Data Mining Scenarios

for the Discovery of Subtypes and the Comparison of Algorithms

Data Mining Scenarios

for the Discovery of Subtypes and the Comparison of Algorithms

PROEFSCHRIFT

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus Prof. mr. P. F. van der Heijden,
volgens besluit van het College voor Promoties
te verdedigen op woensdag 4 maart 2009
te klokke 13.45 uur

door Fabrice Pierre Robert Colas

geboren te Laval, France in 1981.

Promotiecommissie:

Prof. dr. J.N. Kok, LIACS, Universiteit Leiden — Promotor
Dr. Ingrid Meulenbelt, LUMC
Prof. dr. F. Famili, NRC-IIT, Ottawa, Canada
Prof. dr. T.H.W. Bäck, LIACS, Universiteit Leiden
Prof. dr. G. Rozenberg, LIACS, Universiteit Leiden
Prof. dr. B.R. Katzy, LIACS, Universiteit Leiden

This work was carried out under a grant from the Netherlands BioInformatics Center (NBIC).

Data Mining Scenarios for the Discovery of Subtypes and the Comparison of Algorithms
Fabrice Pierre Robert Colas
Thesis Universiteit Leiden
ISBN 978-90-9023888-3

Printed in France

Email: fabrice.colas@grano-salis.net

To my parents

Contents

Introduction

The word *methodology* comes from the latin *methodus* and *logia*. It is defined by the Longman dictionary as *a body of methods and rules employed by a science or a discipline* [lon84]. It is defined by the Trésor de la Langue Française as *un ensemble de règles et de démarches adoptées pour conduire une recherche* [cnr85].

The work presented in this thesis contributes to the field of *data mining* in its *methodological* aspects. This means that we also consider what is needed in applications besides the selection of algorithms.

The outline of the rest of this introduction is as follows. We start by introducing the concept of *data mining scenarios* which underlies our two contributions. Then, we set the context for our first *scenario* that searches for homogeneous *subtypes* of complex diseases like Osteoarthritis (OA) and Parkinson's disease (PD), and it is also used to identify molecule *subtypes* in the field of drug discovery. Next, we introduce our second *scenario* for the *comparison of algorithms* in text classification. Finally, in this introduction, we relate the content of the chapters with our publications.

Data mining scenarios

A *scenario* is defined by the Longman as *(an account or synopsis of) a projected, planned, anticipated course of action or events*.

Therefore, we define a *data mining scenario* as a logical sequence of *data mining* steps to infer patterns from data. A *scenario* will involve the data preparation methods, it will model the data for patterns, it will validate the patterns and assess their reliability.

In this thesis, we present two data mining scenarios: one for *subtype discovery by cluster analysis* and one for the *comparison of algorithms* in text classification.

Part I: Subtype Discovery by Cluster Analysis

For diseases like Osteoarthritis (OA) and Parkinson's disease (PD), we are interested to identify *homogeneous subtypes by cluster analysis* because it may provide a more sensitive classification and hence contribute to the search for the underlying disease mechanism; we do so by searching for subtypes in values of markers that reflect the severity of the diseases.

In drug discovery, subtype discovery of chemical databases may help to understand the relationship between bioactivity classes of molecules, thus improving our understanding of the similarity (and distance) between drug- and chemical-induced phenotypic effects.

To this aim, we developed a *data mining scenario* for *subtype discovery* (see Figure 1 for an overview) and we implemented it as the R SubtypeDiscovery package. This scenario facilitates and enhances the task of discovering subtypes in data. It features various data preparation techniques, an approach that repeats data modeling in order to select for the number of subtypes and/or the type of model, along with a selection of methods to characterize, compare and evaluate statistically the most likely models.

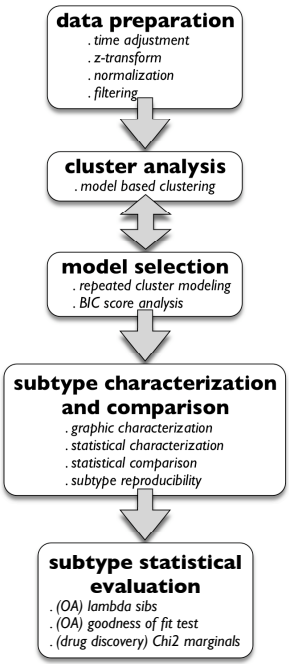

Figure 1: *Workflow of our subtype scenario.*

In Chapter 1, we describe the background of the current three applications.

