} \mathbf{y}$$

The matrix  $\mathbf{G}$  is called the *Gram matrix* (see e.g. Shawe-Taylor & Cristianini, 2004, p. 32) over the basis vectors. We observe two important details concerning  $\mathbf{G}$ :

1. Its diagonal entries are all 1, since we have stipulated that  $\mathbf{e}_\mu \cdot \mathbf{e}_\mu = 1$  for all the basis vectors.
2. By letting the off-diagonal entries be 0, i.e.

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

the corresponding inner product equals the ordinary Euclidean dot product, since the following condition will hold:

$$\langle \mathbf{x}, \mathbf{y} \rangle = x^\mu y^\mu = \mathbf{x} \cdot \mathbf{y}$$

The classical definition of the vector space model is based on the assumption that documents can be modelled as vectors in a Euclidean vector space, having an orthonormal basis of which each basis vector corresponds to a term in the indexing vocabulary. Since the term vectors are reciprocally orthogonal it follows that they are linearly independent. The underlying assumption of such a vector basis is therefore that the relatedness, or the similarity, between any two terms  $k_i, k_j$  in a vocabulary  $V$  can be expressed as a function  $\rho : V \times V \rightarrow \{0, 1\}$  with the following definition:

$$\rho(k_i, k_j) = \begin{cases} 1 & \text{if } k_i = k_j \\ 0 & \text{otherwise} \end{cases}$$

This function yields a simple topological structure on  $V$  called a *discrete topology*, and  $\rho$  is correspondingly called a *discrete metric* (Willard, 2004, p. 17, 24) in this context.

However, in the *generalized vector space model* (Wong et al., 1985) the cosine similarity (and thereby the inner product, assuming that document vectors have been normalized to unit length) between any two document vectors  $\mathbf{x}, \mathbf{y}$  is defined

$$\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{x}^\top \mathbf{G} \mathbf{y}$$

of which  $\mathbf{G}$  has the following structure

$$\mathbf{G} = \begin{bmatrix} 1 & \rho_{1,2} & \dots & \rho_{1,n} \\ \rho_{2,1} & 1 & \dots & \rho_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n,1} & \rho_{n,2} & \dots & 1 \end{bmatrix}$$

where  $\rho_{i,j}$  denotes the magnitude of the semantic similarity between

$k_i$  and  $k_j$ , a value that can be assumed to be normalized to the range  $[0, 1]$ .

### 7.1.1 Formal definition of a semantic kernel

Formally, a semantic similarity measure  $\text{sim}$  over a vocabulary  $V$  is in this work defined as a function

$$\text{sim} : V \times V \rightarrow [0, 1]$$

such that

$$\begin{aligned} \text{sim}(x, y) &= \text{sim}(y, x) && \text{(commutativity)} \\ \text{sim}(x, x) &= 1 \end{aligned}$$

The general interpretation of this family of similarity measures is that given two terms  $k_i$  and  $k_j$ , a return value of 1 denotes complete semantic similarity 0 complete dissimilarity between  $k_i$  and  $k_j$ . Expressed differently: every semantic similarity measure induces a symmetric and square matrix  $\mathbf{S}$  of similarity values with 1's along the diagonal and  $0 \leq \text{sim}(x, y) \leq 1$  elsewhere. Each matrix  $\mathbf{S}$ , as generated by a specific method for statistical semantics, is in this work used to induce a semantic linear kernel in the following fashion:

$$\begin{aligned} \phi(\mathbf{x}) &:= \mathbf{S}\mathbf{x} \\ \kappa_{\mathbf{S}}(\mathbf{x}, \mathbf{y}) &= \phi(\mathbf{x}) \cdot \phi(\mathbf{y}) = (\mathbf{S}\mathbf{x})^{\top}(\mathbf{S}\mathbf{y}) = \mathbf{x}^{\top}\mathbf{G}\mathbf{y} \quad \text{where } \mathbf{G} := \mathbf{S}^2 \end{aligned}$$

Since  $\kappa$  is defined as the dot product over vectors transformed by the map  $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  it follows that  $\kappa$  is a positive-definite kernel (Hofmann et al., 2008). In this work we only study *linear* kernels induced by a semantic matrix, but analogous extensions could be made of, for instance, the *radial basis function* (RBF) kernel. Since the dot product

is distributive it follows that

$$\begin{aligned}
 \|\phi(\mathbf{x}) - \phi(\mathbf{y})\|^2 &= (\phi(\mathbf{x}) - \phi(\mathbf{y}))^\top (\phi(\mathbf{x}) - \phi(\mathbf{y})) \\
 &= (\mathbf{S}(\mathbf{x} - \mathbf{y}))^\top \mathbf{S}(\mathbf{x} - \mathbf{y}) \\
 &= (\mathbf{x} - \mathbf{y})^\top \mathbf{S}^2(\mathbf{x} - \mathbf{y}) \\
 &= (\mathbf{x} - \mathbf{y})^\top \mathbf{G}(\mathbf{x} - \mathbf{y})
 \end{aligned}$$

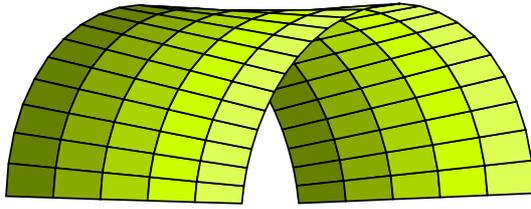
The corresponding semantic RBF kernel would therefore have the general form

$$\kappa_{\mathbf{S}}(\mathbf{x}, \mathbf{y}) := \exp\{-\gamma(\mathbf{x} - \mathbf{y})^\top \mathbf{G}(\mathbf{x} - \mathbf{y})\}$$

### 7.1.2 The metric tensor of Mercer kernels

In this section we will briefly present the geometry induced by Mercer kernels, including the semantic kernels described in the previous section. A *manifold* is a topological space that locally (close to each point) “looks like”, and is *homeomorphic* to, a Euclidean space (Boothby, 2003, p. 6). For instance, a circle segment is homeomorphic to a line segment (which in turn is a subset of  $\mathbb{R}^1$ ), and is therefore a manifold. A topological space  $M$ , such that its tangent space  $T_x M$  defined at a point  $\mathbf{x}$  has an inner product (or generally speaking, a positive-definite bilinear form) is called *Riemannian manifold*. The defining property of Riemannian manifolds makes it possible to construct measures like *arc length* on a surface (Boothby, 2003, p. 179-181). To this end a tensor called a *Riemannian metric* is used. For an illustration of a Riemannian manifold, see figure 7.1.

The derivation of the connection between Mercer kernels and the Riemannian metric that follows below is based on the presentation given by Amari & Wu (1999). Let  $X$  be a Euclidean vector space and  $M$  a Riemannian manifold. Consider a smooth map  $\phi : X \rightarrow M$ .



**Figure 7.1.** A Riemannian manifold is a topological space that locally (from a “micro-perspective”) resembles a Euclidean space, such as a two-dimensional plane, but which globally (from a “macro-perspective”) has other properties (cf. the surface of a sphere which locally may be perceived as “flat”).

From the definition of the partial derivative of  $\phi$  in any of its arguments  $x^\mu$  it follows that

$$dz^\mu = \sum_{\mu} \frac{\partial}{\partial x^\mu} \phi(\mathbf{x}) dx^\mu$$

and hence

$$d\mathbf{z} = \nabla \phi(\mathbf{x}) \cdot d\mathbf{x}$$

The squared  $\ell^2$ -norm of  $d\mathbf{z}$  is given by

$$\begin{aligned} \|d\mathbf{z}\|^2 &= \sum_{\mu} (dz^\mu)^2 = \left( \sum_{\mu} \frac{\partial}{\partial x^\mu} \phi(\mathbf{x}) dx^\mu \right)^2 \\ &= \sum_{\mu, \nu} \left( \frac{\partial}{\partial x^\mu} \phi(\mathbf{x}) \right) \cdot \left( \frac{\partial}{\partial x^\nu} \phi(\mathbf{x}) \right) dx^\mu dx^\nu \end{aligned}$$

By letting

$$g_{\mu\nu}(\mathbf{x}) := \left( \frac{\partial}{\partial x^\mu} \phi(\mathbf{x}) \right) \cdot \left( \frac{\partial}{\partial x^\nu} \phi(\mathbf{x}) \right)$$

we obtain the expression for a *Riemannian metric* (Boothby, 2003, p. 182) defined at a point  $\phi(\mathbf{x})$ , using Einstein summation:

$$\|d\mathbf{z}\|^2 = g_{\mu\nu}(\mathbf{x})dx^\mu dx^\nu$$

The function  $g_{\mu\nu}(\mathbf{x})$  is called the *metric tensor* at  $\phi(\mathbf{x})$  (De et al., 2008, p. 64). Now, since we have assumed that

$$\kappa(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$$

and

$$\frac{\partial^2}{\partial x^\mu \partial y^\nu} \kappa(\mathbf{x}, \mathbf{y}) = \left( \frac{\partial}{\partial x^\mu} \phi(\mathbf{x}) \right) \cdot \left( \frac{\partial}{\partial y^\nu} \phi(\mathbf{x}) \right)$$

it immediately follows that the metric tensor induced by the kernel  $\kappa$  for  $\mathbf{x}$  is given by

$$g_{\mu\nu}(\mathbf{x}) = \frac{\partial^2}{\partial x^\mu \partial y^\nu} \kappa(\mathbf{x}, \mathbf{y}) \Big|_{\mathbf{y}=\mathbf{x}}$$

Given a semantic kernel based on a term-term similarity matrix  $\mathbf{G}$ , and for which the kernel has the definition

$$\kappa(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{G} \mathbf{y}$$

the second-order partial derivative with respect to two coordinates  $x^\mu$  and  $y^\nu$  is

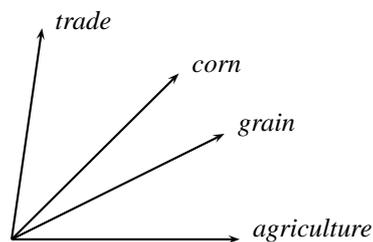
$$\frac{\partial^2}{\partial x^\mu \partial y^\nu} \kappa(\mathbf{x}, \mathbf{y}) = \frac{\partial^2}{\partial x^\mu \partial y^\nu} (\mathbf{x}^\top \mathbf{G} \mathbf{y}) = g_{\mu\nu}$$

We conclude that the term-term similarity matrix  $\mathbf{G}$  is a metric tensor on the Riemannian manifold induced by the semantic kernel.

## 7.2 Distributional semantics

The *distributional hypothesis* of semantics states that similar words occur in similar contexts (Sahlgren, 2008). A natural consequence of this hypothesis is that it should be possible to capture and disambiguate between different senses of words according to their distributions in different contexts and their co-occurrence patterns with other words (Schütze, 1998). By *context* is here loosely meant a set of words within a certain distance from a target word in a text. Sahlgren (2008) differentiates between syntagmatic and paradigmatic types of co-occurrence relations in texts. *Syntagmatic* relations are such word relations that appear by direct co-occurrence in sentence constructions, such as the relation between “eat” and “breakfast” in the sentence *I eat breakfast*. *Paradigmatic* relations appear when two words co-occur indirectly by appearing in the same contexts without being present in the same sentences, such as the relation between “breakfast” and “lunch” in the context of verbs like “eat”. A common representational framework for distributional properties of word semantics is the use of sense vectors in a Euclidean space (Sahlgren, 2008; Widdows, 2008; Clarke, 2012). The space of sense vectors, exemplified in figure 7.2, is called a *word space* by Schütze (1998). The similarity between words in this framework can be computed as the *cosine* or the normalized *correlation* between the word vectors.

The collection and representation of word co-occurrence statistics can be performed by the use of *context windows* of a stipulated size, followed by the computation of *context vectors* (Schütze, 1998; Sahlgren, 2008), or by the dimension reduction of a (typically sparse) term-by-document matrix using techniques like *singular value decomposition* (Landauer & Dumais, 1997). The use of singular value decomposition achieves a smoothing and denoising of term frequencies



**Figure 7.2.** A set of pairwise non-orthogonal sense vectors in a word space.

appearing in sparse document vectors (Schütze, 1992).

A large variety of methods for the acquisition and representation of word senses using co-occurrence statistics have been developed, tested, and used for practical problems, such as pointwise mutual information (Turney, 2001), latent semantic analysis (Landauer & Dumais, 1997), probabilistic latent semantic analysis (Hofmann, 1999), latent Dirichlet allocation (Blei et al., 2003), hyperspace analogue to language (Lund & Burgess, 1996), random indexing (Kanerva et al., 2000), reflective random indexing (Cohen et al., 2010), the Google similarity distance (Cilibrasi & Vitanyi, 2007), as well as measures based on semantic relations found in WordNet (Budanitsky & Hirst, 2006).

### 7.3 Methods for measuring semantic similarity

As mentioned in section 1.1, one of the major research objectives of this work is to study the classification performance of semantic kernels induced by term relations in the training corpus. The methods selected for the empirical study contained in this work are *pointwise mutual information*, *latent semantic analysis*, and *random indexing*.

### 7.3.1 Latent semantic analysis

The *latent semantic analysis* (LSA) method (Deerwester et al., 1990) is based on the assumption that the presence of terms in particular documents can be explained in terms of a smaller number of latent factors, termed *concepts*, in the documents. The aim of LSA is to reduce the dimensionality of the document feature space, and thereby remove statistical noise and enhance associations between groups of terms. We now give a brief, formal description of the LSA method:

Let  $\mathbf{M}$  be a real-valued *term-by-document* matrix of dimensionality  $m \times n$ . The *singular value decomposition* (SVD) theorem states that  $\mathbf{M}$  can be decomposed (factorized) into three matrices  $\mathbf{U}$ ,  $\mathbf{S}$  and  $\mathbf{V}$  such that  $\mathbf{U}$  and  $\mathbf{V}$  are *unitary*,  $\mathbf{S}$  is *diagonal* and

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^{\top} \quad (7.2)$$

We define a sequence of diagonal matrices  $\tilde{\mathbf{S}}^{(1)}, \dots, \tilde{\mathbf{S}}^{(m)}$  such that

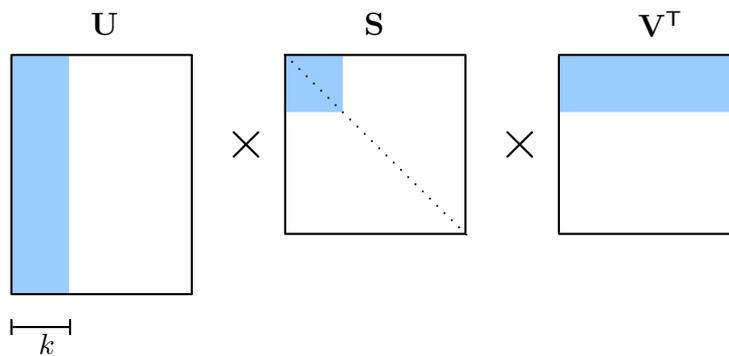
$$\tilde{\mathbf{S}}_{i,j}^{(k)} = \begin{cases} \mathbf{S}_{i,j} & \text{if } i \leq k \\ 0 & \text{otherwise} \end{cases} \quad (7.3)$$

In other words, each matrix  $\tilde{\mathbf{S}}^{(k)}$  retains the  $k$  largest singular values from  $\mathbf{S}$  and consequently has rank  $k$ . Correspondingly, the sequence  $\tilde{\mathbf{M}}^{(1)}, \dots, \tilde{\mathbf{M}}^{(m)}$  generated by the definition

$$\tilde{\mathbf{M}}^{(k)} = \mathbf{U}\tilde{\mathbf{S}}^{(k)}\mathbf{V}^{\top} \quad (7.4)$$

constitute rank- $k$  approximations of  $\mathbf{M}$  (see figure 7.3).

We measure the approximation error of the rank- $k$  matrix  $\tilde{\mathbf{M}}^{(k)}$



**Figure 7.3.** A diagrammatic view of the dimension reduction of a matrix decomposed by SVD. The blue area represents the reduction to  $k$  dimensions.

according to

$$\text{err}(\tilde{\mathbf{M}}^{(k)}) := \|\mathbf{M} - \tilde{\mathbf{M}}^{(k)}\|_F, \quad (7.5)$$

where  $\|\cdot\|_F$  is the *Frobenius norm*. This error is minimized if the column vectors of  $\mathbf{U}$  are the eigenvectors of  $\mathbf{M}\mathbf{M}^\top$  and the column vectors of  $\mathbf{V}$  are the eigenvectors of  $\mathbf{M}^\top\mathbf{M}$ . Hence, the optimal rank- $k$  approximation of  $\mathbf{M}$  can be obtained by SVD based on this eigen-decomposition.

The matrix  $\mathbf{P}$  of dot products of the row vectors in  $\mathbf{M}$  is obtained

by  $\mathbf{P} = \mathbf{M}\mathbf{M}^\top$ , which can be written, using the SVD of  $\mathbf{M}$ ,

$$\begin{aligned} \mathbf{P} &= (\mathbf{U}\mathbf{S}\mathbf{V}^\top) (\mathbf{U}\mathbf{S}\mathbf{V}^\top)^\top \\ &= (\mathbf{U}\mathbf{S}\mathbf{V}^\top) (\mathbf{V}\mathbf{S}\mathbf{U}^\top) \\ &= (\mathbf{U}\mathbf{S}) (\mathbf{S}\mathbf{U}^\top) \\ &= (\mathbf{U}\mathbf{S}) (\mathbf{U}\mathbf{S})^\top \end{aligned}$$

In conclusion we see that the  $\mathbf{M}$  can be replaced by  $\mathbf{U}\mathbf{S}$  for computations based on the dot product of the row vectors of  $\mathbf{M}$ . The optimal rank- $k$  approximation of  $\mathbf{M}$  for dot product computations is consequently  $\mathbf{U}\tilde{\mathbf{S}}^{(k)}$ , which is the matrix used in LSA to obtain a dimension-reduced and denoised term representation.

The LSA method has been studied and used in the context of learning word senses for a vocabulary (Landauer & Dumais, 1997), information retrieval (Berry et al., 1995; Grönqvist, 2006), text categorization (Zelikovitz & Hirsh, 2001; Wu & Gunopulos, 2002), and text summarization (Gong & Liu, 2001). A well-known disadvantage of this method is, however, that it is computationally expensive (Chen & Saad, 2009), which necessitates the use of efficient methods for computing the SVD factorization, such as Lanczos solvers, distributed algorithms, and stochastic algorithms (Řehůřek, 2011).

### 7.3.2 Random indexing

In contrast to the LSA method, *random indexing* (RI) utilizes term *context windows* to create representation vectors (called *context vectors*) for a given vocabulary (Karlgrén & Sahlgrén, 2001). The notion of term context is illustrated in figure 7.4. Each term  $t_i$  in a vocabulary  $V$  is initially assigned an *index vector*  $\mathbf{n}_i$ , which is a sparse vector having a small proportion of randomly positioned non-zero values (an

equal number of the values  $-1$  and  $+1$ ). The dimensionality of the index vectors is typically much smaller than the number of contexts used for the random indexing procedure. This is theoretically justified by the Johnson-Lindenstrauss lemma, which states that the projection of vectors in a vector space  $X$  into a randomly selected subspace of  $Y \subset X$  will approximately preserve the distances between the projected vectors if  $Y$  has a sufficiently high dimensionality. Having only a small number of randomly positioned non-zero values, the index vectors will also be nearly orthogonal (Sahlgren & Karlgren, 2005).

“To sleep, perchance to dream – ay, there’s the rub.”  


**Figure 7.4.** Words occurring in each other’s context will obtain similar vectorial representations.

Each term  $t_i$  is also assigned a *context vector*  $\mathbf{c}_i$  in  $\mathbb{R}^n$ , which is initially set to  $\mathbf{0}$  (the zero vector in  $\mathbb{R}^n$ ). To compute context vectors a set of texts is scanned. For each occurrence of a term  $t_i$  the *context* of  $t_i$ , consisting of a fixed number of terms preceding  $t_i$  and succeeding  $t_i$ , is used to update the context vector of  $t_i$ . Let  $N(i, r)$  be the  $r$ th context of  $t_i$ . Then the context vector  $\mathbf{c}_i$  is updated according to

$$\mathbf{c}_i \leftarrow \mathbf{c}_i + \sum_{t_j \in N(i, r)} g(i, j) \cdot \mathbf{n}_j$$

where  $g(i, j)$  is a weight function dependent on the distance between the specific occurrences of  $t_i$  and  $t_j$ . The context vectors are used as sense vectors for the terms in  $V$ . The computational advantage of this method over LSA described in the previous section is that random indexing is easier to update with new information and that the computations basically consist of vector additions, which are considerably less computationally expensive than factorizations of large matrices.