In Chapter 2, we discuss the different steps of our subtyping scenario. We motivate our choice for a particular clustering approach and we present additional methods that help us to select, characterize, compare and evaluate cluster results. Additionally, we discuss some issues that occur when preparing the data.

Chapter 3 investigates two key elements when searching for subtypes. First, when preparing the data, because age in OA and disease duration in PD are known to contribute largely to the variability, *how to deal with the time dimension?* And second, as we aim for reliable subtypes exhibiting robustness to little changes in the data, *how to assess the reliability of the results?*

In Chapter 4, we report subtyping results in OA and PD, as well as in drug discovery. Because of the confidentiality issues regarding the clinical data and the drug discovery data, only a subset of the results is presented.

Finally, Chapter 5 describes the implementation as the R SubtypeDiscovery package of our *scenario*. By making it available as a package, it can be used in the search for subtypes in other application areas.

Part II: Automatic Text Classification

In text categorization, we can use machine learning techniques to build systems which are able to automatically classify documents into categories. For this purpose, the most often used feature space is the *bag of words* feature space where each dimension corresponds to the number of occurrences of the words in a document. This feature space is particularly popular because of its rather simple implementation and because of its wide use in the field of information retrieval.

Yet, the task of classifying text documents into categories is difficult because the size of the *bag of words* feature space is high; in typical problems, its size commonly exceeds the tens of thousands of words. Furthermore, what makes the text classification problem complex is that usually the number of training documents is several orders of magnitude smaller than the feature space size. Therefore, as some algorithms can not deal well with such situations (more features than documents), a number of studies investigated how to reduce the feature space size [Yan97; Rog02; For03].

Next, among the machine learning algorithms applied to text classification, the most prominent one is certainly *linear* Support Vector Machines (SVM). It was first introduced to the field of text categorization by Joachims [Joa98]. Then, SVM were systematically included in every subsequent comparative study on text classification algorithms [Dum98; Yan99b; Zha01; Yan03]; their conclusions suggest that SVM is an outstanding method for text classification. Finally, the extensive study of Forman also confirms SVM as outperforming other techniques for text classification [For03].

However, in several large studies, SVM did not systematically outperform other classifiers. For example the work of [Liu05] showed the difficulty to extend

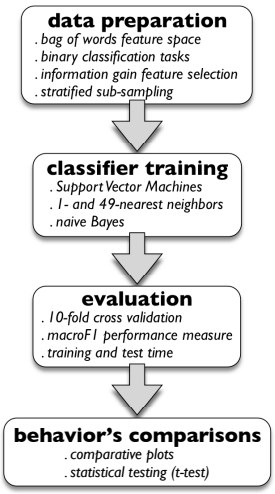

Figure 2: *Workflow of the data mining scenario to compare algorithms in the field of text classification.*

SVM to large scale taxonomies. Other papers showed that depending on experimental conditions, the k nearest neighbors classifier or the naive Bayes classifier can achieve better performance [Dav04; Col06b; Sch06].

Therefore, we consider in this thesis the problem of conducting comparative experiments between text classification algorithms. We adress this problem by defining a *data mining scenario* for the *comparison of algorithms* (see Figure 2 for an overview). As we apply this scenario to text classification algorithms, we are especially interested to assess whether one algorithm, in our case SVM, consistently outperforms others. Next, we also use this scenario to develop a better understanding of the behavior of text classification algorithms.

In this regard, it was shown in [Dae03] that experimental set-up parameters can have a more important effect on performance than the individual choice of a particular learning technique. Indeed, classification tasks are often highly unbalanced and the way training documents are sampled has a large impact on performance. Similarly, if one wants to do *fair* comparisons between the classifiers, the aggregating multi-class strategy which generalizes binary classifiers to the multi-class problem, e.g. SVM one-versus-all, should be taken into account. In this regard, [Für02] showed that better results are achieved if classifiers natively able to handle multi-class are reduced to a set of binary problems and then, aggregated by pairwise classification.