In a study of random indexing involving 37,600 text samples and a vocabulary of 79,000 words, Kanerva et al. (2000) found that the un-normalized context vectors (1800 dimensions) yielded between 35% and 44% correct answers on a TOEFL (Test of English as a Foreign Language) test. Using thresholding on the words-by-context matrix by mapping it onto a matrix consisting of the values  $\{-1, 0, 1\}$  increased the result to between 48% and 51% correct answers on the same test.

### 7.3.3 Pointwise mutual information

The *mutual information* between two stochastic variables  $X$  and  $Y$  is, intuitively speaking, a measure that quantifies how much information the variables provide about each other. Viewed differently, the mutual information between two variables  $X$  and  $Y$  is the amount of *uncertainty* concerning  $X$  that is reduced by observing  $Y$  (and vice versa). Letting  $H(\cdot)$  denote *information entropy* the mutual information  $\text{MI}(X, Y)$  is defined (Bouma, 2009):

$$\begin{aligned} \text{MI}(X, Y) &:= H(X) + H(Y) - H(X, Y) = \\ &\sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left( \frac{p(x, y)}{p(x)p(y)} \right) \end{aligned}$$

The *pointwise* mutual information is correspondingly the amount of information available between two *outcomes*  $x \in X$  and  $y \in Y$ , and has the definition (Bouma, 2009)

$$\text{PMI}(x, y) = \log \left( \frac{p(x, y)}{p(x)p(y)} \right)$$

This measure can further be *normalized* into the range  $[-1, 1]$  as follows, yielding the *normalized* pointwise mutual information:

$$\text{NPMI}(x, y) = -\frac{\text{PMI}(x, y)}{\log(p(x, y))}$$

Since the joint probability  $p(x, y) = p(x)p(y)$  iff  $x$  and  $y$  are statistically independent it follows that  $\text{PMI}(x, y) = \log 1 = 0$  and hence also  $\text{NPMI}(x, y) = 0$ . Further, assume that  $x$  and  $y$  are always observed together (i.e. if we observe  $x$  we will also observe  $y$ , and vice versa). Then it holds that  $p(x, y) = p(x) = p(y)$ . Then  $\text{PMI}(x, y) = -\log(p(x, y))$  and hence  $\text{NPMI}(x, y) = 1$ .

Hence, the NPMI is useful as a normalized measure of positive correlation between terms, for which we only consider values in the range  $[0, 1]$  (i.e. we discard term correlations below zero). The PMI measure was proposed by K. W. Church & Hanks (1990) as a measure of *association ratio* between words in full-text corpora. It has also been successfully used to detect synonymy relations between terms in web pages and has shown a performance comparable to (and even surpassing) the latent semantic analysis method on a TOEFL test (Turney, 2001).

### 7.3.3.1 A comparison between the measures

Both the LSA and the RI methods produce term vectors containing information about term co-occurrence in the analyzed corpora. However, the LSA method does not (unlike random indexing) take paradigmatic term co-occurrence (and to some extent textual distance between terms) into account. Since it is also computationally costlier it is naturally of interest to make a comparison between the two methods in terms of classification performance. Does the choice between LSA and RI involve a trade-off between computational complexity

and classification performance, or is the RI method comparable to LSA also in terms of classification performance? Further, the NPMI measure is computationally simple, but makes only pairwise comparisons between terms, focusing on the syntagmatic co-occurrence of words. Unlike the other two methods it is based on a probabilistic, information-theoretic framework. In conclusion, these methods have quite specific, and essentially different, properties that make them interesting for comparison in the context of text categorization with semantic kernels.

## Chapter 8

# Experimental setup

In the previous chapters we have formally analyzed the theoretical properties of document classification, as well as given a presentation of automatic classification in general and the support vector machine (SVM) algorithm in particular. We have also discussed the notion of SVM kernels enriched with semantic information, called *semantic* kernels. One major research objective of the empirical part of this thesis is to explore and comparatively study the classification performance of different semantic kernels for SVM, and also to compare the performance of semantic kernels to the performance of an “ordinary” linear kernel.

The semantic information used in these kernels has been obtained from three different methods for statistical semantics: pointwise mutual information (PMI), latent semantic analysis (LSA), and random indexing (RI). The underlying theory and algorithmic structure of these methods has been presented in chapter 7. The data underlying the induction of the semantic kernels has been the full set of training documents available for the selected classes in the reference collection. In other words, the training documents available in the reference collec-

tion have been used both to generate a semantic model for the class-specific vocabulary, as well as to induce an SVM classifier.

Another research objective is to compare the classification performance of two different term weighting schemes: the commonly used *tf-idf* weighting scheme, and the *divergence from randomness* (henceforth abbreviated *dfr*) weighting scheme respectively. The latter weighting scheme was initially developed as language model for information retrieval (see Amati & Van Rijsbergen, 2002), but has been adapted in this work for use as a content representation model for automatic classification.

## 8.1 General procedure

The overall structure of the experimental setup of this study can be summarized in the following steps:

1. Selection of reference collections.
2. Generation of document feature vectors, using the weighting schemes *tf-idf* and *dfr*.
3. Generation of semantic kernels, using the methods pointwise mutual information, latent semantic analysis, and random indexing.
4. Training of classifiers over differently sized proportions of the training sets and over different numbers of features.
5. Testing and evaluation on samples from the test sets.
6. Statistical analysis of the empirical results.

We will now present and discuss each step of the methodological approach used in this study.

## 8.2 Selection of reference collections

The reference collections used in this study consist of short articles, abstracts and discussions in three different standard reference collections: the *Reuters-21578* data set, the first 20,000 abstracts of the *OHSUMED* reference collection, as well as *20 Newsgroups* data set. These collections are regarded as standard benchmark collections for performing automatic classification experiments and to make the results comparable to other studies (Manning et al., 2008, p. 142; Baeza-Yates & Ribeiro-Neto, 2011, p. 329). In the following sections we will briefly present these collections together with a description of how they have been used in this study. The *Reuters-21578* and the *OHSUMED* collections have been obtained from Moschitti (n.d.), whereas the *20 Newsgroups* collection has been obtained from Rennie (n.d.).

### 8.2.1 Reuters-21578

The *Reuters-21578* data set is a text collection consisting of 21,578 records generated from documents published by the Reuters newswire service in 1987. The current version of the collection became stable, after a period of editing and tagging, in 1996. It is claimed to be the most widely used test collection for text categorization studies (Baeza-Yates & Ribeiro-Neto, 2011, p. 329).

A subset of the *Reuters-21578* collection that is particularly used for text categorization experiments is the *ModApte split* (Joachims, 2002, p. 32) consisting of 9,603 documents for training and 3,299 test documents. The 10 most frequent categories (out of 90 categories) in the *ModApte split* were used in this study, which is a commonly used subcollection of the *ModApte split* (see e.g. Baeza-Yates & Ribeiro-Neto, 2011, p. 329). These categories together with their number of

training and test documents are listed in table 8.1.

<b>Class</b>	<b># of training docs</b>	<b># of test docs</b>
acq	1650	719
corn	181	56
crude	389	189
earn	2877	1087
grain	433	149
interest	347	131
money-fx	538	179
ship	197	89
trade	369	117
wheat	212	71
<b>Sum:</b>	<b>7193</b>	<b>2787</b>

**Table 8.1.** The 10 categories in the Reuters-21578 collection used in this study.

### 8.2.2 OHSUMED

The *OHSUMED* data set consists of 348,566 medical references published between the years 1987 and 1991 in the MEDLINE database and was prepared at the Oregon Health Sciences University (OHSU) to facilitate information research (Hersh et al., 1994). Each record is tagged with Medical Subject Headings (MeSH) terms, which in this study are used to induce categories on the collection. Apart from being used for classification studies it also serves as a reference collection for information retrieval experiments (Baeza-Yates & Ribeiro-Neto, 2011, p. 329-330). Joachims (1998) used a subselection of the *OHSUMED* records published in 1991, consisting of the 20,000 first records having abstracts, of which the first 10,000 records were

used for training and the subsequent 10,000 records were used for testing. This subset of the OHSUMED data set has also been used in this study to facilitate comparison. In the classification experiments reported in Joachims (1998) the performance of the SVM algorithm using polynomial and RBF kernels was compared against other algorithmic classifiers, more specifically the Naive Bayesian classifier, the k-NN method, the Rocchio method (see e.g. Baeza-Yates & Ribeiro-Neto, 2011, p. 300-303), and the C4.5 decision tree algorithm (see e.g. Kotsiantis, 2007). The obtained *precision/recall breakeven* values (i.e. the point where precision and recall are equal) for the polynomial kernel were between 58.2% for the *Pathology* category and 74.5% for the *Digestive System* category. The combined microaverage precision/recall for all the polynomial kernels used was 65.9%.

For this study 10 classes were randomly selected, which are listed in table 8.2.

<b>Class</b>	<b># of training docs</b>	<b># of test docs</b>
C01	423	506
C04	1163	1467
C06	588	632
C08	473	600
C10	621	941
C12	491	548
C14	1249	1301
C18	388	400
C20	525	695
C21	546	717
<b>Sum:</b>	<b>6467</b>	<b>7807</b>

**Table 8.2.** The 10 classes from the OHSUMED reference collection used in this study.

### 8.2.3 20 Newsgroups

The *20 Newsgroups* data set consists of messages posted in 20 different Usenet newsgroups (see e.g. Baeza-Yates & Ribeiro-Neto, 2011, p. 330). The categorization task involved in this data set is to assign each message to the newsgroup in which it was posted. For this study we have used the *20news-bydate* split of this dataset, prepared and published at (Rennie, n.d.). The *20news-bydate* dataset consists of 18,846 documents and is partitioned into 60% training documents and 40% test documents.

For this study 10 classes were randomly selected, which are listed in table 8.3.

Class	# of training docs	# of test docs
comp.graphics	584	389
comp.sys.mac.hardware	578	385
misc.forsale	585	390
rec.autos	594	396
rec.sport.hockey	600	399
sci.crypt	595	396
sci.space	593	394
talk.politics.mideast	564	376
talk.politics.misc	465	310
talk.religion.misc	377	251
<b>Sum:</b>	5535	3686

**Table 8.3.** The 10 classes from the 20 Newsgroups reference collection used in this study.

In order to generate data for the training sets and the test sets, the data (document texts and class information) in the reference collections has been processed through a number of transformation steps. The procedures used in this workflow are presented in the next section.

### 8.3 Generation of document representations

The following steps have been performed in order to generate document representations (document vectors). This procedure has been deliberately designed to follow the outline presented in (Joachims, 2002).

1. The basic unit of lexical analysis in this study are the *words* of the document texts. Here we define a *word* as a sequence of non-whitespace characters (Joachims, 2002, p. 33).
2. The texts have been converted into streams of words by means of *tokenization* (see e.g. Manning et al., 2008, p. 22). In this process numeric values (i.e. words consisting entirely of numeric characters) have been removed since numbers are generally not useful as index terms without a context (Baeza-Yates & Ribeiro-Neto, 2011, p. 224f). All words in the tokenized stream have been standardized to lower case (see e.g. Joachims, 2002, p. 33).
3. In order to reduce the index further a *stop list* has been used to filter out common function words such as prepositions and conjunctions. Following the example of Joachims (2002, p. 33) the stop list of the FreeWAIS information retrieval system has been employed in this work.
4. No morphological normalization (stemming or lemmatization) has been used.
5. For each target class, the terms in the collection vocabulary were weighted according of the chi-square measure (see section 5.4) and ranked in descending order according to their chi-square values, whereby the top  $k$  terms were retained as de-

scriptors for the target class. In this study we have performed the classification experiments for  $k \in \{ 100, 300 \}$ . We call the selected subset of terms the class-specific *vocabulary* of the documents.

6. When the class-specific vocabulary has been established the texts in the reference collections have been mapped onto document representations consisting of feature vectors (see section 5.4). To this end we have used two different term weighting schemes, *tf-idf* and *divergence from randomness*.

## 8.4 Term weighting

The following term weighting schemes, and configurations of these schemes, have been used in this study.

### 8.4.1 Tf-idf

In this study we have used a version of the *tf-idf* weighting scheme consisting of unnormalized *tf* weights, together with an inverse document frequency (*idf*) component defined as

$$\text{idf}(k_i) := \log_2 \left( \frac{N}{df_i} \right)$$

where  $N$  denotes the number of training documents in the reference collection, and  $df_i$  denotes the number of training documents in which the term  $k_i$  occurs. The document vectors have been normalized to unit length after computing the *tf-idf* weights. This configuration of the *tf-idf* scheme has been assigned the classification code *tf* by Salton & Buckley (1988) and is also one of the configurations used in the classification experiments reported by Joachims (2002).

### 8.4.2 Divergence from randomness

This is, as mentioned in section 5.4.2, a weighting scheme that hitherto appears to have been applied and studied exclusively as an information retrieval model. We have adapted this model for text categorization with SVM by letting the collection parameters  $F$  (the accumulated term frequency over all documents in the collection) and  $sl$  (the average length of a document in the collection) be obtained from the training set, while applying these parameters in the term weighting of *both* the training documents and the test documents. The underlying assumption of this procedure is that the training documents and the test documents have been generated by the same term distribution model. For the calculation of the term weights we have used the *Bose-Einstein* randomness model together with the *Bernoulli* model for computing the risk associated with selecting a specific term (see section 5.4.2). The resulting document vectors have finally been length-normalized to unit length.

## 8.5 Generation of semantic kernels

In section 7.1.1 we defined a linear semantic kernel  $\kappa$  as a symmetric function with the general definition

$$\kappa(\mathbf{x}, \mathbf{y}) := \mathbf{x}^\top \mathbf{G} \mathbf{y}$$

where  $\mathbf{G}$  is a positive-semidefinite, symmetric matrix. Further, we have defined  $\mathbf{G}$  by means of a term-term similarity matrix  $\mathbf{S}$  according to

$$\mathbf{G} := \mathbf{S}^2$$

Each entry  $s_{i,j}$  in  $\mathbf{S}$  corresponds to a similarity value between two terms  $k_i$  and  $k_j$  in an indexing vocabulary. If we let  $\mathbf{G} = \mathbf{I}$ , where  $\mathbf{I}$  denotes the *identity matrix*, we get the ordinary linear kernel. In this study we have obtained these similarity values from different methods for computing semantic similarity, more specifically *pointwise mutual information* (PMI), *latent semantic analysis* (LSA), and *random indexing* (RI). The *data* used as input to these methods has been the set of documents (both positive and negative examples) used for sampling the training sets. In other words, we have used the same set of documents for the training of the classifiers as for inducing semantic kernels. However, we have used the *full* set of training documents for generating the semantic kernels, whereas we have applied samples of different proportion sizes to induce the classifier. For example, in the Reuters-21578 collection we have used all the 7193 training documents available in the dataset to generate semantic kernels for the  $k$  terms selected for each target class.

Although all these methods theoretically output values in the range  $[-1, 1]$  we have retained the non-negative values for the construction of semantic matrices, letting negative output values be mapped to 0. The justification for this procedure is that we only consider the *positive* correlation between terms for augmenting the SVM kernels. It is also theoretically difficult to interpret the negative term weights that may ensue by using “non-truncated” similarity measures. These methods have been used as follows.

### 8.5.1 Pointwise mutual information (PMI)

For a description of this method see section 7.3.3. Unlike the other methods for statistical semantics used in this work, PMI is a measure based on (probabilistic) entropy rather than vectorial semantics. The

variant of PMI used in this study is the *normalized* measure (NPMI), yielding an output in the range  $[-1, 1]$ . The output value 0 denotes pairwise statistical independence and 1 denotes perfect correlation between terms (Bouma, 2009).

### 8.5.2 Latent semantic analysis (LSA)

For a description of this method see section 7.3.1. Following Turney (2001), we start by constructing a term-by-document matrix  $\mathbf{M}$  consisting of tf-idf weights.  $\mathbf{M}$  is then factorized using singular value decomposition into the matrices  $\mathbf{U}$ ,  $\Sigma$ , and  $\mathbf{V}$ . We then proceed to *truncate* the matrices, retaining only the  $m$  dimensions corresponding to the  $m$  highest singular values in  $\Sigma$ . In this study we have used  $m \in \{100, 300\}$ . The row vectors of the dimension-reduced matrix product  $\mathbf{U}_m \Sigma_m$  will contain  $m$ -dimensional approximations of the original term vectors in  $\mathbf{M}$ . The similarity between two terms  $k_i$  and  $k_j$  is then computed as the *cosine* between the corresponding term vectors in  $\mathbf{U}_m \Sigma_m$ .

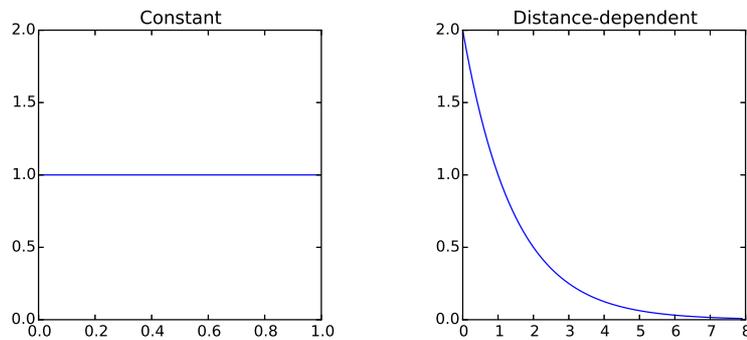
### 8.5.3 Random indexing (RI)

For a description of this method see section 7.3.2. A context window consisting of the  $m$  terms to the left as well as  $m$  terms to the right of the focus word  $k_i$  has been used, where  $m \in \{2, 4, 8\}$ . This means that we have used fairly small context windows in this study. The analysis has been performed with index vectors of dimensionality 1000, each index vector having 8 randomly selected coordinates set to  $-1$  or  $+1$  and the remaining coordinates set to 0. In other words, 0.8% of the coordinates in the index vectors were non-zero. The contribution to the context vector for a certain term  $k$  in the analysis window has been weighted by a coefficient  $w$  defined in terms of the distance from the

focus term, according to two different weighting strategies. We have used one *constant* weighting strategy assigning equal weight to all terms in the analysis windows, and one *distance-dependent* weighting strategy, where the influence of the terms in the context is monotonically decreasing with their distance to the focus term. Let  $\delta(k_i, k_j)$  denote the textual distance between the terms  $k_i$  and  $k_j$ . Then the weighting coefficient  $g$  is defined:

$$g(k_i, k_j) := \begin{cases} 1.0 & \text{if constant weighting is used} \\ 2^{1-\delta(k_i, k_j)} & \text{if distance-dependent weighting is used} \end{cases}$$

Figure 8.1 shows the graphs of the weighting coefficient for the two weighting strategies described above.



**Figure 8.1.** The weighting coefficient  $g$  as a function of the textual distance between two terms, using constant and distance-dependent weighting respectively.

As with the LSA method, we use the *cosine* between the context vectors to compute pairwise term-term similarity values.

## 8.6 Training and testing of SVM classifiers

Our investigation of the performance of different SVM classifiers induced by various semantic kernels involve a number of variables presented below. In this section we also describe our strategies for tuning the hyperparameters of the SVM algorithm, the sampling strategy for obtaining training sets from the reference collections, and the measures used to evaluate the performance of the classifiers.

### 8.6.1 Variables

This study involves the comparative study of one *dependent* variable, classification performance, and a number of *independent* variables described below.

1. *The kernel.* The linear kernel has been used as a baseline approach. For the semantic kernels the underlying semantic similarity measures have been parametrized according to the data given in table 8.4. PMI denotes *pointwise mutual information*, LSA- $m$  denotes latent semantic analysis with dimensionality  $m$ , RI-C- $m$  denotes *random indexing* with constant weighting and window size  $m$ , and RI-M- $m$  denotes *random indexing* with distance-dependent weighting and window size  $m$ .
2. *The number of features.* The chi-square measure was used to identify and select the  $k$  terms, where  $k \in \{ 100, 300 \}$ , that are most closely associated with the target class.
3. *The proportion of positive training examples.* The training sets were generated by randomly sampling  $k\%$  of the positive examples, where  $k \in \{ 10, 30, 50, 70, 90 \}$ , together with an equal amount of the negative examples from the collection of train-

Method	Parametrization
PMI	No parametrization is available, but the measure has been normalized to the range $[-1, 1]$ .
LSA- $m$	The $U$ , $\Sigma$ , and $V$ matrices have been truncated to $m \in \{100, 300\}$ dimensions.
RI-C- $m$	The terms in the context have been weighted equally. $m \in \{2, 4, 8\}$ denotes the size of the left and the right window.
RI-M- $m$	The terms in the context have been weighted according to $2^{1-\delta}$ , where $\delta$ denotes the textual distance to the index term. $m \in \{2, 4, 8\}$ denotes the size of the left and the right window.