For these reasons, we only study situations where the number of documents in each class is the same and we choose binary classification tasks as the baseline

of this work. Besides, selecting the right parameters of the SVM, e.g. the upper bound of Lagrange multipliers (C), the kernel, the tolerance of the optimizer (ϵ) and the right implementation is non-trivial. Here, we only consider *linear* SVM because a *linear* kernel is commonly regarded as yielding to the same performance than non-linear kernels in text categorization [Yan99b].

In Chapter 6, we start by describing our *data mining scenario* to compare text classification algorithms. This scenario focuses especially on the definition of the experimental set-up which can influence the result of the comparisons; in this sense, we discuss *fair* classifier comparisons. Then, we give some background on the k nearest neighbors classifier, the naive Bayes classifier and the SVM. Next, we describe the dimensions of the experimentation, our evaluation methodology and the experimental data.

Because many authors did present SVM as the leading classification method, we analyze in Chapter 7 its behavior in comparison to those of the k-nearest neighbors classifier and the naive Bayes on a large number of classification tasks.

In Chapter 8, we develop a better understanding of SVM behavior in typical text categorization problems represented by sparse *bag of words* feature spaces. To this end, we study in detail the performance and the number of support vectors when varying the training set size, the number of features and, unlike existing studies, also the SVM free parameter C, which is the upper bound of the Lagrange multipliers in the SVM dual representation.

Publications

The two *data mining scenarios* presented in this thesis are based on a number of publications in refereed international conferences. In the following, we give an overview of these publications, first for part I and later for part II of the thesis.

Part I: Subtype Discovery by Cluster Analysis

In Chapter 2, we give details about the methodological aspects underlying our *subtyping scenario*. This scenario stems from an initial discussion in:

- F. Colas, I. Meulenbelt, J.J. Houwing-Duistermaat, P.E. Slagboom, and J.N. Kok. A comparison of two methods for finding groups using heat maps and model based clustering. In *Proceedings of the 27th Annual International Conference of the British Computer Society (SGAI), AI-2007, Cambridge, UK*, pp. 119-131. December 2007.

The complete scenario was presented in:

- F. Colas, I. Meulenbelt, J.J. Houwing-Duistermaat, M. Kloppenburg, I. Watt, S.M. van Rooden, M. Visser, H. Marinus, E.O. Cannon, A. Bender, J. J. van Hilten, P.E. Slagboom, and J.N. Kok. A scenario implementation in R

for subtype discovery examplified on chemoinformatics data. In *Proceedings of Leveraging Applications of Formal Methods, Verication and Validation*, vol.17 of CCIS, pp. 669-683. Springer, 2008.

In Chapter 3, following the use of our scenario in complex diseases subtyping, we present an additional method that we developed to select the best time adjustment by cluster result reliability assessment; this was published in:

- F. Colas, I. Meulenbelt, J.J. Houwing-Duistermaat, M. Kloppenburg, I. Watt, S.M. van Rooden, M. Visser, H. Marinus, J.J. van Hilten, P.E. Slagboom, and J.N. Kok. Stability of clusters for different time adjustments in complex disease research. In *Proceedings of the 30th Annual International IEEE EMBS Conference (EMBC08), Vancouver, British Columbia, Canada.* August 2008.

We report in Chapter 4 *subtyping* results on OA, PD and in drug discovery. Some of the resulting subtypes are submitted for publication. For PD, this is the case for:

- S.M. van Rooden, F. Colas, M. Visser, D. Verbaan, J. Marinus, J.N. Kok, and J.J. van Hilten. Discovery and validation of subtypes in parkinson's disease. *Submitted for publication,* 2008.

and

- S.M. van Rooden, M. Visser, F. Colas, D. Verbaan, J. Marinus, J.N. Kok, and J.J. van Hilten. Factors and subtypes in motor impairment in parkinson's disease: a data driven approach. *Submitted for publication,* 2008.