**Table 8.4.** The parametrization of the semantic similarity measures used in this study.

ing documents, yielding training sets with an equal amount of positive and negative examples (see section 8.6.3).

## 8.6.2 Configuration of SVM hyperparameters

As indicated in chapter 6 the SVM algorithm involves a number of configurable hyperparameters for fine-tuning the training of the discrimination function. These include parameters that are specific for the selected kernel (such as the  $\gamma$  parameter for the radial basis function kernel), as well as the parameter  $C$  determining the trade-off between margin width and training error of the induced classifier. For certain classification problems the configuration of these parameters can have a significant impact on the eventual performance of the induced classifier (Duan et al., 2003). The higher value on the trade-off parameter, the less the classifier will be affected by outliers in the training data – but more support vectors will be produced, which increases the risk for generating an over-fitting model.

The values of the hyperparameters were selected using *grid search* (see e.g. Bergstra & Bengio, 2012). To this end we have followed the procedure recommended in (Hsu et al., 2010) by first using a coarse grid followed by a search in a finer grid around the optimal values found during the first iteration on the coarse grid. The search space for the coarse grid search proposed in (Hsu et al., 2010), and used in this work, are powers of 2 in the range  $2^{-5}, 2^{-3}, \dots, 2^{15}$  for the  $C$  trade-off parameter, and  $2^{-15}, 2^{-13}, \dots, 2^3$  for the  $\gamma$  parameter. The final selection of hyperparameters was based on the highest average F1 score on a 5-fold cross-validation test (for instance used in Duan et al., 2003).

### 8.6.3 Sampling procedure

A problem with many datasets, and among the datasets used in this study the Reuters-21578 collection in particular, is the imbalanced distribution of classes in the set. A naive approach to the sampling of such a dataset may yield classifiers that are prone to assign most or all of the documents to the majority class (Ling et al., 1998). For classes having few positive examples the induced classifiers would tend to output only negative class labels. To approach this problem Kubat & Matwin (1997) suggest a simple and straightforward approach, namely that the majority of the negative examples should be removed in order to make distribution of the positive and negative examples balanced. This is an approach called *undersampling* by Estabrooks et al. (2004). In this study we have used undersampling to generate completely balanced training sets, consisting of an equal amount of positive and negative examples. On the other hand, all the test documents have been retained for the evaluation of the classifiers. In other words: we have retained the class imbalance in the test sets.

The sampling of training documents followed by evaluation on the test set has been repeated  $n = 30$  times for each session, in order to be able to compute confidence intervals for the performance measures.

For the generation of samples for cross-fold validation, training, testing, as well as the generation of random index vectors for the random indexing method, a modern random number generation algorithm called the *Mersenne Twister* (published in Matsumoto & Nishimura, 1998) has been used, more specifically the MT19937 algorithm, having the extremely large period  $2^{19937} - 1$ .

#### 8.6.4 Evaluation

By the *performance* of a machine-based document classifier we denote different properties related to the extent to which a machine classifier accurately reproduces a manual classification of a set of documents in a set of test documents. To define various performance measures it is convenient to use a  $2 \times 2$  *contingency table* of frequencies related to the classification performance. For a given target class  $c_i \in C$  we let, as previously,  $\varphi_i$  denote a binary target classifier, and  $\psi_i$  a trained machine classifier. Then to each classified document  $d_j$  in the test set we assign one of the following observation categories:

- *true positive* (TP) if  $\varphi_i(d_j) = 1$  and  $\psi_i(d_j) = 1$
- *false positive* (FP) if  $\varphi_i(d_j) = 0$  and  $\psi_i(d_j) = 1$
- *true negative* (TN) if  $\varphi_i(d_j) = 0$  and  $\psi_i(d_j) = 0$
- *false negative* (FN) if  $\varphi_i(d_j) = 1$  and  $\psi_i(d_j) = 0$

We count the number of documents and generate the following contingency table:

	$\psi_i(d_j) = 1$	$\psi_i(d_j) = 0$
$\varphi_i(d_j) = 1$	$TP$	$FN$
$\varphi_i(d_j) = 0$	$FP$	$TN$

**Table 8.5.** A contingency table for classification evaluation.

It should be noted that the abbreviations (TP etc) in table 8.5 denote *frequencies*. We will now define a few common performance measures using the frequencies as defined in table 8.5. In addition we will use the symbol  $N$  to denote the total number of classified documents (i.e.  $N := TP + TN + FP + FN$ ).

#### 8.6.4.1 Accuracy

*Accuracy* is defined as the proportion of correctly classified documents in the test set and has the following definition (Baeza-Yates & Ribeiro-Neto, 2011, p. 326):

$$\text{Acc} := \frac{TP + TN}{N}$$

#### 8.6.4.2 Error

Another related measure is *error* (Baeza-Yates & Ribeiro-Neto, 2011, p. 326), which provides an estimate of the probability that an induced classifier will predict the wrong class label. This measure is defined:

$$\text{Err} := \frac{FP + FN}{N}$$

From the definition of  $N$  it follows that  $FP + FN = N - (TP + TN)$ . Hence,

$$\text{Err} = \frac{N - (TP + TN)}{N} = 1 - \frac{TP + TN}{N} = 1 - \text{Acc}$$

It is generally disadvised to use *accuracy* and *error* for binary classification tasks, since there are in general few *positive* examples of the target class available in the given test set (cf. the discussion in section 8.6.3 below), and a classifier that avoids assigning any documents to the target class may therefore obtain a high accuracy and a low error, which would be misleading performance indicators (Baeza-Yates & Ribeiro-Neto, 2011, p. 326).

#### 8.6.4.3 Precision, recall, and $F_1$

Two other measures, which are also often used in the evaluation of information retrieval studies, are *precision* and *recall* (Baeza-Yates & Ribeiro-Neto, 2011, p. 327). Given a  $2 \times 2$  contingency table for a single class  $c_i$  the *precision* measure has the following definition:

$$\text{Prec}(i) := \frac{TP}{TP + FP}$$

The *recall* measure is defined

$$\text{Rec}(i) := \frac{TP}{TP + FN}$$

To generate a single-value summary of precision and recall the family of *F-measures* is often applied, which are defined as the the weighted harmonic mean of precision and recall. The most commonly used variant of the F-measures, the so called  $F_1$  measure, has the definition (Joachims, 2002, p. 30; Baeza-Yates & Ribeiro-Neto, 2011, p. 328):

$$F_1 := \frac{2 \cdot \text{Prec}(i) \cdot \text{Rec}(i)}{\text{Prec}(i) + \text{Rec}(i)}$$

To compute an average of the performance measures recall and precision over a set  $C = \{c_1, \dots, c_k\}$  of classification codes two dif-

ferent approaches, called *micro-averaging* and *macro-averaging* are typically used (Baeza-Yates & Ribeiro-Neto, 2011, p. 328). When *micro-averaging* over  $C$  the performance measures is computed by aggregating all classification decisions:

$$\text{Rec}_{\text{micro}}(\mathcal{X}, C, \psi) := \frac{\sum_{i=1}^{|C|} |\{(\mathbf{x}, y) \in \mathcal{X} : y = c_i \text{ and } \psi(\mathbf{x}) = c_i\}|}{\sum_{i=1}^{|C|} |\{(\mathbf{x}, y) \in \mathcal{X} : y = c_i\}|}$$

$$\text{Prec}_{\text{micro}}(\mathcal{X}, C, \psi) := \frac{\sum_{i=1}^{|C|} |\{(\mathbf{x}, y) \in \mathcal{X} : y = c_i \text{ and } \psi(\mathbf{x}) = c_i\}|}{\sum_{i=1}^{|C|} |\{(\mathbf{x}, y) \in \mathcal{X} : \psi(\mathbf{x}) = c_i\}|}$$

Conversely, when *macro-averaging* over  $C$  the performance measures are calculated separately for each class in  $C$  and an average of the performance values is computed.

$$\text{Rec}_{\text{macro}}(\mathcal{X}, C, \psi) := \frac{1}{|C|} \sum_{i=1}^{|C|} \text{Rec}(\mathcal{X}, c_i, \psi)$$

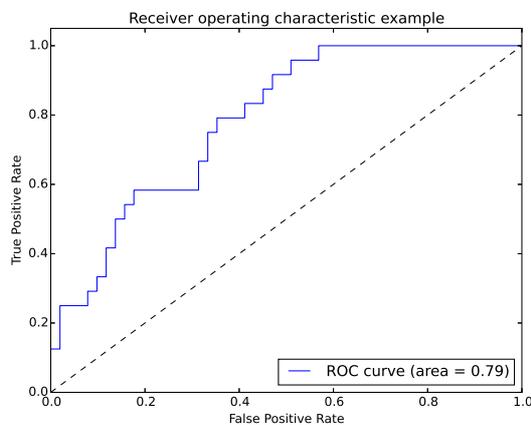
$$\text{Prec}_{\text{macro}}(\mathcal{X}, C, \psi) := \frac{1}{|C|} \sum_{i=1}^{|C|} \text{Prec}(\mathcal{X}, c_i, \psi)$$

As we can see from the definitions, micro-averaging focuses on the single *document* decisions whereas macro-averaging focuses on the *class* decisions.

#### 8.6.4.4 ROC statistics

A family of performance measures that have gained popularity over the last two decades are measures based on *receiver operating characteristics* (ROC) data (Fawcett, 2006, p. 861). These measures are based on plots of points in a two-dimensional *ROC space* in which the true positive rate (TPR) is plotted in the  $y$  dimension against the false positive rate (FPR) in the  $x$  dimension. The resulting plots are

comparable to *precision-recall graphs* commonly used in information retrieval evaluation, since both diagrams will contain information about the amount of false positive information (falsely assigned classification codes, non-relevant documents retrieved) that is generated together with a certain amount of true positive information (correctly assigned classification codes, relevant documents retrieved). The diagonal line in the ROC space from  $(0, 0)$  to  $(1, 1)$  corresponds to performance values that would be obtained from a classifier that randomly (and uniformly) assigns classification codes to documents (Fawcett, 2006, p. 863).



**Figure 8.2.** An example of a ROC curve, generated with scikit-learn and matplotlib.

For classifiers that produce real value outputs rather than binary decisions (such as Bayesian classifiers) it is common to generate ROC statistics over the entire test set ranked according to the output of the classifier. This will yield a monotonously increasing step function in ROC space called a *ROC curve* (see figure 8.2). One way of generating

summary statistics from the ROC curve for comparing different classification models is to compute the *area under the ROC curve* (AUC) as the sum of trapezoids or rectangles under the linearly interpolated graph between the ROC points.

## 8.7 Software used in this work

The scripts used for document preprocessing, term weighting, pre-computation of semantic kernels, as well as the general workflow of the experiments, have been written by the author of this thesis in the Python programming language. Some performance-critical parts of the code, involving computations on vectors and matrices, have utilized the NumPy and the SciPy libraries of the SciPy stack (see Jones et al., 2001–). All processes involving training, tuning, and testing of the SVM classification algorithm have been performed using the scikit-learn machine learning library (see Pedregosa et al., 2011). This library in turn contains and utilizes the LIBSVM (see Chang & Lin, 2011) library, developed at the National Taiwan University, and which consists of a variety of implementations of the SVM algorithm.

For the computations involving the random indexing algorithm a Java implementation published by Hassel (2013) has been used, slightly modified to adapt it to the experimental setup of this study. For the computation of the singular value decomposition (SVD) of term-by-document matrices the SVDLIBC library (see Rohde, 2007) has been used. This implementation uses the Lanczos method (see Lanczos, 1950; Golub et al., 1981) for finding the largest  $k$  singular values of the SVD factorization of sparse matrices.

## Chapter 9

# Results

In this chapter we present the results from the empirical part of this thesis, having the objective to investigate the classification performance of semantic SVM kernels induced by different measures for computing semantic relatedness between terms, as well as the performance of two term weighting schemes. In the presentation of the results below we will consistently use the following abbreviations for the methods for statistical semantics:

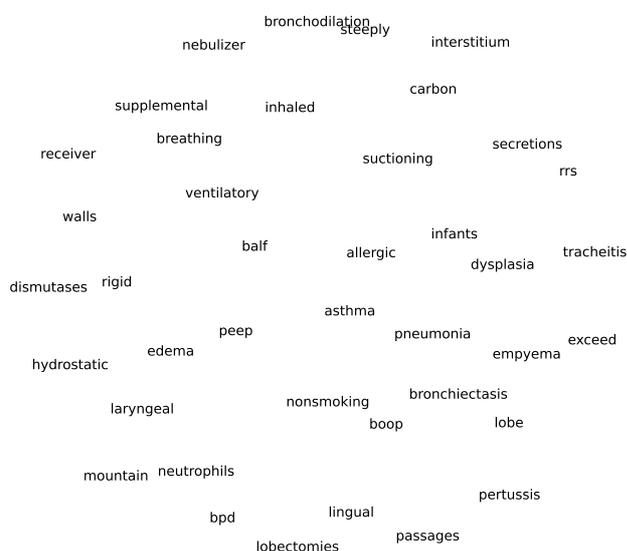
- pmi** Pointwise mutual information
- lsa- $m$**  Latent semantic analysis of dimensionality  $m$
- ri- $cm$**  Random indexing with constant weighting,  
window size  $m$
- ri- $mm$**  Random indexing with distance weighting,  
window size  $m$

The *latent semantic analysis* method has been computed over dimensionality  $m = 100$  and  $m = 300$ , which is denoted *lsa-100* and *lsa-300* respectively in the result tables. Further, the *random indexing* method has been used in two different variants: with context vectors weighted by distance (denoted by the symbol  $m$ ) and context vectors with no

distance weighting (denoted by the symbol  $c$ ). The number following the weighting code corresponds to the size of the left and the right analysis windows (see section 8.5.3). For example, the code *ri-m4* denotes random indexing used with context vectors weighted by distance, and with an analysis window of size 4. By *baseline* in the result presentation we refer to the ordinary linear kernel used as a reference baseline in these experiments.

The semantic similarity measures used in this study have been normalized to yield values between 0 and 1, where a value closer to 1 indicates a higher semantic similarity. The similarity values between any two terms  $k_i, k_j \in V$  can be transformed into *dissimilarity* values by the subtraction  $1 - sim(k_i, k_j)$ . For instance, a similarity value of 0.35 yields a dissimilarity value of 0.65. The symmetric matrix of dissimilarity can be visualized using a *manifold learning* algorithm such as *multidimensional scaling* (MDS). In figure 9.1 we have used metric MDS to visualize the word structure in a sample of terms from the class C08 in the Ohsumed reference collection. The semantic similarity measure used for this visualization is the *ri-m8* method.

The measures used to quantify and compare the classification performance of the kernels and the weighting schemes are primarily the  $F_1$  measure and the *area under the ROC curve* (AUC) measure, but also the precision and the recall scores are presented to give the complementary details. The performance scores are presented and discussed separately for each dataset. By *proportion* in the presentation of the results below we denote the proportion of *positive* examples used for training. As explained in section 8.6.3, the datasets for training the classifiers are sampled in such a way that an equal number of positive and negative examples are used in each set. Further, by *dimensionality* in the result tables we denote the number of features (words) used for training and testing.



**Figure 9.1.** Semantic map generated by multidimensional scaling.

Most of the evaluation figures in this chapter consist of the average performance values over all selected classes in the reference collections, so called *macro-average* scores (see section 8.6.4.3). Since each classification performance test has been performed in  $n = 30$  samples from the test collections we are also able to compute statistical error margins (and consequently also confidence intervals) for the average score over all samples. We assume that the average performance scores are normally distributed over the document samples. Since the standard deviation of the performance score has been estimated from the test samples we have computed the error margins using a  $t$  distri-

bution with an  $n - 1$  degree of freedom. Let  $\Phi_t^{-1}(df, p)$  denote the *inverse cumulative distribution function* (also called the quantile function) of a  $t$  distribution defined for  $df$  degrees of freedom. Then the margin of error (denoted  $ME$ ) for a performance score is calculated according to

$$ME = \Phi_t^{-1}(n - 1, (\alpha + 1)/2) \cdot \frac{s}{\sqrt{n}}$$

where  $\alpha$  denotes the stipulated confidence level and  $s$  the sample standard deviation. The quotient  $s/\sqrt{n}$  is also known as the *standard error of the mean*.

## 9.1 Results for the Reuters-21578 collection

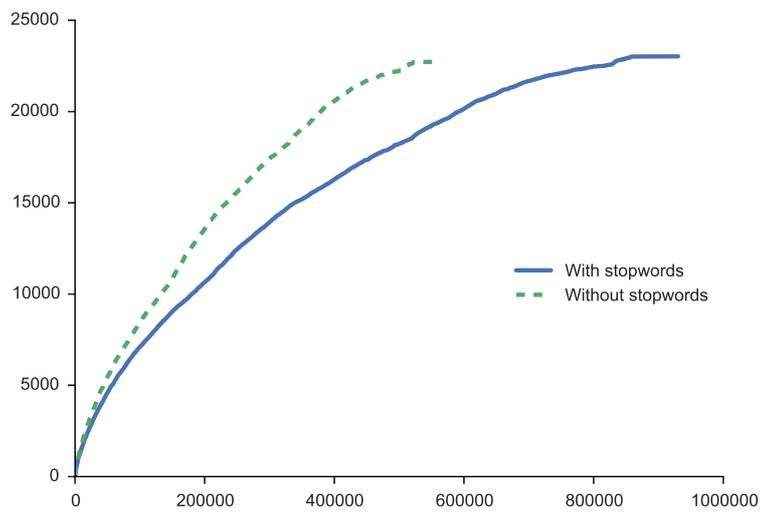
The Reuters-21578 reference collection is described in section 8.2.1. Below we present the document and word statistics pertaining to this collection, both before and after stopword removal. We also present the estimated parameters for the Heaps' distribution (see section 5.3.1). According to Heap's law it holds that for a set of documents containing a total amount of  $n$  tokens, the number of unique words in this document set is given by a function on the form

$$f(n) = kn^\beta \quad (9.1)$$

where  $k$  and  $\beta$  are free parameters. We have estimated these parameters (see e.g. Manning et al., 2008, p. 82) using a non-linear least squares fit to the equation (9.1) over the accumulation of documents in the training and test collections respectively. Below we present the statistical properties of the training and the tests respectively of the Reuters-21578 collection.

	Training set	Test set
Number of documents:	7193	2787
Vocabulary size (with stopwords):	23023	14304
Vocabulary size (without stopwords):	22722	14010
Token-to-type ratio (without stopwords):	1.6270	1.6361
Token-to-type ratio (with stopwords):	1.5004	1.5362
Heaps parameters (with stopwords):	$k = 15.2303,$ $\beta = 0.5381$	$k = 6.3102,$ $\beta = 0.6140$

The statistical relationship between the number of *tokens* (word occurrences) and the number of *types* (unique words) is visualized in figure 9.2.



**Figure 9.2.** The number of types versus the number of tokens in the Reuters-21578 collection.

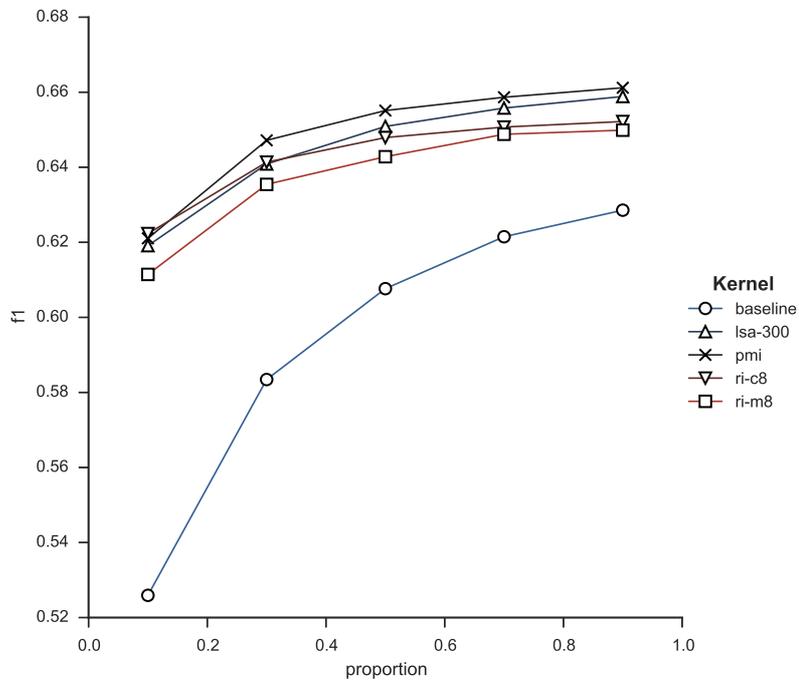
### 9.1.1 Using the *tf-idf* weighting scheme

As we can see in table 9.1, the semantic kernels consistently yield higher  $F_1$  and AUC scores, as compared to the baseline kernel, when 100 features have been used for training and classification. The difference between the baseline kernel and the semantic kernels is higher when small proportions of the training set has been used. Another observable trend in the data is that the classification performance increases with the proportion of documents used for training. The diagram in figure 9.3 also demonstrates that the semantic kernels consistently perform better than the baseline linear kernel, in particular when a small proportion of training documents have been used. As an example of the degree of variability among the performance scores we investigate the class *ship* and provide the estimated error margins for the  $F_1$  and the AUC scores, which are presented in table 9.2. The error margin is in general  $< 0.01$  on a confidence level of 95%.

When 300 features have been used for training and classification we observe a slight advantage for the LSA kernels, with the LSA-300 and PMI being the highest ranked (see table 9.3). Another noticeable result is that the baseline kernel yields comparatively high recall values, especially when a small proportion of the training documents are used. We also notice that the precision is consistently higher for the semantic kernels, which means that the positive decisions made by the classifiers based on the semantic kernels are more accurate.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.4100	0.9462	0.5259	0.9467
	lsa-100	0.4909	0.9216	0.6079	0.9620
	lsa-300	0.4998	0.9251	0.6191	0.9638
	pmi	0.5030	0.9206	0.6211	0.9626
	ri-c2	0.4936	0.9184	0.6119	0.9610
	ri-c4	0.4974	0.9202	0.6165	0.9624
	ri-c8	0.5038	0.9190	0.6223	0.9627
	ri-m2	0.4871	0.9134	0.6048	0.9591
	ri-m4	0.4915	0.9197	0.6111	0.9611
ri-m8	0.4924	0.9201	0.6114	0.9615	
0.5	baseline	0.4823	0.9593	0.6077	0.9635
	lsa-100	0.5174	0.9523	0.6415	0.9698
	lsa-300	0.5242	0.9565	0.6509	0.9717
	pmi	0.5316	0.9515	0.6551	0.9712
	ri-c2	0.5199	0.9506	0.6434	0.9698
	ri-c4	0.5177	0.9493	0.6410	0.9693
	ri-c8	0.5232	0.9535	0.6479	0.9702
	ri-m2	0.5146	0.9493	0.6381	0.9688
	ri-m4	0.5228	0.9515	0.6462	0.9699
ri-m8	0.5196	0.9495	0.6428	0.9693	
0.9	baseline	0.5029	0.9599	0.6286	0.9670
	lsa-100	0.5250	0.9580	0.6498	0.9712
	lsa-300	0.5319	0.9593	0.6589	0.9729
	pmi	0.5374	0.9567	0.6612	0.9726
	ri-c2	0.5294	0.9565	0.6527	0.9713
	ri-c4	0.5254	0.9538	0.6484	0.9706
	ri-c8	0.5275	0.9573	0.6522	0.9714
	ri-m2	0.5249	0.9537	0.6483	0.9704
	ri-m4	0.5289	0.9560	0.6524	0.9711
ri-m8	0.5265	0.9542	0.6499	0.9704	

**Table 9.1.** Macro-average scores for the *tf-idf* weighting scheme.  
Dimensionality = 100.



**Figure 9.3.** The average F1 score for a selection of semantic kernels obtained by sampling different proportions of the Reuters reference collection, using the *tf-idf* weighting scheme. Dimensionality = 100.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.2650	0.8727	$0.4008 \pm 0.0161$	$0.8229 \pm 0.0196$
	lsa-100	0.4809	0.8453	$0.6074 \pm 0.0140$	$0.9081 \pm 0.0065$
	lsa-300	0.4938	0.8213	$0.6110 \pm 0.0158$	$0.9095 \pm 0.0075$
	pmi	0.5174	0.8127	$0.6267 \pm 0.0161$	$0.9115 \pm 0.0139$
	ri-c2	0.4596	0.8944	$0.6055 \pm 0.0081$	$0.9164 \pm 0.0027$
	ri-c4	0.4801	0.8753	$0.6172 \pm 0.0106$	$0.9135 \pm 0.0039$
	ri-c8	0.5046	0.8461	$0.6289 \pm 0.0070$	$0.9116 \pm 0.0047$
	ri-m2	0.4697	0.8764	$0.6095 \pm 0.0137$	$0.9137 \pm 0.0038$
	ri-m4	0.4601	0.8869	$0.6027 \pm 0.0169$	$0.9139 \pm 0.0039$
ri-m8	0.4715	0.8850	$0.6123 \pm 0.0151$	$0.9129 \pm 0.0040$	
0.5	baseline	0.3377	0.9483	$0.4971 \pm 0.0097$	$0.8893 \pm 0.0076$
	lsa-100	0.5045	0.8843	$0.6412 \pm 0.0099$	$0.9254 \pm 0.0021$
	lsa-300	0.5161	0.9007	$0.6555 \pm 0.0059$	$0.9302 \pm 0.0014$
	pmi	0.5300	0.8835	$0.6620 \pm 0.0075$	$0.9291 \pm 0.0017$
	ri-c2	0.4739	0.9393	$0.6293 \pm 0.0081$	$0.9287 \pm 0.0011$
	ri-c4	0.4843	0.9225	$0.6339 \pm 0.0088$	$0.9268 \pm 0.0013$
	ri-c8	0.5110	0.9217	$0.6571 \pm 0.0040$	$0.9277 \pm 0.0010$
	ri-m2	0.4702	0.9382	$0.6257 \pm 0.0093$	$0.9288 \pm 0.0007$
	ri-m4	0.4869	0.9210	$0.6359 \pm 0.0104$	$0.9275 \pm 0.0016$
ri-m8	0.4888	0.9206	$0.6377 \pm 0.0099$	$0.9266 \pm 0.0018$	
0.9	baseline	0.3895	0.9322	$0.5486 \pm 0.0095$	$0.8966 \pm 0.0047$
	lsa-100	0.5207	0.8846	$0.6551 \pm 0.0049$	$0.9273 \pm 0.0013$
	lsa-300	0.5245	0.8959	$0.6612 \pm 0.0034$	$0.9320 \pm 0.0011$
	pmi	0.5370	0.8854	$0.6679 \pm 0.0061$	$0.9310 \pm 0.0016$
	ri-c2	0.5030	0.9251	$0.6512 \pm 0.0060$	$0.9302 \pm 0.0012$
	ri-c4	0.5003	0.9090	$0.6447 \pm 0.0073$	$0.9277 \pm 0.0015$
	ri-c8	0.5147	0.9150	$0.6584 \pm 0.0043$	$0.9292 \pm 0.0008$
	ri-m2	0.4899	0.9277	$0.6409 \pm 0.0058$	$0.9311 \pm 0.0006$
	ri-m4	0.5010	0.9075	$0.6451 \pm 0.0074$	$0.9263 \pm 0.0018$
ri-m8	0.4978	0.9030	$0.6413 \pm 0.0088$	$0.9258 \pm 0.0020$	

**Table 9.2.** Average scores with error margins on the confidence level 95% for the class *ship*. Dimensionality = 100.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.3828	0.9760	0.4956	0.9749
	lsa-100	0.4703	0.9335	0.5873	0.9726
	lsa-300	0.4858	0.9443	0.6055	0.9766
	pmi	0.4813	0.9378	0.6006	0.9746
	ri-c2	0.4528	0.9420	0.5751	0.9731
	ri-c4	0.4721	0.9474	0.5936	0.9749
	ri-c8	0.4733	0.9475	0.5934	0.9748
	ri-m2	0.4558	0.9391	0.5775	0.9724
	ri-m4	0.4612	0.9396	0.5820	0.9730
ri-m8	0.4624	0.9432	0.5837	0.9739	
0.5	baseline	0.4596	0.9788	0.5851	0.9804
	lsa-100	0.5004	0.9737	0.6272	0.9812
	lsa-300	0.5129	0.9760	0.6403	0.9825
	pmi	0.5101	0.9719	0.6365	0.9811
	ri-c2	0.4948	0.9659	0.6209	0.9800
	ri-c4	0.5013	0.9709	0.6273	0.9807
	ri-c8	0.5064	0.9708	0.6310	0.9803
	ri-m2	0.5002	0.9661	0.6261	0.9801
	ri-m4	0.4990	0.9679	0.6247	0.9804
ri-m8	0.5012	0.9682	0.6269	0.9803	
0.9	baseline	0.4835	0.9798	0.6105	0.9819
	lsa-100	0.5093	0.9796	0.6374	0.9821
	lsa-300	0.5180	0.9796	0.6461	0.9833
	pmi	0.5179	0.9794	0.6455	0.9823
	ri-c2	0.5075	0.9721	0.6338	0.9814
	ri-c4	0.5110	0.9757	0.6374	0.9817
	ri-c8	0.5139	0.9773	0.6398	0.9813
	ri-m2	0.5124	0.9714	0.6387	0.9815
	ri-m4	0.5090	0.9752	0.6352	0.9815
ri-m8	0.5135	0.9741	0.6397	0.9816	

**Table 9.3.** Macro-average scores for the *tf-idf* weighting scheme.  
Dimensionality = 300.

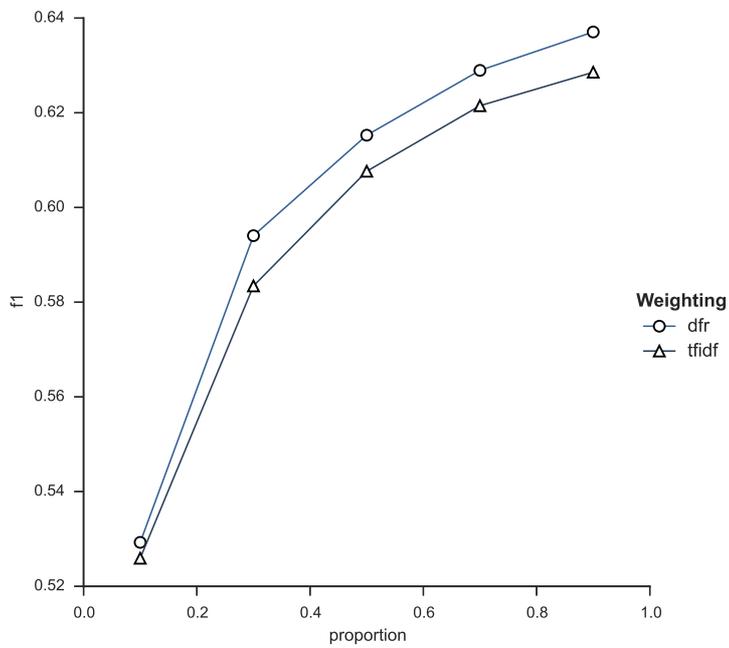
### 9.1.2 Using the *dfr* weighting scheme

In table 9.4 we note that the *divergence from randomness* weighting scheme outperforms the traditional *tf-idf* weighting scheme for all proportions of the training set, as we can see in figure 9.4. We also notice that the semantic kernels manage to increase the  $F_1$  and the AUC, as compared to the baseline kernel. By the visual comparison in figure 9.4 of the performance of the *tf-idf* and the *dfr* weighting schemes for the baseline kernel we clearly observe that the *dfr* scheme consistently yields a higher  $F_1$  score. We also note that the difference becomes larger when a larger proportion of the training documents are used.

Also for the *dfr* weighting scheme we observe that the advantage is less noticeable for the semantic kernels when a higher number of features have been used. The performance figures obtained from this combination of parameters are generally higher than the corresponding measurement values for the *tf-idf* weighting scheme, as we can see in table 9.5.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.4138	0.9543	0.5293	0.9510
	lsa-100	0.4955	0.9244	0.6146	0.9625
	lsa-300	0.4952	0.9212	0.6133	0.9625
	pmi	0.4983	0.9215	0.6160	0.9615
	ri-c2	0.4853	0.9275	0.6062	0.9607
	ri-c4	0.4884	0.9197	0.6069	0.9601
	ri-c8	0.4970	0.9222	0.6153	0.9621
	ri-m2	0.4836	0.9275	0.6044	0.9600
	ri-m4	0.4835	0.9257	0.6050	0.9605
ri-m8	0.4842	0.9250	0.6051	0.9610	
0.5	baseline	0.4883	0.9679	0.6153	0.9634
	lsa-100	0.5159	0.9598	0.6430	0.9712
	lsa-300	0.5246	0.9627	0.6525	0.9725
	pmi	0.5264	0.9605	0.6512	0.9714
	ri-c2	0.5159	0.9620	0.6421	0.9710
	ri-c4	0.5149	0.9564	0.6385	0.9698
	ri-c8	0.5173	0.9601	0.6423	0.9705
	ri-m2	0.5150	0.9595	0.6419	0.9706
	ri-m4	0.5163	0.9608	0.6429	0.9708
ri-m8	0.5138	0.9605	0.6400	0.9705	
0.9	baseline	0.5088	0.9695	0.6371	0.9656
	lsa-100	0.5234	0.9630	0.6506	0.9725
	lsa-300	0.5326	0.9673	0.6608	0.9737
	pmi	0.5322	0.9649	0.6579	0.9729
	ri-c2	0.5239	0.9670	0.6507	0.9725
	ri-c4	0.5213	0.9628	0.6456	0.9712
	ri-c8	0.5226	0.9631	0.6475	0.9716
	ri-m2	0.5243	0.9647	0.6521	0.9721
	ri-m4	0.5245	0.9656	0.6519	0.9722
ri-m8	0.5195	0.9651	0.6465	0.9718	

**Table 9.4.** Macro-average scores for the *dfr* weighting scheme.  
Dimensionality = 100.



**Figure 9.4.** The average F1 score obtained in the Reuters-21578 reference collection for the baseline kernel using the *tfidf* and the *dfr* weighting schemes. Dimensionality = 100.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.4221	0.9776	0.5448	0.9774
	lsa-100	0.4764	0.9438	0.5970	0.9750
	lsa-300	0.4799	0.9520	0.6010	0.9769
	pmi	0.4844	0.9447	0.6047	0.9749
	ri-c2	0.4562	0.9491	0.5791	0.9736
	ri-c4	0.4643	0.9439	0.5863	0.9734
	ri-c8	0.4721	0.9510	0.5943	0.9755
	ri-m2	0.4582	0.9504	0.5811	0.9739
	ri-m4	0.4575	0.9486	0.5805	0.9738
ri-m8	0.4581	0.9574	0.5817	0.9753	
0.5	baseline	0.4798	0.9836	0.6095	0.9821
	lsa-100	0.5003	0.9787	0.6287	0.9824
	lsa-300	0.5031	0.9805	0.6325	0.9830
	pmi	0.5084	0.9764	0.6355	0.9816
	ri-c2	0.4946	0.9715	0.6215	0.9809
	ri-c4	0.4945	0.9735	0.6225	0.9810
	ri-c8	0.4981	0.9753	0.6261	0.9809
	ri-m2	0.4992	0.9761	0.6273	0.9812
	ri-m4	0.4943	0.9729	0.6217	0.9806
ri-m8	0.4923	0.9775	0.6198	0.9814	
0.9	baseline	0.5014	0.9841	0.6324	0.9832
	lsa-100	0.5046	0.9833	0.6339	0.9832
	lsa-300	0.5105	0.9832	0.6405	0.9838
	pmi	0.5164	0.9821	0.6445	0.9829
	ri-c2	0.5094	0.9781	0.6369	0.9825
	ri-c4	0.5034	0.9799	0.6321	0.9825
	ri-c8	0.5028	0.9787	0.6312	0.9818
	ri-m2	0.5143	0.9805	0.6432	0.9828
	ri-m4	0.5079	0.9792	0.6368	0.9825
ri-m8	0.5059	0.9805	0.6344	0.9826	

**Table 9.5.** Macro-average scores for the *dfr* weighting scheme.  
Dimensionality = 300.

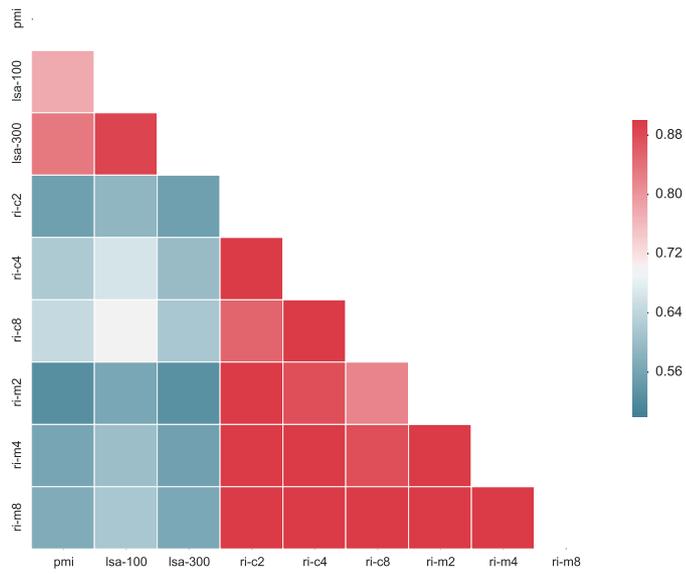
### 9.1.3 Comparison between the semantic kernels

In the evaluation result above we have noticed that the semantic kernels consistently perform better than the baseline kernel, as measured by the  $F_1$  and the AUC scores. We have also noticed that the LSA-300 and the PMI kernels perform somewhat better than the other semantic kernels. This naturally raises the question whether this difference can be traced to the term-term similarity values generated by the underlying semantic similarity measures. In table 9.6 we show the pairwise cosine similarity between the symmetric similarity matrices underlying semantic kernels.

	pmi	lsa-100	lsa-300	ri-c2	ri-c4	ri-c8	ri-m2	ri-m4	ri-m8
pmi	1.000	0.777	0.831	0.553	0.622	0.646	0.527	0.563	0.573
lsa-100	0.777	1.000	0.889	0.590	0.663	0.698	0.565	0.602	0.617
lsa-300	0.831	0.889	1.000	0.554	0.597	0.616	0.530	0.556	0.566
ri-c2	0.553	0.590	0.554	1.000	0.909	0.854	0.961	0.963	0.963
ri-c4	0.622	0.663	0.597	0.909	1.000	0.965	0.878	0.931	0.945
ri-c8	0.646	0.698	0.616	0.854	0.965	1.000	0.819	0.878	0.900
ri-m2	0.527	0.565	0.530	0.961	0.878	0.819	1.000	0.967	0.960
ri-m4	0.563	0.602	0.556	0.963	0.931	0.878	0.967	1.000	0.978
ri-m8	0.573	0.617	0.566	0.963	0.945	0.900	0.960	0.978	1.000

**Table 9.6.** The cosine similarity of the semantic kernels in the class *wheat*.

When we inspect the heat map in figure 9.5 we notice that there are two discernible groups among the kernels: the LSA kernels together with the PMI kernel, and the RI kernels respectively. This result is in line with the observation above that the PMI kernel and the LSA-300 kernel yield a comparable performance.