Finally, in Chapter 5 we describe the implementation as an R package. First, it was presented at the BioConductor '08 conference in Seattle (Washington, USA) and second, in:

- F. Colas, S.M. van Rooden, I. Meulenbelt, J.J. Houwing-Duistermaat, A. Bender, E.O. Cannon, M. Visser, H. Marinus, J.J. van Hilten, P.E. Slagboom, and J.N. Kok. An R package for subtype discovery examplified on chemoinformatics data. In *Statistical and Relational Learning in Bioinformatics (StReBio ECML-PKDD08 Workshop).* September 2008.

Part II: Automatic Text Classification

In Chapter 6, we describe our scenario to conduct comparative experiments between text classification algorithms. This *scenario* stems initially from our master thesis [Col05] and it was step-by-step expanded with additional contributions [Col06a; Col06b], in particular with [Col07b].

Next, in Chapter 7, we detail the result of our large scale comparison of text classification algorithms; our first results were published in:

- F. Colas and P. Brazdil. Comparison of svm and some older classification algorithms in text classification tasks. In *Proceedings of the IFIP-AI 2006 World Computer Congress, Santiago de Chile, Chile,* IFIP 217, pp.169-178. Springer, August 2006.

while extended results were presented in:

- F. Colas and P. Brazdil. On the behavior of svm and some older algorithms in binary text classification tasks. In *Proceedings of Text Speech and Dialogue (TSD2006), Brno, Czech Republic,* Lecture Notes in Computer Science 4188, pp. 45-52. Springer, September 2006.

Finally, in Chapter 8, we conducted experiments in order to better understand the behaviors of SVM in text classification; these results were published in

- F. Colas, P. Paclík, J.N. Kok, and P. Brazdil. Does svm really scale up to large bag of words feature spaces? In *Proceedings of Intelligent Data Analysis (IDA2007), Ljubljana, Slovenia,* Lecture Notes in Computer Science 4723, pp.296-307. Springer, September 2007.

Part I

Subtype Discovery by Cluster Analysis

Chapter 1

Application Domains

We present the three application domains for our subtyping scenario. The scenario will be applied to Osteoarthritis, Parkinson's disease and drug discovery. For each application domain, we briefly describe the domain, motivate why subtyping is interesting and give details about the datasets that are used later in the thesis.

1.1 Introduction

In the thesis we will introduce a data mining scenario to identify homogeneous subtypes in data. Here, we present three application areas for this scenario in three subsections: Osteoarthritis, Parkinson's disease and drug discovery.

1.2 Osteoarthritis

Osteoarthritis (OA) is a disabling common late onset disease of the joints characterized by cartilage degradation and the formation of new bone [Meu97; Riy06; Min07]. The joint damage is caused by a mixture of systemic factors that predispose to the disease and of local mechanical factors. Together, these factors may dictate the *distribution* and the *severity*.

First, regarding the *distribution* of OA, although OA can occur at any joint, it is most commonly observed in the lumbar and cervical spine, hands, knees and hips. Further, when a single joint is affected, OA is viewed as *localised* but when there are multiple joints affected, it is considered to be *generalised*.

Second, regarding the *severity* of OA, its diagnosis can rely on *radiographic* characteristics (ROA) as specified by Kellgren and Lawrence in [Kel57]: the severity of the radiographic features is scored in terms of a five-points ordinal grading scheme between zero and four. Besides these features, OA is described *clinically*