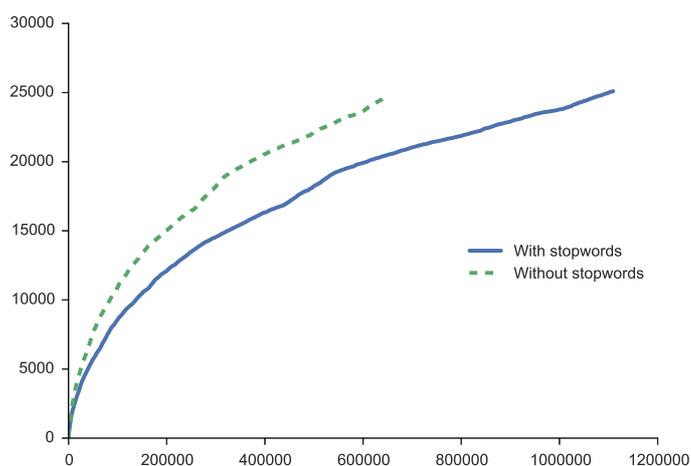
**Figure 9.5.** A heat map visualizing the cosine similarity between the semantic kernels.

## 9.2 Results for the Ohsumed collection

The Ohsumed reference collection is described in section 8.2.2. Below we present the statistical properties of the training and the tests respectively of the Ohsumed collection.

	Training set	Test set
Number of documents:	6467	7807
Vocabulary size (with stopwords):	25098	27509
Vocabulary size (without stopwords):	24801	27211
Token-to-type ratio (without stopwords):	1.8049	1.8158
Token-to-type ratio (with stopwords):	1.5400	1.5437
Heaps parameters (with stopwords):	$k = 58.6717,$ $\beta = 0.4360$	$k = 55.5149,$ $\beta = 0.4396$

The statistical relationship between the number of tokens and the number of types is visualized in figure 9.6.



**Figure 9.6.** The number of types versus the number of tokens in the Ohsumed collection.

### 9.2.1 Using the *tf-idf* weighting scheme

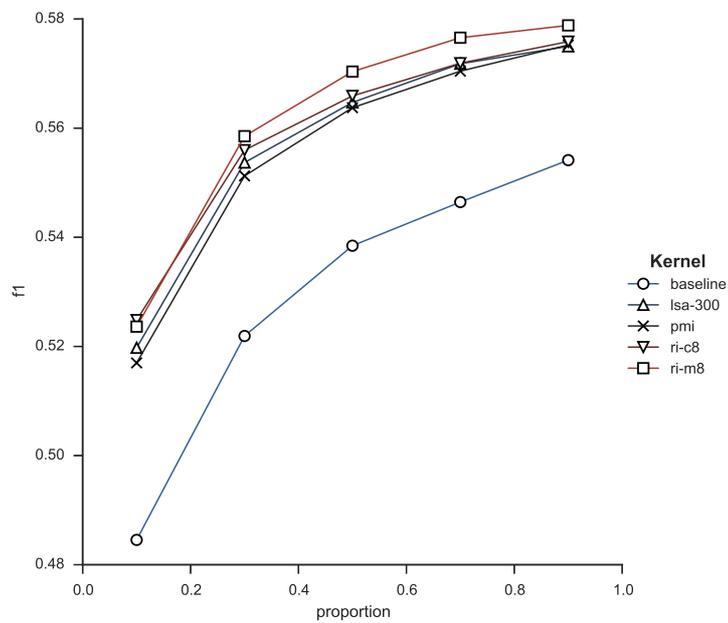
We notice a similar trend in this reference collection, as compared to the Reuters-21578 collection, namely that the recall value for the baseline kernel is higher than that of the semantic kernels, whereas the precision is noticeably lower. There is a noticeable advantage for the semantic kernels with respect to the  $F_1$  and AUC performance measures, with a slight edge for the kernels based on the random indexing method (which we can observe in table 9.7 as well as in figure 9.7). The diagram in figure 9.7 also demonstrates clearly that the semantic kernels consistently perform better than the baseline linear kernel. An interesting difference compared to the evaluation performed in the Reuters-21578 collection is that the kernels based on random indexing perform somewhat better than the other semantic kernels.

As an example of the variability among the performance scores we investigate the class *C08*, together with the estimated error margins for the  $F_1$  and the AUC scores (see table 9.8). Also for this class the error margin is in general less than, and in a few cases slightly above, 0.01 on a confidence level of 95%.

Also when 300 features are used for training and classification we observe that the baseline strategy yields a higher recall, in particular when a small proportion of the training documents is used (see table 9.9). It is also noticeable that the semantic kernels consistently yield high precision values, which also gives them a slight edge with respect to the  $F_1$  and the AUC scores.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.3572	0.8070	0.4845	0.7754
	lsa-100	0.4072	0.7429	0.5192	0.8003
	lsa-300	0.4057	0.7489	0.5197	0.7987
	pmi	0.4053	0.7397	0.5170	0.7959
	ri-c2	0.4070	0.7495	0.5199	0.8003
	ri-c4	0.4154	0.7286	0.5212	0.7990
	ri-c8	0.4234	0.7233	0.5248	0.8022
	ri-m2	0.4052	0.7500	0.5177	0.7987
	ri-m4	0.4112	0.7417	0.5222	0.8024
	ri-m8	0.4123	0.7456	0.5236	0.8056
0.5	baseline	0.4027	0.8487	0.5385	0.8222
	lsa-100	0.4338	0.8107	0.5595	0.8409
	lsa-300	0.4378	0.8204	0.5647	0.8433
	pmi	0.4371	0.8167	0.5637	0.8410
	ri-c2	0.4406	0.8186	0.5665	0.8421
	ri-c4	0.4404	0.8121	0.5650	0.8430
	ri-c8	0.4440	0.8057	0.5659	0.8438
	ri-m2	0.4426	0.8182	0.5678	0.8431
	ri-m4	0.4439	0.8166	0.5696	0.8434
	ri-m8	0.4436	0.8212	0.5704	0.8450
0.9	baseline	0.4182	0.8525	0.5541	0.8342
	lsa-100	0.4379	0.8300	0.5676	0.8491
	lsa-300	0.4445	0.8372	0.5750	0.8528
	pmi	0.4446	0.8362	0.5752	0.8505
	ri-c2	0.4472	0.8358	0.5769	0.8513
	ri-c4	0.4473	0.8325	0.5764	0.8530
	ri-c8	0.4483	0.8279	0.5759	0.8529
	ri-m2	0.4489	0.8354	0.5783	0.8522
	ri-m4	0.4500	0.8352	0.5799	0.8520
	ri-m8	0.4485	0.8376	0.5788	0.8529

**Table 9.7.** Macro-average scores for the *tf-idf* weighting scheme. Dimensionality = 100.



**Figure 9.7.** The average  $F_1$  score for a selection of semantic kernels obtained by sampling different proportions of the Ohsumed reference collection, using the *tf-idf* weighting scheme. Dimensionality = 100.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.3479	0.6563	0.4475 $\pm$ 0.0132	0.6740 $\pm$ 0.0092
	lsa-100	0.3800	0.6449	0.4747 $\pm$ 0.0084	0.7080 $\pm$ 0.0082
	lsa-300	0.3647	0.6494	0.4656 $\pm$ 0.0102	0.6857 $\pm$ 0.0106
	pmi	0.3906	0.6647	0.4901 $\pm$ 0.0067	0.7201 $\pm$ 0.0068
	ri-c2	0.3682	0.6324	0.4633 $\pm$ 0.0117	0.6866 $\pm$ 0.0113
	ri-c4	0.4010	0.6545	0.4943 $\pm$ 0.0087	0.7296 $\pm$ 0.0086
	ri-c8	0.4202	0.6162	0.4955 $\pm$ 0.0089	0.7312 $\pm$ 0.0082
	ri-m2	0.3667	0.6565	0.4683 $\pm$ 0.0104	0.6959 $\pm$ 0.0113
	ri-m4	0.3689	0.6382	0.4646 $\pm$ 0.0098	0.7001 $\pm$ 0.0107
ri-m8	0.3899	0.6500	0.4842 $\pm$ 0.0103	0.7192 $\pm$ 0.0099	
0.5	baseline	0.3866	0.7636	0.5127 $\pm$ 0.0064	0.7330 $\pm$ 0.0045
	lsa-100	0.3921	0.7001	0.5019 $\pm$ 0.0030	0.7410 $\pm$ 0.0036
	lsa-300	0.4119	0.7105	0.5208 $\pm$ 0.0047	0.7512 $\pm$ 0.0041
	pmi	0.4120	0.7333	0.5271 $\pm$ 0.0043	0.7570 $\pm$ 0.0031
	ri-c2	0.4156	0.6950	0.5195 $\pm$ 0.0069	0.7437 $\pm$ 0.0061
	ri-c4	0.4270	0.7068	0.5319 $\pm$ 0.0048	0.7638 $\pm$ 0.0041
	ri-c8	0.4333	0.6846	0.5300 $\pm$ 0.0037	0.7609 $\pm$ 0.0037
	ri-m2	0.4148	0.7254	0.5273 $\pm$ 0.0058	0.7586 $\pm$ 0.0053
	ri-m4	0.4138	0.7054	0.5211 $\pm$ 0.0060	0.7534 $\pm$ 0.0052
ri-m8	0.4191	0.7286	0.5316 $\pm$ 0.0057	0.7636 $\pm$ 0.0048	
0.9	baseline	0.4086	0.7657	0.5321 $\pm$ 0.0037	0.7507 $\pm$ 0.0033
	lsa-100	0.3926	0.7370	0.5117 $\pm$ 0.0032	0.7518 $\pm$ 0.0022
	lsa-300	0.4196	0.7540	0.5388 $\pm$ 0.0042	0.7712 $\pm$ 0.0039
	pmi	0.4174	0.7567	0.5377 $\pm$ 0.0032	0.7650 $\pm$ 0.0028
	ri-c2	0.4233	0.7319	0.5358 $\pm$ 0.0049	0.7630 $\pm$ 0.0039
	ri-c4	0.4342	0.7384	0.5465 $\pm$ 0.0029	0.7782 $\pm$ 0.0025
	ri-c8	0.4371	0.7248	0.5446 $\pm$ 0.0028	0.7744 $\pm$ 0.0024
	ri-m2	0.4222	0.7525	0.5404 $\pm$ 0.0047	0.7707 $\pm$ 0.0036
	ri-m4	0.4245	0.7403	0.5390 $\pm$ 0.0044	0.7678 $\pm$ 0.0031
ri-m8	0.4246	0.7532	0.5426 $\pm$ 0.0022	0.7753 $\pm$ 0.0025	

**Table 9.8.** Average scores with error margins on the confidence level 95% for the class *C08*. Dimensionality = 100.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.3044	0.8582	0.4317	0.8692
	lsa-100	0.3532	0.7972	0.4831	0.8726
	lsa-300	0.3635	0.7938	0.4932	0.8762
	pmi	0.3601	0.7922	0.4885	0.8764
	ri-c2	0.3482	0.7880	0.4759	0.8693
	ri-c4	0.3605	0.7837	0.4873	0.8727
	ri-c8	0.3629	0.7825	0.4895	0.8726
	ri-m2	0.3500	0.7912	0.4780	0.8726
	ri-m4	0.3527	0.7833	0.4797	0.8710
ri-m8	0.3519	0.7767	0.4780	0.8696	
0.5	baseline	0.3511	0.8867	0.4918	0.8995
	lsa-100	0.3834	0.8694	0.5257	0.9065
	lsa-300	0.3858	0.8699	0.5285	0.9069
	pmi	0.3840	0.8728	0.5267	0.9066
	ri-c2	0.3837	0.8621	0.5245	0.9035
	ri-c4	0.3900	0.8622	0.5310	0.9056
	ri-c8	0.3858	0.8641	0.5275	0.9044
	ri-m2	0.3870	0.8631	0.5280	0.9061
	ri-m4	0.3876	0.8568	0.5276	0.9041
ri-m8	0.3873	0.8601	0.5278	0.9051	
0.9	baseline	0.3595	0.8982	0.5032	0.9056
	lsa-100	0.3909	0.8803	0.5353	0.9121
	lsa-300	0.3924	0.8837	0.5376	0.9125
	pmi	0.3919	0.8872	0.5372	0.9127
	ri-c2	0.3943	0.8767	0.5376	0.9103
	ri-c4	0.4009	0.8769	0.5446	0.9123
	ri-c8	0.3933	0.8785	0.5377	0.9107
	ri-m2	0.3979	0.8757	0.5411	0.9123
	ri-m4	0.3988	0.8713	0.5414	0.9113
ri-m8	0.3980	0.8756	0.5412	0.9119	

**Table 9.9.** Macro-average scores for the *tf-idf* weighting scheme.  
Dimensionality = 300.

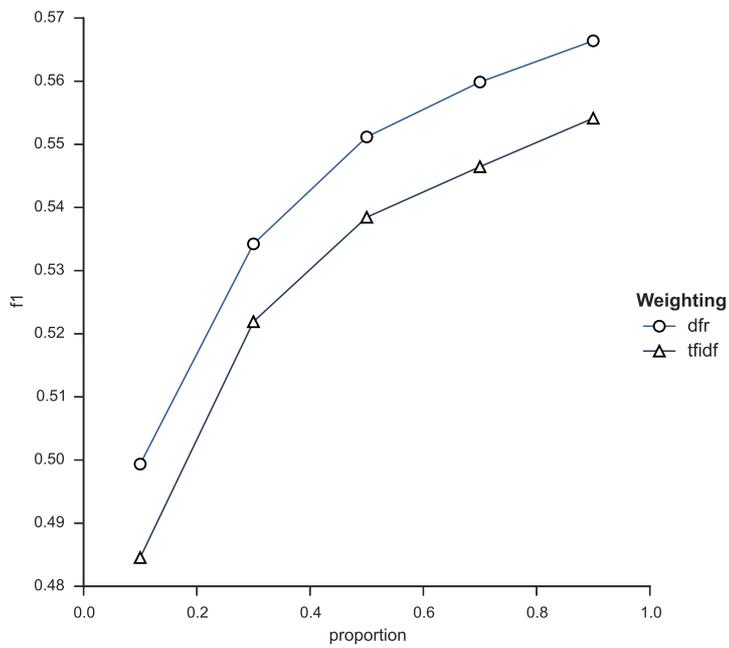
### 9.2.2 Using the *dfr* weighting scheme

We continue to observe that the *dfr* scheme slightly outperforms the traditional *tf-idf* weighting scheme for all proportions of the training set with respect to the  $F_1$  performance measure, as we can see in table 9.10 as well as in figure 9.8. We also notice that the semantic kernels manage to increase the  $F_1$  and the AUC, especially when small proportions of the training set is used, as compared to the baseline kernel. The LSA-300 and the PMI kernels appear to perform best for this feature configuration.

When 300 features are used for training and classification we observe that the precision decreases, whereas the recall increases, as compared to the corresponding performance scores when 100 features are used (see table 9.11). An interesting consequence is that the  $F_1$  score is overall lower for 300 features than for 100 features, whereas the AUC score is generally *higher* for 300 features. This reflects the higher probability of the classifier based on 300 features to yield a correct decision with respect to the positive examples, as compared to yielding a correct decision for the negative examples.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.3680	0.8155	0.4994	0.7811
	lsa-100	0.4177	0.7524	0.5302	0.8099
	lsa-300	0.4162	0.7480	0.5280	0.8041
	pmi	0.4174	0.7415	0.5283	0.8008
	ri-c2	0.4226	0.7575	0.5347	0.8129
	ri-c4	0.4280	0.7354	0.5348	0.8088
	ri-c8	0.4275	0.7364	0.5342	0.8075
	ri-m2	0.4208	0.7524	0.5333	0.8074
	ri-m4	0.4191	0.7496	0.5300	0.8078
ri-m8	0.4222	0.7568	0.5355	0.8108	
0.5	baseline	0.4142	0.8580	0.5512	0.8265
	lsa-100	0.4446	0.8190	0.5701	0.8478
	lsa-300	0.4418	0.8268	0.5693	0.8470
	pmi	0.4466	0.8161	0.5722	0.8450
	ri-c2	0.4536	0.8239	0.5794	0.8491
	ri-c4	0.4568	0.8085	0.5794	0.8477
	ri-c8	0.4506	0.8159	0.5747	0.8480
	ri-m2	0.4519	0.8193	0.5774	0.8467
	ri-m4	0.4469	0.8256	0.5732	0.8472
ri-m8	0.4505	0.8251	0.5777	0.8474	
0.9	baseline	0.4288	0.8645	0.5664	0.8380
	lsa-100	0.4494	0.8382	0.5789	0.8550
	lsa-300	0.4476	0.8442	0.5789	0.8557
	pmi	0.4533	0.8361	0.5831	0.8550
	ri-c2	0.4571	0.8407	0.5871	0.8566
	ri-c4	0.4626	0.8284	0.5898	0.8567
	ri-c8	0.4566	0.8333	0.5846	0.8563
	ri-m2	0.4580	0.8353	0.5869	0.8552
	ri-m4	0.4515	0.8458	0.5824	0.8551
ri-m8	0.4569	0.8416	0.5877	0.8552	

**Table 9.10.** Macro-average scores for the *dfr* weighting scheme.  
Dimensionality = 100.



**Figure 9.8.** The average  $F_1$  score in the Ohsumed collection for the *tf-idf* and the *dfr* weighting schemes. Dimensionality = 100.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.3097	0.8593	0.4375	0.8734
	lsa-100	0.3529	0.8130	0.4845	0.8792
	lsa-300	0.3681	0.8002	0.4981	0.8818
	pmi	0.3690	0.7914	0.4980	0.8803
	ri-c2	0.3588	0.7843	0.4853	0.8729
	ri-c4	0.3664	0.7903	0.4943	0.8782
	ri-c8	0.3685	0.7856	0.4956	0.8777
	ri-m2	0.3599	0.7898	0.4875	0.8773
	ri-m4	0.3653	0.7878	0.4930	0.8773
ri-m8	0.3620	0.7877	0.4900	0.8764	
0.5	baseline	0.3579	0.8963	0.5023	0.9029
	lsa-100	0.3855	0.8753	0.5285	0.9094
	lsa-300	0.3896	0.8762	0.5331	0.9110
	pmi	0.3860	0.8761	0.5294	0.9092
	ri-c2	0.3937	0.8608	0.5343	0.9058
	ri-c4	0.3954	0.8672	0.5373	0.9092
	ri-c8	0.3950	0.8678	0.5371	0.9091
	ri-m2	0.3936	0.8651	0.5348	0.9093
	ri-m4	0.3946	0.8631	0.5356	0.9078
ri-m8	0.3958	0.8671	0.5378	0.9091	
0.9	baseline	0.3740	0.8992	0.5201	0.9091
	lsa-100	0.3938	0.8867	0.5391	0.9144
	lsa-300	0.3965	0.8879	0.5421	0.9160
	pmi	0.3946	0.8910	0.5405	0.9155
	ri-c2	0.4029	0.8784	0.5466	0.9127
	ri-c4	0.4034	0.8813	0.5477	0.9149
	ri-c8	0.4039	0.8825	0.5486	0.9152
	ri-m2	0.4029	0.8796	0.5467	0.9151
	ri-m4	0.4032	0.8774	0.5469	0.9140
ri-m8	0.4051	0.8818	0.5497	0.9150	

**Table 9.11.** Macro-average scores for the *dfr* weighting scheme.  
Dimensionality = 300.

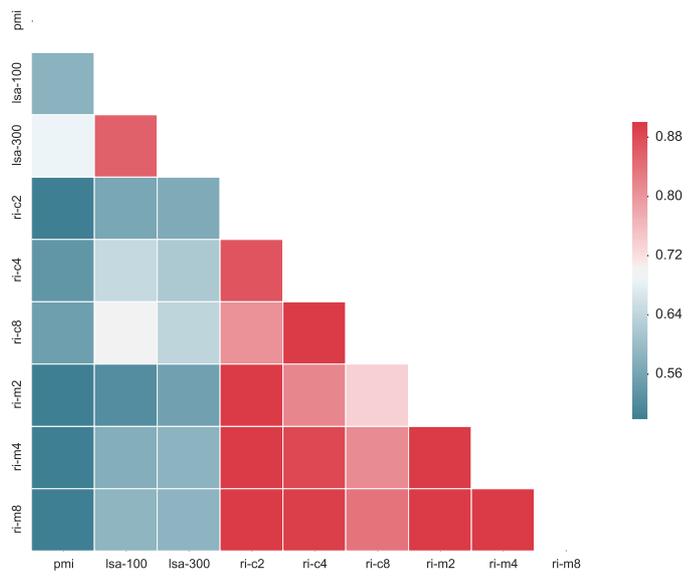
### 9.2.3 Comparison between the semantic kernels

In table 9.12 we show the pairwise cosine similarity between the symmetric similarity matrices underlying the semantic kernels.

	pmi	lsa-100	lsa-300	ri-c2	ri-c4	ri-c8	ri-m2	ri-m4	ri-m8
pmi	1.000	0.583	0.689	0.487	0.538	0.552	0.469	0.496	0.500
lsa-100	0.583	1.000	0.857	0.565	0.646	0.705	0.525	0.578	0.588
lsa-300	0.689	0.857	1.000	0.571	0.620	0.641	0.556	0.586	0.586
ri-c2	0.487	0.565	0.571	1.000	0.871	0.803	0.914	0.923	0.924
ri-c4	0.538	0.646	0.620	0.871	1.000	0.926	0.816	0.884	0.894
ri-c8	0.552	0.705	0.641	0.803	0.926	1.000	0.737	0.812	0.836
ri-m2	0.469	0.525	0.556	0.914	0.816	0.737	1.000	0.922	0.916
ri-m4	0.496	0.578	0.586	0.923	0.884	0.812	0.922	1.000	0.940
ri-m8	0.500	0.588	0.586	0.924	0.894	0.836	0.916	0.940	1.000

**Table 9.12.** The cosine similarity of the semantic kernels in the class *C08*.

As with the comparison of the semantic kernels in the Reuters-21578 collection we observe in figure 9.9 that there are two distinct groups of kernels: the LSA kernels, and the RI kernels respectively. We also notice a moderate similarity between the PMI kernel and the LSA-300 kernel.



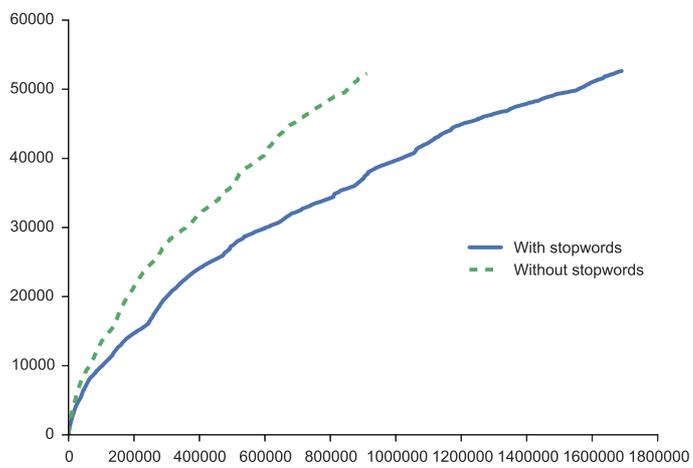
**Figure 9.9.** A heat map visualizing the cosine similarity between the semantic kernels.

### 9.3 Results for the 20 Newsgroups collection

The *20 Newsgroups* reference collection is presented in section 8.2.3. Below we present the statistical properties of the training and the tests respectively of the 20 Newsgroups collection.

	Training set	Test set
Number of documents:	5535	3686
Vocabulary size (with stopwords):	52643	42247
Vocabulary size (without stopwords):	52318	41922
Token-to-type ratio (without stopwords):	1.6173	1.6062
Token-to-type ratio (with stopwords):	1.3610	1.3585
Heaps parameters (with stopwords):	$k = 13.8973,$ $\beta = 0.5757$	$k = 9.2597,$ $\beta = 0.6080$

The statistical relationship between the number of tokens and the number of types in the 20 Newsgroups collection is visualized in figure 9.10.



**Figure 9.10.** The number of types versus the number of tokens in the 20 Newsgroups collection.

### 9.3.1 Using the *tf-idf* weighting scheme

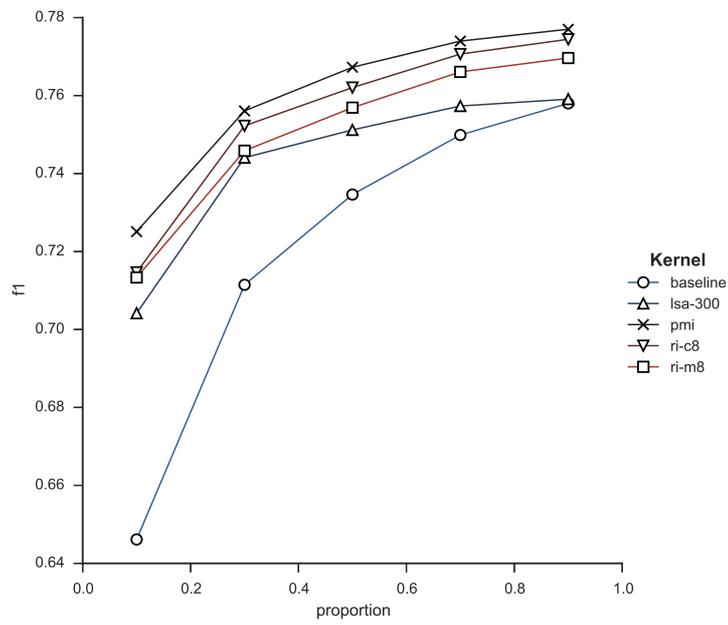
The trend that we have observed for the Reuters-21578 and the Ohsumed collections is also discernible in the 20 Newsgroups collection (see table 9.13 and figure 9.11). We notice that the  $F_1$  score and the AUC score is consistently higher for the semantic kernels, as compared to the baseline kernel. What is remarkable about the results for the 20 Newsgroups collection is that also the recall values are higher for several of the semantic kernels (in particular the RI kernels). Another detail in the result that should be pointed out is that the LSA-100 kernel performs somewhat worse than the baseline kernel when 90% of the training collection is used. The diagram in figure 9.11 demonstrates that the semantic kernels consistently perform better than the baseline linear kernel with regard to the  $F_1$  score. We notice in particular that the kernels based on random indexing and pointwise mutual information yield the best performance.

In table 9.14 we show the performance scores for the class *sci.space*, together with the estimated error margins for the  $F_1$  and the AUC scores. Also for this class we observe that the error margin is in general less than, and in a few cases slightly above, 0.01 on a confidence level of 95%.

Just as we observed in the Ohsumed collection there is a marked decrease of the recall values when 300 features are used, as compared to 100 features, in combination with using a small proportion of the training set (see table 9.15). When 90% of the training documents are used the  $F_1$  scores are comparable to, or even surpass, the corresponding  $F_1$  scores for 100 features. The  $F_1$  and the AUC scores are consistently higher for the semantic kernels.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.5787	0.7721	0.6461	0.8732
	lsa-100	0.6623	0.8023	0.7139	0.9096
	lsa-300	0.6497	0.7965	0.7042	0.9048
	pmi	0.6728	0.8124	0.7251	0.9161
	ri-c2	0.6624	0.7949	0.7107	0.9103
	ri-c4	0.6648	0.7894	0.7092	0.9061
	ri-c8	0.6693	0.7924	0.7147	0.9093
	ri-m2	0.6616	0.7970	0.7108	0.9094
	ri-m4	0.6640	0.7903	0.7103	0.9062
ri-m8	0.6613	0.8069	0.7133	0.9142	
0.5	baseline	0.6744	0.8400	0.7346	0.9214
	lsa-100	0.6926	0.8303	0.7455	0.9282
	lsa-300	0.6956	0.8389	0.7512	0.9305
	pmi	0.7115	0.8527	0.7673	0.9373
	ri-c2	0.7110	0.8391	0.7613	0.9315
	ri-c4	0.7072	0.8388	0.7583	0.9306
	ri-c8	0.7108	0.8412	0.7621	0.9321
	ri-m2	0.7121	0.8402	0.7625	0.9307
	ri-m4	0.7138	0.8405	0.7637	0.9305
ri-m8	0.7023	0.8451	0.7569	0.9334	
0.9	baseline	0.7032	0.8511	0.7580	0.9305
	lsa-100	0.6996	0.8358	0.7527	0.9320
	lsa-300	0.7055	0.8427	0.7591	0.9340
	pmi	0.7220	0.8594	0.7770	0.9406
	ri-c2	0.7233	0.8468	0.7729	0.9354
	ri-c4	0.7195	0.8485	0.7705	0.9354
	ri-c8	0.7239	0.8499	0.7744	0.9366
	ri-m2	0.7258	0.8493	0.7753	0.9345
	ri-m4	0.7276	0.8503	0.7767	0.9348
ri-m8	0.7175	0.8509	0.7696	0.9372	

**Table 9.13.** Macro-average scores for the *tf-idf* weighting scheme. Dimensionality = 100.



**Figure 9.11.** The average  $F_1$  score for a selection of semantic kernels obtained by sampling different proportions of the 20 Newsgroups reference collection. Dimensionality = 100.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.6655	0.7325	0.6949 ± 0.0151	0.8513 ± 0.0117
	lsa-100	0.7858	0.8276	0.8046 ± 0.0074	0.9269 ± 0.0056
	lsa-300	0.7659	0.8025	0.7814 ± 0.0080	0.9122 ± 0.0047
	pmi	0.7664	0.8384	0.7989 ± 0.0089	0.9225 ± 0.0052
	ri-c2	0.7596	0.7837	0.7697 ± 0.0109	0.8977 ± 0.0062
	ri-c4	0.7729	0.7901	0.7801 ± 0.0104	0.9044 ± 0.0061
	ri-c8	0.7738	0.8146	0.7924 ± 0.0092	0.9164 ± 0.0053
	ri-m2	0.7655	0.7935	0.7778 ± 0.0101	0.8999 ± 0.0055
	ri-m4	0.7624	0.7877	0.7738 ± 0.0084	0.8977 ± 0.0048
ri-m8	0.7644	0.7839	0.7728 ± 0.0105	0.8941 ± 0.0061	
0.5	baseline	0.8163	0.7940	0.8048 ± 0.0060	0.9212 ± 0.0038
	lsa-100	0.8269	0.8367	0.8316 ± 0.0041	0.9415 ± 0.0017
	lsa-300	0.8254	0.8204	0.8227 ± 0.0039	0.9341 ± 0.0022
	pmi	0.8187	0.8551	0.8362 ± 0.0028	0.9412 ± 0.0019
	ri-c2	0.8170	0.8283	0.8224 ± 0.0039	0.9244 ± 0.0027
	ri-c4	0.8139	0.8322	0.8227 ± 0.0048	0.9274 ± 0.0021
	ri-c8	0.8168	0.8326	0.8244 ± 0.0038	0.9318 ± 0.0019
	ri-m2	0.8127	0.8348	0.8234 ± 0.0039	0.9254 ± 0.0027
	ri-m4	0.8098	0.8250	0.8171 ± 0.0046	0.9216 ± 0.0026
ri-m8	0.8136	0.8277	0.8203 ± 0.0048	0.9230 ± 0.0024	
0.9	baseline	0.8256	0.8128	0.8190 ± 0.0041	0.9294 ± 0.0020
	lsa-100	0.8325	0.8357	0.8340 ± 0.0036	0.9445 ± 0.0011
	lsa-300	0.8352	0.8173	0.8260 ± 0.0021	0.9375 ± 0.0015
	pmi	0.8268	0.8601	0.8430 ± 0.0039	0.9441 ± 0.0018
	ri-c2	0.8288	0.8448	0.8366 ± 0.0028	0.9307 ± 0.0015
	ri-c4	0.8241	0.8431	0.8334 ± 0.0028	0.9333 ± 0.0017
	ri-c8	0.8348	0.8363	0.8354 ± 0.0026	0.9342 ± 0.0021
	ri-m2	0.8235	0.8447	0.8338 ± 0.0021	0.9326 ± 0.0017
	ri-m4	0.8234	0.8403	0.8317 ± 0.0030	0.9298 ± 0.0018
ri-m8	0.8273	0.8413	0.8341 ± 0.0023	0.9296 ± 0.0018	

**Table 9.14.** Average scores with error margins on the confidence level 95% for the class *sci.space*. Dimensionality = 100.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.4303	0.8470	0.5599	0.9100
	lsa-100	0.5668	0.8402	0.6694	0.9384
	lsa-300	0.5933	0.8290	0.6854	0.9392
	pmi	0.6203	0.8372	0.7052	0.9445
	ri-c2	0.5800	0.8196	0.6705	0.9362
	ri-c4	0.5756	0.8070	0.6636	0.9300
	ri-c8	0.6021	0.8120	0.6836	0.9354
	ri-m2	0.5735	0.8255	0.6687	0.9356
	ri-m4	0.5786	0.8151	0.6679	0.9333
ri-m8	0.5748	0.8156	0.6660	0.9326	
0.5	baseline	0.5871	0.8795	0.6946	0.9479
	lsa-100	0.6359	0.8802	0.7334	0.9566
	lsa-300	0.6534	0.8781	0.7448	0.9584
	pmi	0.6793	0.8866	0.7649	0.9638
	ri-c2	0.6642	0.8668	0.7459	0.9569
	ri-c4	0.6600	0.8662	0.7442	0.9557
	ri-c8	0.6736	0.8721	0.7550	0.9588
	ri-m2	0.6587	0.8694	0.7444	0.9555
	ri-m4	0.6675	0.8690	0.7495	0.9568
ri-m8	0.6660	0.8675	0.7478	0.9555	
0.9	baseline	0.6366	0.8884	0.7335	0.9555
	lsa-100	0.6548	0.8840	0.7477	0.9590
	lsa-300	0.6772	0.8859	0.7633	0.9618
	pmi	0.7021	0.8962	0.7835	0.9663
	ri-c2	0.6895	0.8782	0.7668	0.9606
	ri-c4	0.6907	0.8792	0.7688	0.9605
	ri-c8	0.6968	0.8827	0.7742	0.9627
	ri-m2	0.6875	0.8805	0.7674	0.9592
	ri-m4	0.6967	0.8813	0.7729	0.9614
ri-m8	0.6937	0.8801	0.7709	0.9599	

**Table 9.15.** Macro-average scores for the *tf-idf* weighting scheme. Dimensionality = 300.

### 9.3.2 Using the *dfr* weighting scheme

Also for the 20 Newsgroups collection we notice that the *divergence from randomness* weighting scheme outperforms the *tf-idf* scheme, and that the semantic kernels generally manage to increase the classification performance (see table 9.16). The difference between the baseline kernel and the semantic kernels is particularly large for small sizes of the training set, as we can see in figure 9.12. We observe the same trend for the *dfr* weighting scheme as for the *tf-idf* scheme, namely that the precision decreases when 300 features are used for training and classification, as compared to 100 features (see table 9.17). This trend is particularly discernible when small proportions of the training collection are used. We can also in the 20 Newsgroups collection observe that the *dfr* weighting scheme outperforms the *tf-idf* weighting scheme with regard to  $F_1$  score, as illustrated in figure 9.13. We also notice that the pointwise mutual information and the random indexing kernels perform slightly better than the kernels based on latent semantic analysis.

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.5977	0.7838	0.6659	0.8811
	lsa-100	0.6745	0.8145	0.7270	0.9200
	lsa-300	0.6606	0.8056	0.7162	0.9099
	pmi	0.6836	0.8235	0.7372	0.9200
	ri-c2	0.6764	0.8074	0.7249	0.9149
	ri-c4	0.6721	0.8037	0.7202	0.9154
	ri-c8	0.6793	0.8048	0.7261	0.9170
	ri-m2	0.6756	0.8087	0.7233	0.9165
	ri-m4	0.6773	0.8089	0.7250	0.9173
ri-m8	0.6770	0.8064	0.7251	0.9156	
0.5	baseline	0.7012	0.8481	0.7579	0.9262
	lsa-100	0.7047	0.8403	0.7576	0.9350
	lsa-300	0.7054	0.8497	0.7622	0.9343
	pmi	0.7238	0.8621	0.7791	0.9413
	ri-c2	0.7181	0.8483	0.7697	0.9341
	ri-c4	0.7143	0.8527	0.7686	0.9379
	ri-c8	0.7177	0.8494	0.7695	0.9371
	ri-m2	0.7173	0.8498	0.7689	0.9357
	ri-m4	0.7180	0.8540	0.7712	0.9372
ri-m8	0.7202	0.8523	0.7727	0.9370	
0.9	baseline	0.7265	0.8565	0.7775	0.9341
	lsa-100	0.7126	0.8445	0.7642	0.9383
	lsa-300	0.7144	0.8559	0.7707	0.9381
	pmi	0.7337	0.8687	0.7885	0.9439
	ri-c2	0.7322	0.8545	0.7812	0.9372
	ri-c4	0.7274	0.8614	0.7809	0.9414
	ri-c8	0.7292	0.8599	0.7813	0.9409
	ri-m2	0.7279	0.8575	0.7794	0.9392
	ri-m4	0.7308	0.8614	0.7826	0.9401
ri-m8	0.7348	0.8599	0.7852	0.9409	

**Table 9.16.** Macro-average scores for the *dfr* weighting scheme.  
Dimensionality = 100.

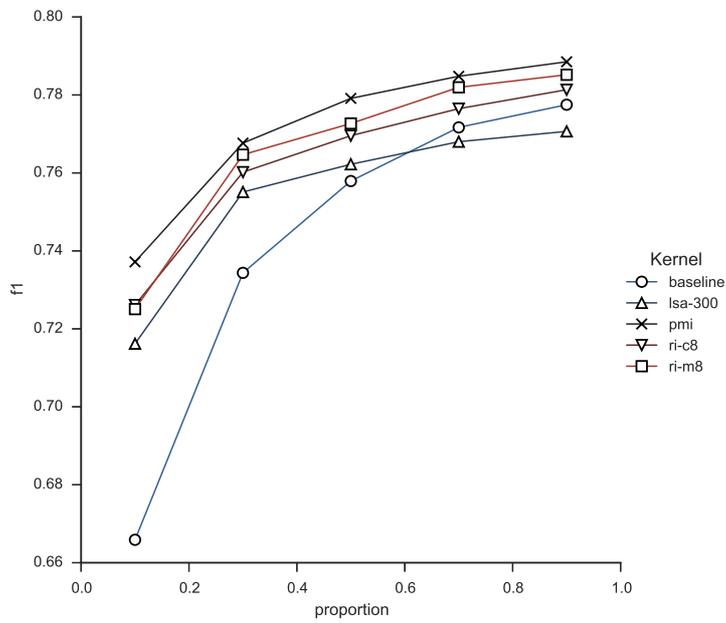
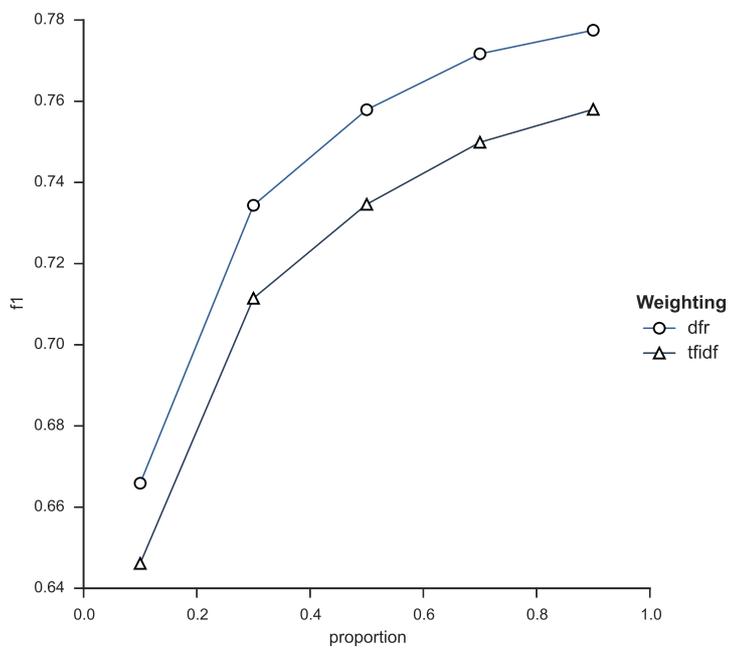


Figure 9.12. Macro-average scores. Dimensionality = 100.