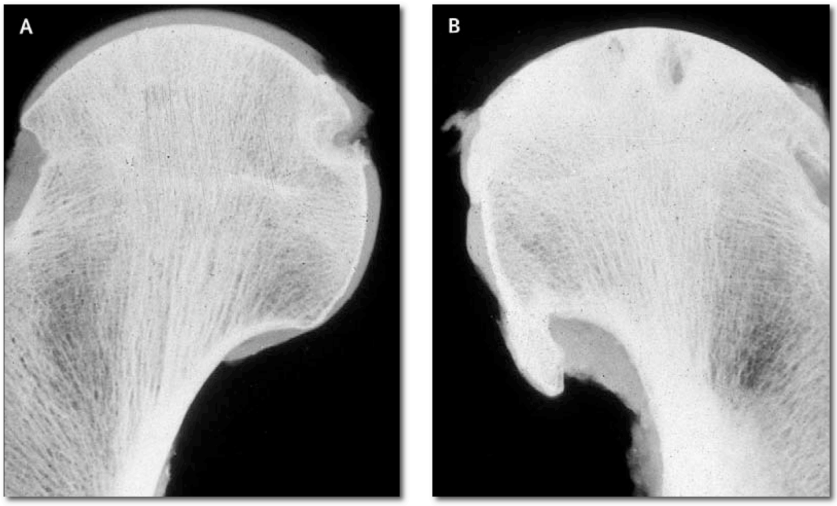

Figure 1.1: *Radiograph of (A) normal and (B) osteoarthritic femoral head. Radiographic image of osteoarthritic joints shows marginal osteophytes, change in shape of bone, subchondral bone cysts, and focal area of extensive loss of articular cartilage (with permission [Die05]).*

by joint pain, tenderness, limitation of movement, friction sensation between bone and cartilage, occasional effusion, inflammation.

The incidence of ROA may result of *systemic* factors like age, gender, genetics, bone density and obesity but also of *local* factors like joint injury, muscle weakness, malalignment and developmental deformity. However, clinical *symptoms* and radiographic characteristics of OA often correlate poorly. In fact, the prevalence of symptomatic OA is considerably lower than the one of ROA because of the high proportion of subjects not having joint pains.

For these reasons, OA is now regarded as a group of disctinct overlapping diseases whose particular phenotype may reflect different pathological processes. As a result, OA is referred to as a *complex disease* since both environmental and genetic determinants influence its aetiology. It is likely that most individuals are affected by OA because of a combination of environmental and genetic factors.

Why subtyping OA? Our investigations may allow to study the spread of the disease across different joint sites and to show whether it is stochastic or follows a particular pattern depending on the underlying disease aetiology. For this purpose, our subtyping scenario can provide a tool to identify and characterize subtypes of OA (e.g. in terms of hereditability). Such subtypes could contribute to

elucidate the clinical heterogeneity of OA and therefore enhance the identification of the disease pathways (genetics, pathophysiological mechanisms).

Patients We will consider a study called GARP which consists of Caucasian sibling pairs of Dutch ancestry with predominantly symptomatic OA at multiple sites; more background on GARP and already published work can be found online [LUM08a]. Here we describe the study briefly.

Symptomatic OA of a joint was defined as the presence of symptoms of OA and ROA. The scoring of symptomatic OA was previously described in [Riy06]. Probands (ages 40-70 years) and their siblings had OA at multiple joint sites of the hand or in two or more of the following joint sites: hand, spine (cervical or lumbar), knee or hip. Subjects with symptomatic OA in just one site were required to have structural abnormalities in at least one other joint site, defined by the presence of ROA in any of the four joints or the presence of two or more Heberden's nodes, Bouchard's nodes, or squaring of at least one carpometacarpal joint on physical examination.

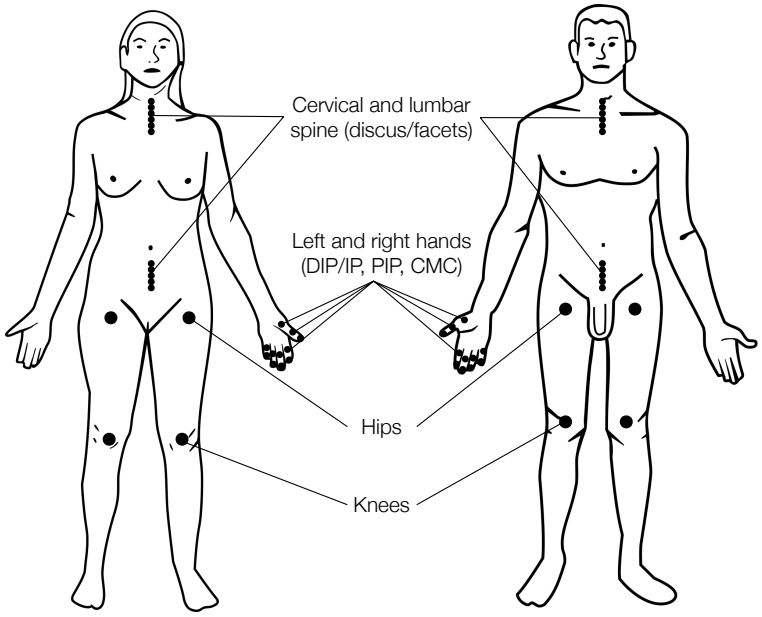

Figure 1.2: *The different joint locations assessed for ROA.*

For an overview of the different joint locations where ROA was assessed, see Table 1.1 and Figure 1.2. The scoring is done using the Kellgren and Lawrence

ordinal grading scheme [Kel57]. As some individuals had an incomplete ROA phenotype, they were discarded and we also decided to restrict our analysis to family sibships involving only two members (proband / sibling); we left out a total of 13 individuals. Therefore, for the analysis presented in this thesis, we analysed the ROA profiles of 211 sibling pairs ($N = 422$ patients).

Table 1.1: *Table describing the 45 joint locations where the individuals were measured.*

Main site	Joint location
Hips	Left and Right
Knees	Left and Right
Hands DIP+IP	Thumb (IP) and DIP 2, 3, 4, 5 on the Left and Right
Hands PIP	PIP 2, 3, 4, 5 on the Left and Right
Spine Discus	Cervical 23, 34, 45, 56, 67 and Lumbar 12, 23, 34, 45, 56
Spine Facets	Cervical 12, 23, 34, 45, 56, 67 and Lumbar 12, 23, 34, 45, 56

1.3 Parkinson's disease

Parkinson's disease (PD) is a degenerative disorder of the central nervous system that often impairs the sufferer's motor skills and speech, as well as other functions [Jan08]. In the following two paragraphs, we give further characteristics of PD taken from the online article of PD on the Wikipedia [Wik08]:

> PD is characterized by muscle rigidity, tremor, a slowing of physical movement (bradykinesia) and, in extreme cases, a loss of physical movement (akinesia). The primary symptoms are the results of decreased stimulation of the motor cortex by the basal ganglia, normally caused by the insufficient formation and action of dopamine[1], which is produced in the dopaminergic neurons of the brain. Secondary symptoms may include high level cognitive dysfunction and subtle language problems. PD is both chronic and progressive.
>
> PD is the most common cause of chronic progressive parkinsonism, a term which refers to the syndrome of tremor, rigidity, bradykinesia and postural instability. PD is also called "primary parkinsonism" or "idiopathic PD" (classically meaning having no known cause although

[1] *dopamine*: a chemical compound that occurs especially as a substance that transmits electrical impulses from one nerve to another (neurotransmitter) in the brain and as an intermediate compound in the synthesis of adrenalin in body tissue (from the Longman dictionary).

this term is not strictly true in light of the plethora of newly discovered genetic mutations). While many forms of parkinsonism are "idiopathic", "secondary" cases may result from toxicity most notably of drugs, head trauma, or other medical disorders. The disease is named after the English physician James Parkinson; who made a detailed description of the disease in his essay: "An Essay on the Shaking Palsy" (1817).

Why subtyping PD? Among the PD patients, there is marked heterogeneity, both in presence and severity of different impairments and in other variables like age at onset or family history. The progression and course of PD vary widely among individual patients and understanding more about these PD subtypes and how they relate to an individual's disease course could improve patient treatment with existing therapies and help develop new treatments (e.g. see [MJF08] PD-subtype research program). Until now, studies on heterogeneity that are using a large cohort of patients and that are assessing the full spectrum of PD are lacking.

Data acquisition We will use data from both the PROPARK (PROfiling PARKinson's disease) and the SCOPA (SCales for Outcomes in PArkinsons disease) projects. In order to evaluate the longitudinal course of PD, the PROPARK project was started in 2003 [LUM08b]. In this study, a cohort of 420 patients is screened annually on: the whole spectrum of impairments, the problems related to daily living activities and the quality of life. These instruments of measure are derived from the SCOPA project which purpose was to evaluate and / or to develop valid and reliable instruments that are specific for PD, for more details see [LUM08b; Mar03b; Mar03c; Vis04; Mar04; Vis06; Vis07].

Cohort recruitment We first describe how the cohort was recruited. It is stemming from patients from two university hospitals (Leiden and Rotterdam) and nine regional hospitals in the western part of the Netherlands.