## 9.3.2.1 Number of features = 300

proportion	kernel	precision	recall	f1	auc
0.1	baseline	0.4699	0.8476	0.5951	0.9182
	lsa-100	0.5844	0.8523	0.6860	0.9434
	lsa-300	0.5738	0.8367	0.6733	0.9363
	pmi	0.6215	0.8449	0.7094	0.9450
	ri-c2	0.5916	0.8324	0.6836	0.9408
	ri-c4	0.5947	0.8217	0.6819	0.9374
	ri-c8	0.6066	0.8150	0.6885	0.9365
	ri-m2	0.5913	0.8405	0.6855	0.9404
	ri-m4	0.5980	0.8248	0.6853	0.9391
ri-m8	0.5942	0.8292	0.6846	0.9400	
0.5	baseline	0.6138	0.8789	0.7153	0.9496
	lsa-100	0.6444	0.8826	0.7398	0.9585
	lsa-300	0.6506	0.8812	0.7436	0.9582
	pmi	0.6819	0.8881	0.7679	0.9634
	ri-c2	0.6699	0.8704	0.7510	0.9575
	ri-c4	0.6684	0.8708	0.7503	0.9583
	ri-c8	0.6752	0.8731	0.7572	0.9591
	ri-m2	0.6617	0.8724	0.7462	0.9555
	ri-m4	0.6800	0.8725	0.7591	0.9591
ri-m8	0.6725	0.8717	0.7535	0.9589	
0.9	baseline	0.6570	0.8906	0.7497	0.9564
	lsa-100	0.6627	0.8851	0.7529	0.9606
	lsa-300	0.6782	0.8870	0.7640	0.9618
	pmi	0.7042	0.8956	0.7854	0.9658
	ri-c2	0.6957	0.8773	0.7702	0.9604
	ri-c4	0.6979	0.8822	0.7736	0.9624
	ri-c8	0.7014	0.8836	0.7779	0.9631
	ri-m2	0.6888	0.8804	0.7671	0.9580
	ri-m4	0.7056	0.8845	0.7803	0.9629
ri-m8	0.6975	0.8805	0.7730	0.9620	

**Table 9.17.** Macro-average scores for the *dfr* weighting scheme. Dimensionality = 300.



**Figure 9.13.** The average F1 score in the 20 Newsgroups collection for the *tf-idf* and the *dfr* weighting schemes. Dimensionality = 100.

### 9.3.3 Comparison between the semantic kernels

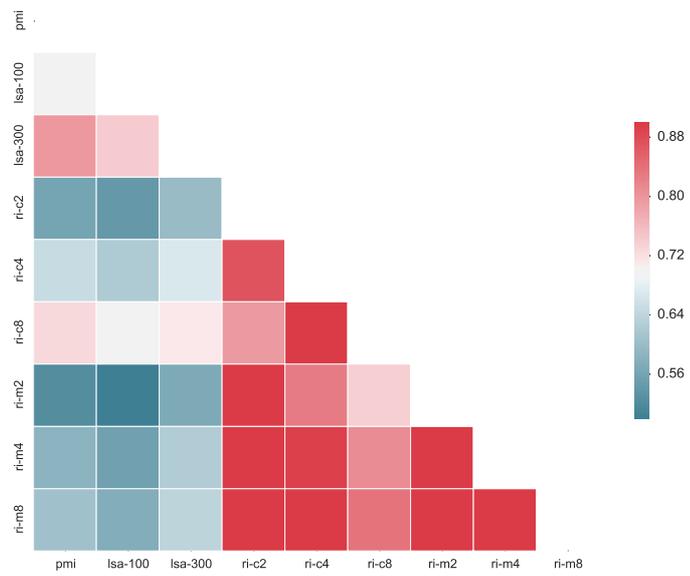
In table 9.18 we show the pairwise cosine similarity between the symmetric similarity matrices underlying the semantic kernels.

	pmi	lsa-100	lsa-300	ri-c2	ri-c4	ri-c8	ri-m2	ri-m4	ri-m8
pmi	1.000	0.695	0.797	0.559	0.649	0.728	0.523	0.585	0.607
lsa-100	0.695	1.000	0.741	0.542	0.622	0.704	0.499	0.556	0.576
lsa-300	0.797	0.741	1.000	0.598	0.670	0.711	0.570	0.625	0.638
ri-c2	0.559	0.542	0.598	1.000	0.872	0.795	0.918	0.920	0.921
ri-c4	0.649	0.622	0.670	0.872	1.000	0.915	0.829	0.893	0.905
ri-c8	0.728	0.704	0.711	0.795	0.915	1.000	0.738	0.811	0.836
ri-m2	0.523	0.499	0.570	0.918	0.829	0.738	1.000	0.928	0.922
ri-m4	0.585	0.556	0.625	0.920	0.893	0.811	0.928	1.000	0.946
ri-m8	0.607	0.576	0.638	0.921	0.905	0.836	0.922	0.946	1.000

**Table 9.18.** The cosine similarity of the semantic kernels in the class *sci.space*.

The heat map in figure 9.14 yet again shows that the LSA kernels and the RI kernels form two distinct groups with respect to their underlying semantic similarity matrices. We also notice that the PMI kernel is more similar to the LSA kernels, than to the RI kernels.

We have thereby concluded the presentation of the results from the empirical investigation of the selected semantic kernels and term weighting schemes and will now summarize this study, as well as the conclusions that can be drawn, in chapter 10.



**Figure 9.14.** A heat map visualizing the cosine similarity between the semantic kernels.

## Chapter 10

# Conclusions

This thesis is structured around two major research objectives, one theoretical and one empirical research objective respectively. In chapters 2 – 4 we investigate document classification from a theoretical perspective. The purpose of the theoretical investigation is to study how *text categorization* can be defined and characterized using different theories within the fields of formal linguistics and mathematics. We also seek to theoretically investigate the formal structures of hierarchical classification schedules as well as document collections that have been subjected to a particular classification.

The purpose of the empirical investigation is to study the classification performance of the support vector machine (SVM) algorithm of by comparing a category of kernels called *semantic* kernels. These semantic kernels are in turn induced by different statistical methods for computing semantic relatedness between terms in a document collection. The underlying theoretical framework of these methods for computing semantic relatedness is based on the *distributional hypothesis* of semantics, which captures the notion that semantically related terms tend to appear in similar contexts. We also investigate and compare

two weighting schemes for document representation, the *tf-idf* scheme and the *divergence from randomness* scheme respectively. The divergence from randomness weighting has so far received little (if any) attention within the research field of automatic text categorization. The methodology of the empirical investigation is outlined in 8 and the results of the experiments presented in chapter 9.

With respect to our theoretical investigation of the properties of classification we can succinctly summarize our findings as follows. Classification schedules can be defined and analyzed as formal languages and each classification decision corresponds to a sentence in a classificatory language. Each classificatory language has an internal structure that can be comprehensively described using notions of *topology*. Interestingly, also the classification of a particular document collection yields a topological space on that collection. This means that the structure imposed on a document collection by means of classification is inherent in the classificatory language used for making the classification decisions explicit. We have also found, using concepts from a relatively young mathematical branch called *category theory*, that the notion of *space* induced by a *basis* (an underlying set of basic “constituents” of the space) is present on several levels of our analysis – classification schedules and document collections are spaces, as well as the vector spaces used to represent and automatically classify document collections. Category theory also offers a novel way to formalize classification without making initial assumptions about the kind of categories (such as sets) that should be used.

In the empirical part of this thesis we found that semantic kernels generally improve the classification performance of the SVM algorithm, as measured by the  $F_1$  and the AUC performance measures. This is particularly noticeable when small proportions of the training documents have been used to induce a classifier. As was observed by

Basili et al. (2006) it appears that the presence of semantic information in the kernel can to a certain extent compensate for the lack of training data for the classifier. In this study we have, however, not utilized an external data source like WordNet for producing the semantic kernels, but instead the set of training documents used to induce classifiers.

We have investigated different statistical methods for generating the semantic information contained in these kernels. All these methods use the notion of *co-occurrence* (and from a statistical perspective: *co-occurrence frequencies*) as the basis for computing semantic relatedness between terms. Two of the methods used, the *pointwise mutual information* measure and the *latent semantic analysis* method collect the co-occurrence frequencies from the larger units like entire documents. The third method used, the *random indexing method*, collects the statistical information from the immediate context of the respective terms. This method can also be adjusted so that all the terms in the analysis window have an equal influence on the contextual information for a specific target term, or that a higher weight is given to terms that are within a shorter textual distance from the target term.

None of the semantic kernels perform consistently better than the other kernels. We have noticed that the PMI kernel is the best performer in the Reuters-21578 and the 20 Newsgroups collections, whereas the kernels based on random indexing perform best in the Ohsumed collection. As our statistical analysis of the Ohsumed collection shows, it has a higher token-to-type ratio than the other collections, as well as a lower  $\beta$  increase rate. This could indicate that the random indexing (RI) kernels that utilize contextual information rather than term-term co-occurrence frequencies are favored in text collections having a smaller number of distinct words. With regard to the RI kernels there is no discernible performance difference between *constant* and *distance-dependent* weighting (see section 8.5.3) of the

context vectors. There is also no consistent advantage for a certain size of the analysis window.

One result that stands out from the empirical part of this work is that the *divergence from randomness* weighting scheme appears to outperform the *tf-idf* weighting scheme. The former weighting scheme is based on the probabilistic principle that terms should be weighted according to the extent to which their local frequencies diverge from the expected frequencies that would be observed if the terms were randomly distributed among the documents in the collection. This is, in a sense, a more sophisticated weighting scheme than the *tf-idf* scheme which is “merely” based on the balance between the local (within-document) and the global (within-collection) frequencies of terms. The results from this study clearly show that the divergence from randomness scheme deserves more attention in text categorization research, as a potential standard approach to term weighting.

# Bibliography

- Akhiezer, N. I., & Glazman, I. M. (1993). *Theory of linear operators in Hilbert space*. New York: Dover.
- Amari, S., & Wu, S. (1999). Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, *12*, 783–789.
- Amati, G., & Van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, *20*(4), 357–389.
- Aratljo, M., Navarro, G., & Ziviani, N. (1997). Large text searching allowing errors. In R. Baeza-Yates (Ed.), *Proceedings of the Fourth South American Workshop on String Processing*, publisher = Carleton University Press International Informatics Series, v. 8 (pp. 2–20).
- Arenas, F. G. (1999). Alexandroff spaces. *Acta Mathematica Universitatis Comenianae*, *68*(1), 17–25.
- Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, *68*(3), 337–404.

- Baeza-Yates, R., & Ribeiro-Neto, B. d. A. (2011). *Modern information retrieval : the concepts and technology behind search* (2nd ed.). Harlow: Addison-Wesley.
- Basili, R., Cammisa, M., & Moschitti, A. (2006). A semantic kernel to classify texts with very few training examples. *Informatica*, 30(2), 163–172.
- Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. *Journal of Machine Learning Research*, 2, 125–137.
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281–305.
- Berry, M. W., Dumais, S. T., & O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37(4), 573–595.
- Bertsekas, D. P. (1999). *Nonlinear programming* (2nd ed.). Belmont, Mass.: Athena Scientific.
- Blackburn, P., & Bos, J. (2005). *Representation and inference for natural language : a first course in computational semantics*. Stanford, Calif.: Center for the Study of Language and Information.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022.
- Bloehdorn, S., Basili, R., Cammisa, M., & Moschitti, A. (2006). Semantic kernels for text classification based on topological

- measures of feature similarity. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006)*.
- Bondy, J. A., & Murty, U. S. R. (2008). *Graph theory*. New York: Springer.
- Boothby, W. M. (2003). *An introduction to differentiable manifolds and Riemannian geometry* (2nd ed.). Amsterdam: Academic Press.
- Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144–152). New York, NY, USA: ACM.
- Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*.
- Bourbaki, N. (2004). *Theory of sets*. Berlin: Springer.
- Buchanan, B. (1979). *Theory of library classification*. London: Bingley.
- Budanitsky, A., & Hirst, G. (2006). Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1), 13–47.
- Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, 121–167.
- Cai, L., & Hofmann, T. (2004). Hierarchical document categorization with support vector machines. In *CIKM '04: Proceedings of the*

- thirteenth ACM international conference on Information and knowledge management* (pp. 78–87). New York, NY, USA: ACM.
- Cardoso-Cachopo, A., Oliveira, A. L., & Redol, R. A. (2003). An empirical comparison of text categorization methods. In *String Processing and Information Retrieval, 10th International Symposium* (pp. 183–196). Springer Verlag.
- Carnap, R. (1955). Meaning and synonymy in natural languages. *Philosophical Studies*, 6(3), 33–47.
- Chan, L. M. (1994). *Cataloging and classification : an introduction* (2nd ed.). New York: McGraw-Hill.
- Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2, 27:1–27:27.
- Chen, J., & Saad, Y. (2009). Lanczos vectors versus singular vectors for effective dimension reduction. *IEEE Transactions on Knowledge and Data Engineering*, 21(8), 1091–1103.
- Chu, C. M., & O'Brien, A. (1993). Subject analysis: the critical first stage in indexing. *Journal of Information Science*, 19(6), 439–454.
- Church, A. (1996). *Introduction to mathematical logic*. Princeton, N.J.: Princeton University Press.
- Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1), 22–29.

- Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3), 370–383.
- Clarke, D. (2012). A context-theoretic framework for compositionality in distributional semantics. *Computational Linguistics*, 38(1), 41–71.
- Cohen, T., Schvaneveldt, R., & Widdows, D. (2010). Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43(2), 240 - 256.
- Cohn, D. L. (2013). *Measure theory* (2nd ed.). New York: Springer.
- Cortes, C., & Vapnik, V. (1995). Support vector networks. In *Machine Learning* (pp. 273–297).
- Crammer, K., Singer, Y., Cristianini, N., Shawe-taylor, J., & Williamson, B. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2, 265–292.
- Crespi Reghizzi, S., Breveglieri, L., & Morzenti, A. (2013). *Formal languages and compilation* (2nd ed.).
- Cristianini, N., Shawe-Taylor, J., & Lodhi, H. (2002). Latent semantic kernels. *Journal of Intelligent Information Systems*, 18(2-3), 127–152.
- Cruse, D. A. (2004). *Meaning in language : an introduction to semantics and pragmatics* (2nd ed.). Oxford: Oxford University Press.

- De, U. C., Shaikh, A. A., & Sengupta, J. (2008). *Tensor calculus* (2nd ed.). Oxford: Alpha Science International Ltd.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 391–407.
- Deskins, W. E. (1995). *Abstract algebra*. New York: Dover Publications.
- Dewey, M., Mitchell, J. S., & Beall, J. (2003). *Dewey decimal classification and relative index* (22nd ed.). Dublin, Ohio: OCLC Online Computer Library Center.
- Dominich, S. (2000). A unified mathematical definition of classical information retrieval. *Journal of the American Society for Information Science*, 51(7), 614–624.
- Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A., & Vapnik, V. (1997). Support vector regression machines. In *Advances in Neural Information Processing Systems 9* (pp. 155–161). MIT Press.
- Duan, K., Keerthi, S. S., & Poo, A. N. (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51, 41–59.
- Edgar, G. A. (2008). *Measure, topology, and fractal geometry* (2nd ed.). New York: Springer-Verlag.
- Egghe, L., & Rousseau, R. (1998). Topological aspects of information retrieval. *Journal of the American Society for Information Science*, 49(13), 1144–1160.

- Estabrooks, A., Jo, T., & Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 18–36.
- Everett, D., & Cater, S. (1992). Topology of document retrieval systems. *Journal of the American Society for Information Science*, 43(10), 658–673.
- Falconer, K. J. (2003). *Fractal geometry : mathematical foundations and applications* (2nd ed.). Chichester: Wiley.
- Farahat, A. K., & Kamel, M. S. (2011). Statistical semantics for enhancing document clustering. *Knowledge and Information Systems*, 28(2), 365–393.
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874.
- Fletcher, R. (1987). *Practical methods of optimization* (2nd ed.). Chichester: Wiley.
- Foskett, A. C. (1996). *The subject approach to information* (5th ed.). London: Library Association Publ.
- Frank, E., & Paynter, G. W. (2004). Predicting Library of Congress classifications from Library of Congress subject headings. *Journal of the American Society for Information Science and Technology*, 55(3), 214–227.
- Fürnkranz, J. (2002). Round robin classification. *The Journal of Machine Learning Research*, 2, 721–747.
- Gardin, J.-C. (1973). Document analysis and linguistic theory. *Journal of Documentation*, 29(2), 137–168.

- Golub, G. H., Luk, F. T., & Overton, M. L. (1981). A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix. *ACM Transactions on Mathematical Software*, 7(2), 149–169.
- Gong, Y., & Liu, X. (2001). Creating generic text summaries. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on* (p. 903-907).
- Grönqvist, L. (2006). *Exploring latent semantic vector models enriched with n-grams*. Växjö: Växjö University Press. (Diss. Växjö : Växjö universitet, 2006)
- Hassel, M. (2013). *JavaSDM - A Java tool-kit for working with Random Indexing*. Open Source, code released under GPL. Retrieved from <http://www.nada.kth.se/~xmartin/java/> (Accessed: 2016-03-11)
- Heaps, H. S. (1978). *Information retrieval : computational and theoretical aspects*. New York: Academic Press.
- Hersh, W., Buckley, C., Leone, T. J., & Hickam, D. (1994). OHSUMED: an interactive retrieval evaluation and new large test collection for research. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 192–201). New York, NY, USA: Springer-Verlag New York, Inc.
- Hinman, P. G. (2005). *Fundamentals of mathematical logic*. Wellesley, Mass.: A.K. Peters.
- Hjørland, B. (1992). The concept of 'subject' in information science. *Journal of Documentation*, 48(2), 172–200.

- Hjørland, B. (2001). Towards a theory of aboutness, subject, topicality, the domain, field, content ... and relevance. *Journal of the American Society for Information Science*, 52, 775–778.
- Hjørland, B. (2005). Empiricism, rationalism and positivism in library and information science. *Journal of Documentation*, 61(1), 130–155.
- Hjørland, B., & Pedersen, K. N. (2005). A substantive theory of classification for information retrieval. *Journal of Documentation*, 61(5), 582–597.
- Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 50–57). New York, NY, USA: ACM.
- Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *Annals of Statistics*, 36(3), 1171–1220.
- Howell, K. E. (2013). *An introduction to the philosophy of methodology*. Los Angeles: SAGE.
- Hrbacek, K., & Jech, T. (1999). *Introduction to set theory* (3rd ed.). New York: Marcel Dekker.
- Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2010). *A practical guide to support vector classification* (Tech. Rep.). Department of Computer Science, National Taiwan University.
- Hutchins, W. J. (1975). *Languages of indexing and classification : a linguistic study of structures and functions*. Stevenage: Peregrinus.

Jacob, E. K. (2004). Classification and categorization: A difference that makes a difference. *Library Trends*, 52, 2004.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning* (pp. 137–142). London, UK: Springer-Verlag.

Joachims, T. (2002). *Learning to classify text using support vector machines*. Boston: Kluwer Academic Publishers.

Jones, E., Oliphant, T., Peterson, P., et al. (2001–). *SciPy: Open source scientific tools for Python*. Retrieved from <http://www.scipy.org/> (Accessed: 2015-06-08)

Kanerva, P., Kristoferson, J., & Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society* (pp. 103–106). Erlbaum.

Karlgren, J., & Sahlgren, M. (2001). From words to understanding. In *Foundations of real-world intelligence*. Retrieved from <http://eprints.sics.se/131/01/KarlgrenSahlgren2001.pdf> (Accessed: 2016-03-11)

Katona, G., & Nemetz, O. (1976). Huffman codes and self-information. *IEEE Transactions on Information Theory*, 22(3), 337-340.

Kimeldorf, G., & Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1), 82–95.

- Kjørup, S. (1999). *Människovetenskaperna : problem och traditioner i humanioras vetenskapsteori*. Lund: Studentlitteratur.
- Koike, H. (1995). Fractal views: a fractal-based method for controlling information display. *ACM Transactions on Information Systems (TOIS)*, 13(3), 305–323.
- Kolmogorov, A. N., Silverman, R. A., & Fomin, S. V. (1975). *Introductory real analysis* (Rev. ed.). New York: Dover.
- Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies* (pp. 3–24). Amsterdam, The Netherlands, The Netherlands: IOS Press.
- Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 179–186). Morgan Kaufmann.
- Lanczos, C. (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4), 255–282.
- Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2), 211–240.
- Langridge, D. W. (1989). *Subject analysis : principles and procedures*. London: Bowker-Saur.

- Larson, R. R. (1992). Experiments in automatic Library of Congress classification. *Journal of the American Society for Information Science*, 43(2), 130–148.
- Lee, J.-W., & Khargonekar, P. P. (2009). Distribution-free consistency of empirical risk minimization and support vector regression. *Mathematics of Control, Signals, and Systems (MCSS)*, 21(2), 111–125.
- Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5, 361–397.
- Ling, C., Ling, C. X., & Li, C. (1998). Data mining for direct marketing: Problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)* (pp. 73–79). AAAI Press.
- Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4), 309–317.
- Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2), 203–208.
- Mac Lane, S. (1998). *Categories for the working mathematician* (2nd ed.). New York: Springer.
- Mandelbrot, B., & Green, M. (1999). The canopy and shortest path in a self-contacting fractal tree. *The Mathematical Intelligencer*, 21(2), 18–27.

- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.
- Marcella, R., & Newton, R. (1994). *A new manual of classification*. Aldershot: Gower.
- Marker, D. (2002). *Model theory : an introduction*. New York: Springer.
- Marradi, A. (1990). Classification, typology, taxonomy. *Quality and Quantity*, 24(2), 129–157.
- Matsumoto, M., & Nishimura, T. (1998). Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1), 3–30.
- Mattila, P. (1995). *Geometry of sets and measures in Euclidean spaces : fractals and rectifiability*. Cambridge: Cambridge Univ. Press.
- McIlwaine, I. C. (2000). *The Universal Decimal Classification : a guide to its use*. The Hague: UDC Consortium.
- Mehler, A., Waltinger, U., & Wegner, A. (2007). A formal text representation model based on lexical chaining. In *Proceedings of the KI 2007 Workshop on Learning from Non-Vectorial Data (LNVD 2007) September 10, Osnabrück* (pp. 17–26).
- Mendelson, B. (1990). *Introduction to topology* (3rd ed.). New York: Dover.
- Meyer, D., Leisch, F., & Hornik, K. (2003). The support vector machine under test. *Neurocomputing*, 55(1-2), 169–186.

Mitchell, T. M. (1997). *Machine learning*. New York: McGraw-Hill.

Mokhtar, U. A., & Yusof, Z. M. (2015). Classification: The understudied concept. *International Journal of Information Management*, 35(2), 176–182.

Moschitti, A. (n.d.). *Text categorization corpora*. Retrieved from <http://disi.unitn.it/moschitti/corpora.htm> (Accessed: 2014-11-13)

Munkres, J. R. (2000). *Topology* (two ed.). Upper Saddle River, N.J.: Prentice Hall.

Nguyen, H. (1995). Some mathematical tools for decision making under partial knowledge. In *Mathematical models for handling partial knowledge in artificial intelligence*. New York: Plenum.

Nisa, R., & Qamar, U. (2014). A text mining based approach for web service classification. *Information Systems and e-Business Management*, 1–18.

Nocedal, J., & Wright, S. J. (2006). *Numerical optimization* (two ed.). Berlin: Springer.

Ong, T., Leggett, J. J., & Yun, U. (2005). Visualizing hierarchies and collection structures with fractal trees. In *ENC '05: Proceedings of the Sixth Mexican International Conference on Computer Science* (pp. 31–40). Washington, DC, USA: IEEE Computer Society.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

- Pierce, B. C. (1991). *Basic category theory for computer scientists*. Cambridge, Mass.: The MIT Press.
- Popper, K. (1992). *The logic of scientific discovery*. London: Routledge.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 3(14), 130–137.
- Ranganathan, S. R. (1989). *Philosophy of library classification* (1. Indian repr. ed.). Bangalore: Sarada Ranganathan Endowment for Library Science.
- Reitz, J. M. (2004). *Dictionary for library and information science*. Westport, Conn.: Libraries Unlimited.
- Rellick, L. M., Edgar, G. A., & Klapper, M. H. (1991). Calculating the Hausdorff dimension of tree structures. *Journal of Statistical Physics*, 64(1–2), 77–85.
- Rennie, J. (n.d.). *20 newsgroups*. Retrieved from <http://qwone.com/~jason/20Newsgroups/> (Accessed: 2014-11-13)
- Renteln, P., & Dundes, A. (2005). Foolproof: A sampling of mathematical folk humor. *Notices of the American Mathematical Society*, 52(1), 24–34.
- Restle, F. (1959). A metric and an ordering on sets. *Psychometrika*, 24(3), 207–220.
- Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5), 503–520.

- Rohde, D. (2007). *SVDLIBC: A C library for computing singular value decompositions*. Retrieved from <http://tedlab.mit.edu/~dr/SVDLIBC/> (Accessed: 2014-11-13)
- Roman, S. (2008). *Lattices and ordered sets*. New York: Springer.
- Rosch, E. (1975). Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104(3), 192–233.
- Rosenberg, A. (1995). *Philosophy of social science* (2nd ed.). Boulder: Westview.
- Rosenberg, A. (2005). *Philosophy of science : a contemporary introduction* (2nd ed.). New York: Routledge.
- Sahlgren, M. (2008). The distributional hypothesis. *Rivista di Linguistica (Italian Journal of Linguistics)*, 20(1), 33-53.
- Sahlgren, M., & Karlgren, J. (2005). Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Nat. Lang. Eng.*, 11(3), 327–341.
- Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523.
- Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613–620.
- Schäuble, P. (1987). Thesaurus based concept spaces. In *SIGIR '87: Proceedings of the 10th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 254–262). New York, NY, USA: ACM.

- Schölkopf, B., Burges, C. J. C., & Smola, A. J. (1999). *Advances in kernel methods : support vector learning*. Cambridge, Mass.: MIT Press.
- Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels : support vector machines, regularization, optimization, and beyond*. Cambridge, Mass.: MIT Press.
- Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, 12(5), 1207–1245.
- Schütze, H. (1992). Dimensions of meaning. In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing* (pp. 787–796). Los Alamitos, CA, USA: IEEE Computer Society Press.
- Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1), 97–123.
- Sebastiani, F. (2005). Text categorization. In *Text Mining and its Applications to Intelligence, CRM and Knowledge Management* (pp. 109–129). WIT Press.
- Shafer, K. E. (2001). Automatic subject assignment via the Scorpion system. *Journal of Library Administration*, 34(1), 187–189.
- Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge: Cambridge University Press.
- Siolas, G., & d'Alché Buc, F. (2000). Support vector machines based on a semantic kernel for text categorization. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)-Volume 5 - Volume 5* (pp. 205–209). Washington, DC, USA: IEEE Computer Society.

- Slavic, A. (2008). Faceted classification: Management and use. *Axiomathes*, 18(2), 257–271.
- Spärck Jones, K. (1970). Some thoughts on classification for retrieval. *Journal of Documentation*, 26, 89–101.
- Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1), 11–21.
- Stavrianou, A., Andritsos, P., & Nicoloyannis, N. (2007). Overview and semantic issues of text mining. *ACM SIGMOD Record*, 36(3), 23–34.
- Stock, W. G. (2010). Concepts and semantic relations in information science. *Journal of the American Society for Information Science and Technology*, 61, 1951–1969.
- Tarski, A. (1956). *Logic, semantics, metamathematics : papers from 1923 to 1983*. Oxford: Clarendon.
- Taylor, A. G., & Miller, D. P. (2006). *Introduction to cataloging and classification* (10th ed.). Westport, Conn.: Libraries Unlimited.
- Thompson, R., Shafer, K., & Vizine-Goetz, D. (1997). Evaluating Dewey concepts as a knowledge base for automatic subject assignment. In *DL '97: Proceedings of the second ACM international conference on Digital libraries* (pp. 37–46). New York, NY, USA: ACM.
- Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6, 1453–1484.

- Turney, P. D. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning* (pp. 491–502). London, UK, UK: Springer-Verlag.
- Van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). London: Butterworths. Retrieved from <http://www.dcs.gla.ac.uk/Keith/Preface.html> (Accessed: 2016-03-11)
- Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.
- Řehůřek, R. (2011). Subspace tracking for latent semantic analysis. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval* (pp. 289–300). Berlin, Heidelberg: Springer-Verlag.
- Wang, P., & Domeniconi, C. (2008). Building semantic kernels for text classification using wikipedia. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 713–721). New York, NY, USA: ACM.
- Widdows, D. (2008). Semantic vector products: Some initial investigations. In *Proceedings of the Second AAAI Symposium on Quantum Interaction*.
- Willard, S. (2004). *General topology*. Mineola, N.Y.: Dover Publications.
- Wille, R. (2005). Formal concept analysis as mathematical theory of concepts and concept hierarchies. In *Formal Concept Analysis* (p. 1-33).

- Wong, S. K. M., Ziarko, W., & Wong, P. C. N. (1985). Generalized vector spaces model in information retrieval. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 18–25). New York, NY, USA: ACM.
- Wu, H., & Gunopulos, D. (2002). Evaluating the utility of statistical phrases and latent semantic indexing for text classification. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on* (p. 713-716).
- Yi, K. (2006). Challenges in automated classification using library classification schemes. In *Proceedings of World Library and Information Congress: 72nd IFLA General Conference and Council*.
- Zelikovitz, S., & Hirsh, H. (2001). Using LSI for text classification in the presence of background text. In *Proceedings of the Tenth International Conference on Information and Knowledge Management* (pp. 113–118). New York, NY, USA: ACM.

## Publikationer i serien *Skrifter från VALFRID*

Enmark, Romulo (red.) (1990). *Biblioteksstudier. Folkbibliotek i flervetenskaplig belysning*. (ISBN 91-971457-0-X) Skriftserien; 1

Enmark, Romulo (red.) (1991). *Biblioteken och framtiden. [Bok 1], Framtidsdebatten i nordisk bibliotekspress*. (ISBN 91-971457-1-8) Skriftserien; 2

Selden, Lars (red.) (1992). *Biblioteken och framtiden. [Bok 2], Nordisk idédebatt: konferens i Borås 11-13 november 1991*. (ISBN 91-971457-2-6) Skriftserien; 3

Hjørland, Birger (1993). *Emnerepræsentation og informationssøgning: bidrag til en teori på kundskabsteoretisk grundlag*. 2. uppl. med register. (ISBN 91-971457-4-2) Skriftserien; 4

Höglund, Lars (red.) (1995). *Biblioteken, kulturen och den sociala intelligensen: aktuell forskning inom biblioteks- och informationsvetenskap*. (ISBN 91-971457-5-0) Skriftserien; 5

Hjørland, Birger (1995). *Faglitteratur: kvalitet, vurdering og selektion: grundbog i materialevalg*. (ISBN 91-971457-6-9) Skriftserien; 6

Limberg, Louise (1996). *Skolbiblioteksmodeller. Utvärdering av ett utvecklingsprojekt i Örebro län.* (ISBN 91-971457-7-7) Skriftserien; 7

Hjørland, Birger (1997). *Faglitteratur: kvalitet, vurdering og selektion. Bd1, Materialevalgets almene teori, metoder og forudsætninger.* (ISBN 91-971457-8-5) Skriftserien; 8

Pettersson, Rune (1997). *Verbo-visual Communication: Presentation of Clear*

*Messages for Information and Learning.* (ISBN 91-971457-9-3) Skriftserien ; 9

Pettersson, Rune (1997). *Verbo-visual Communication: 12 Selected Papers.* (ISBN 91-973090-0-1) Skriftserien; 10

Limberg, Louise. (1997). *Skolbiblioteksmodeller: utvärdering av ett utvecklingsprojekt i Örebro län.* (ISBN 91-973090-1-X) Skriftserien; 11 (nytryck av nr 7, bilaga inne i boken)

Eliasson, Anette, Löf Staffan, Rydsjö, Kerstin (red.) (1997). *Barnbibliotek och informationsteknik: elektroniska medier för barn och ungdomar på folkbibliotek.* (ISBN 91-973090-2-8) Skriftserien; 12

Klasson, Maj (red.) (1997). *Folkbildning och bibliotek? På spaning efter spår av folkbildning och livslångt lärande i biblioteksvärlden.* (ISBN 91-973090-3-6) Skriftserien; 13

Zetterlund, Angela (1997). *Utvärdering och folkbibliotek: en studie av utvärderingens teori och praktik med exempel från folkbibliotekens förändrings- och utvecklingsprojekt*. (ISBN 91-973090-4-4) Skriftserien; 14

Myrstener, Mats (1998). *På väg mot ett stadsbibliotek: folkbiblioteksväsendets framväxt i Stockholm t o m 1927*. (ISBN 91-973090-5-2) Skriftserien; 15

Limberg, Louise (2001). *Att söka information för att lära: en studie av samspel mellan informationssökning och lärande*. (ISBN 91-973090-6-0) Första uppl. 1998. Skriftserien; 16

Hansson, Joacim (1998). *Om folkbibliotekens ideologiska identitet: en diskursstudie*. (ISBN 91-973090-7-9) Skriftserien ; 17

Gram, Magdalena (2000). *Konstbiblioteket: en krönika och en fallstudie*. (ISBN 91-973090-8-7) Skriftserien; 18

Hansson, Joacim (1999). *Klassifikation, bibliotek och samhälle: en kritisk hermeneutisk studie av "Klassifikationssystem för svenska bibliotek"*. (ISBN 91-973090-9-5) Skriftserien; 19

Seldén, Lars (2004). *Kapital och karriär: informationssökning i forskningens vardagspraktik*. (ISBN 91-89416-08-2) Första uppl. 1999. Skriftserien; 20

Edström, Göte (2000). *Filter, raster, mönster: litteraturguide i teori- och metodlitteratur för biblioteks- och informationsvetenskap och angränsande ämnen inom humaniora och samhällsvetenskap*. (ISBN 91-89416-01-5) Skriftserien; 21

Klasson, Maj (red.) (2000). *Röster: biblioteksbranden i Linköping*. (ISBN 91-89416-02-3) Skriftserien; 22

Stenberg, Catharina (2001). *Litteraturpolitik och bibliotek: en kulturpolitisk analys av bibliotekens litteraturförvärv speglad i Litteraturutredningen L 68 och Folkbiblioteksutredningen FB 80*. (ISBN 91-89416-03-1) Skriftserien; 23

Edström, Göte (2002). *Filter, raster, mönster: litteraturguide i teori- och metodlitteratur för biblioteks- och informationsvetenskap och angränsande ämnen inom humaniora och samhällsvetenskap*. Andra aktualiserade och utökade upplagan. (ISBN 91-89416-05-8) Skriftserien; 24

Sundin, Olof (2003). *Informationsstrategier och yrkesidentiteter: en studie av sjuksköterskors relation till fackinformation vid arbetsplatsen*. (ISBN 91-89416-06-6) Skriftserien; 25

Hessler, Gunnel (2003). *Identitet och förändring: en studie av ett universitetsbibliotek och dess självproduktion*. (ISBN 91-89416-07-4) Skriftserien; 26

Zetterlund, Angela (2004). *Att utvärdera i praktiken: en retrospektiv fallstudie av tre program för lokal folkbiblioteksutveckling*. (ISBN 91-89416-09-0) Skriftserien; 27

Ahlgren, Per (2004). *The effects on indexing strategy-query term combination on retrieval effectiveness in a Swedish full text database*. (ISBN 91-89416-10-4) Skriftserien; 28

Thórsteinsdóttir, Gudrun (2005). *The information seeking behaviour of distance students: a study of twenty Swedish library and information science students*. (ISBN 91-89416-11-2) Skriftserien; 29

Jarneving, Bo (2005). *The combined application of bibliographic coupling and the complete link cluster method in bibliometric science mapping*. (ISBN 91-89416-12-0) Skriftserien; 30

Limberg, Louise, Folkesson, Lena (2006). *Undervisning i informationssökning: slutrapport från projektet Informationssökning, didaktik och lärande (IDOL)*. (ISBN 91-89416-13-9) Skriftserien; 31

Johannisson, Jenny (2006) *Det lokala möter världen: kulturpolitiskt förändringsarbete i 1990-talets Göteborg*. (ISBN 91-89416-14-7) Skriftserien; 32

Gärdén, Cecilia, Eliasson, Anette, Flöög, Eva-Maria, Persson, Christina, Zetterlund, Angela (2006). *Folkbibliotek och vuxnas lärande: förutsättningar, dilemman och möjligheter i utvecklingsprojekt*. (ISBN 91-89416-15-5) Skriftserien; 33

Dahlström, Mats (2006). *Under utgivning: den vetenskapliga utgivningens bibliografiska funktion*. (ISBN 91-89416-16-3) Skriftserien; 34

Nowé, Karen (2007). *Tensions and Contradictions in Information Management: an activity-theoretical approach to information activities in a Swedish youth/peace organisation*. (ISBN 978-91-85659-08-1) Skriftserien; 35

Francke, Helena (2008). *(Re)creations of Scholarly Journals: document and information architecture in open access journals*. (ISBN 978-91-85659-16-6) Skriftserien; 36

Hultgren, Frances (2009). *Approaching the future: a study of Swedish school leavers' information related activities*. (ISBN 978-91-89416-18-5) Skriftserien; 37

Söderlind, Åsa (2009). *Personlig integritet som informationspolitik: debatt och diskussion i samband med tillkomsten av Datalag (1973:289)*. (ISBN 978-91-89416-20-8) Skriftserien; 38

Nalumaga, Ruth (2009). *Crossing to the mainstream: information challenges and possibilities for female legislators in the Ugandan parliament*. (ISBN 91-89416-20-1) Skriftserien; 39

Johannesson, Krister (2009). *I främsta rummet: planerandet av en högskolebiblioteksbyggnad med studenters arbete i fokus*. (ISBN 978-91-89416-21-5) Skriftserien; 40

Kawalya, Jane (2009). *The National Library of Uganda: its inception, challenges and prospects, 1997-2007*. (ISBN 91-89416-22-8) Skriftserien; 41

Gärdén, Cecilia (2010). *Verktyg för lärande: informationssökning och informationsanvändning i kommunal vuxenutbildning*. (ISBN 978-91-98416-23-9) Skriftserien; 42

Ponti, Marisa (2010). *Actors in collaboration: sociotechnical influence on practice-research collaboration*. (ISBN 978-91-89416-24-6) Skriftserien; 43

Jansson, Bertil (2010). *Bibliotekarien: om yrkets tidiga innehåll och utveckling*. (ISBN 978-91-89416-25-3) Skriftserien; 44

Olson, Nasrine (2010). *Taken for granted: the construction of order in the process of library management system decision making*. (ISBN 978-91-89416-26-0) Skriftserien; 45

Gunnarsson, Mikael (2011). *Classification along genre dimensions: exploring a multidisciplinary problem*. (ISBN 978-91-85659-72-2) Skriftserien; 46

Lundh, Anna (2011). *Doing research in primary school: information activities in project-based learning*. (ISBN 978-91-89416-28-7) Skriftserien; 47

Dolatkhah, Mats (2011). *Det läsande barnet: minnen av läspraktiker, 1900–1940*. (ISBN 978-91-89416-29-5) Skriftserien; 48

Frenander, Anders (red.) (2011). *Arkitekter på armlängds avstånd? Att studera kulturpolitik*. (ISBN 978-91-978768-0-3) Skriftserien; 49

Frenander, Anders & Lindberg, Jenny (red.) (2011). *Styra eller stödja: svensk folkbibliotekspolitik under hundra år*. (ISBN 978-91-978768-1-0) Skriftserien; 50

Isah, Esther Ebole (2012). *Physicians' Informations Practicies: A Case Study of a Medical Team at a Teaching Hospital*. (ISBN 978-91-978768-2-7) Skriftserien; 51

Johansson, Veronica (2012). *A Time and Place for Everything? Social Visualisation Tools and Critical Literacies*. (ISBN 978-91-978768-3-4) Skriftserien; 52

Maurin Söderholm, Hanna (2013). *Emergency Visualized: Exploring Visual Technology for Paramedic-Physician Collaboration in Emergency Care*. (ISBN 978-91-978768-5-8) Skriftserien; 53

Lindsköld, Linnéa (2013). *Betydelsen av kvalitet: en studie av diskursen om statens stöd till ny, svensk skönlitteratur 1975-2009*. (ISBN 978-91-978768-5-6) Skriftserien; 54

Pilerot, Ola (2014). *Design Researchers' Information Sharing: The Enactment of a Discipline*. (ISBN 978-91-978768-8-9) Skriftserien; 55

Lassi, Monica (2014). *Facilitating collaboration: Exploring a socio-technical approach to the design of a collaboratory for Library and Information Science*. (ISBN 978-91-981654-0-1) Skriftserien; 56

Dessne, Karin (2014). *In a world of values and views: Information and learning activities in a military setting*. (ISBN 978-91-981654-2-5) Skriftserien; 57

Lindberg, Jenny (2015). *Att bli bibliotekarie: informationssökning och yrkesidentitet hos B & I-studenter och nyanställda högskolebibliotekarier*. (ISBN 978-91-981654-4-9), Skriftserien; 58

Björk, Lars (2015). *How reproductive is a reproduction? Digital transimisson of textbased documents*. (ISBN 978-91-981654-8-7) Skriftserien; 59