As presented in [Roo08a], the diagnosis of PD was made according to the United Kingdom Brain bank criteria by a movement disorder specialist [Gib88]. The clinical diagnosis of PD was verified at each assessment. During follow-up, patients who developed symptoms and signs that pointed towards other forms of parkinsonism, were excluded from the cohort. Furthermore, participating patients had to be able to comprehend the Dutch language. Patients were not excluded from the SCOPA-PROPARK cohort based on their comorbidity and therefore the cohort provides a better reflection of the general PD population than most trial cohorts.

For the study on subtypes, patients having undergone stereotactical surgery were excluded because of potential confounding effects. At baseline, patients were stratified based on age at onset ($< / > 50$ years) and disease duration ($< / >$ 10 years) because these characteristics are important predictors of PD features

and medication-induced complications [Kos91]. To avoid a bias towards recruiting the less severely affected patients and to decrease the drop-out rate, more severely affected patients were offered an assessment at home. All patients gave written informed consent to participate in the study.

Table 1.2: *Measures of impairments of Parkinson's disease in the SCOPA-PROPARK cohort.*

Cognitive functioning: SCOPA-COG Sumscore	Memory Attention Executive functioning Visuospatial functioning
Motor symptoms: SPES/SCOPA - motor Sumscore	Trembling Stiffness Slowness of movement Axial (including rise, postural instability, gait) Axial2 (including speech, freezing and swallowing)
Motor complications: SPES/SCOPA motor complications	Motor fluctuations Dyskinesia
Psychiatric functioning: SCOPA-PC items 1-5, Psychotic symptoms	
Autonomic functioning: SCOPA-AUT Sumscore	Gastro-intestinal dysfunction (reduced to three items: full quickly, obstipation, hard strain) Urinary dysfunction Cardiovascular dysfunction
Nighttime sleepiness: SCOPA-sleep night-time sleeping Sumscore	
Daytime sleepiness: SCOPA-sleep daytime sleepiness Sumscore	
Depression: Beck Depression Inventory Sumscore	

Assessments The annual assessments encompassed self-assessed scales that patients completed at home as well as a supplementary examination consisting of researcher-administered assessment scales in the LUMC (Leiden University Medical Center), see Table 1.2 and [Mar03b; Mar03c; Vis04; Mar04; Vis06; Vis07] for details. In addition, socio-demographics, age at onset, disease duration, and familial occurrence of PD was recorded at baseline. At each assessment, medication was recorded. Patients were optimally treated and the assessments were executed while the patients are in the so-called on state.

Dataset used in the thesis The participants, have a baseline measurement and are followed-up over 3 years with an interval of a year. At baseline, 417 patients were included. Yet, as 18 patients were subjects to stereotactical surgery and as 66 patients exhibited incomplete PD severity profiles (missing values), subtyping

Table 1.3: *Description of the dataset for the subtyping analysis on year one ($N = 333$).*

Sex: male / female (% male)	220 / 113 (66%)
	Mean (SD)
Age at year one	60.8 (11.4)
Age at onset at year one	50.9 (11.9)
Disease duration at year one	9.9 (6.2)

analysis on year one were conducted on 333 patients. In Table 1.3, we describe the dataset characteristics for year one. For further details consult [Roo08a].

1.4 Drug discovery

A *drug* is *a synthetic or natural substance used as, or in the preparation of, a medication [...], for use in the diagnosis, cure, treatment, or prevention of disease* [lon84].

In this thesis, we conducted *subtyping* analyses in the field of drug discovery based on a list of banned stimulating drugs (i.e. molecules) in sports. This list is maintained by the World Anti Doping Agency (WADA, `www.wada-ama.org`); here, we use the list of 2008.

Why subtyping molecular databases? Subtype discovery of chemical databases may help to understand the relationship between bioactivity classes of molecules, thus improving our understanding of the similarity (and distance) between drug- and chemical-induced phenotypic effects.

Calculating the properties of molecules In order to build statistical models of molecules, they first need to be described in a format understandable by computer algorithms. This step is usually referred to as the calculation of molecular "descriptors". These properties can serve as numerical descriptions (features) of molecules in other calculations like QSAR (Quantitative Structure-Activity Relationships), diversity analysis, combinatorial library design and in this thesis, *subtyping*. In our work, we used descriptors as implemented in MOE (Molecular Operating Environment) [CCGI08]. However, there are many possible sets of features because any molecular property may be used as a molecular descriptor.

These properties are of three types. First, there are *2D descriptors* which only use the atoms and connection information of the molecule for the calculation. They can be calculated from the connection table of a molecule; therefore, they do no depend on the conformation of a molecule. Second, there are *internal 3D descriptors* (i3D) that use the 3D coordinate information of each molecule;

coordinates are considered invariant to rotations and translations of the conformation. Third, there are the *external 3D descriptors* (x3D) where the 3D coordinate information is also used but this time in an absolute frame of reference. A frame of reference can be a receiving molecule to which the molecules should bind themselves; yet, as several orientations are possible, the most likely one is determined by a *docking*-method.

Selected molecular properties In this thesis we conduct *subtyping* analyses on 2D molecular properties; we do not use the 3D descriptors. In Table 1.4, we list the six classes of descriptors for which we selected a number of molecular properties; these properties are explained in Tables A.1, A.2, A.3, A.4, A.5 and A.6, which can be found in Appendix A of this thesis.

Table 1.4: *2D molecular properties that we selected to describe and characterize the databases of molecules.*

Atom and bond counts (ABC)	a_aro, a_count, a_heavy, a_IC, a_ICM, a_nB, a_nBr, a_nC, a_nCl, a_nF, a_nH, a_nI, a_nN, a_nO, a_nP, a_nS, b_1rotN, b_1rotR, b_ar, b_count, b_double, b_heavy, b_rotN, b_rotR, b_single, b_triple, chiral, chiral_u, lip_acc, lip_don, lip_druglike, lip_violation, nmol, opr_brigid, opr_leadlike, opr_nring, opr_nrot, opr_violation, rings, VAdjEq, VAdjMa, VDistEq, VDistMa
Adjacency and distance matrix descriptors (ADDM)	balabanJ, diameter, petitjean, petitjeanSC, radius, weinerPath, weinerPol
Kier and Hall connectivity and kappa shape indices (KH)	KierFlex, zagreb
Partial charge descriptors (PCD)	PC., PC..1, Q_PC., Q_PC..1, Q_RPC., Q_RPC..1, Q_VSA_FHYD, Q_VSA_FNEG, Q_VSA_FPNEG, Q_VSA_FPOL, Q_VSA_FPOS, Q_VSA_FPPOS, Q_VSA_HYD, Q_VSA_NEG, Q_VSA_PNEG, Q_VSA_POL, Q_VSA_POS, Q_VSA_PPOS, RPC., RPC..1
Pharmacophore feature descriptors (PFD)	a_acc, a_acid, a_base, a_don, a_hyd, vsa_acc, vsa_acid, vsa_base, vsa_don, vsa_hyd, vsa_other, vsa_pol
Physical properties (PP)	apol, bpol, density, FCharge, logP.o.w., logS, mr, reactive, SlogP, SMR, TPSA, vdw_area, vdw_vol, Weight

Dataset used in the thesis The dataset is composed of substances taken from the 2008 WADA Prohibited List together with molecules having similar biological activity and chemical structure from the MDL Drug Data Report database; it was generated by Edward O. Cannon. In previous work [Can06a; Can06b; Can08], the purpose was to partition the space of chemical substances into subgroups of bioactivity classes using classification algorithms; the dataset used was the `wada2005` dataset which is based on the 2005 prohibited list. In this work, we use clustering algorithms to identify the subgroups.

The molecules may belong to one of the ten activity classes: the β blockers, anabolic agents, hormones and related substances, β-2 agonists, hormone antagonists and modulators, diuretics and other masking agents, stimulants, narcotics, cannabinoids and glucocorticosteroids. Then, the molecules were imported into MOE from which all 184 two dimensional descriptors were calculated. We embed the `wada2008` dataset within our R SubtypeDiscovery package.

1.5 Concluding remarks

We presented three domains where subtyping can be used to enhance the understanding of the problem.

In OA, the aim of subtyping is to study the spread of the disease across different joint sites and to show whether it is stochastic or follows a particular pattern (subtype).

In PD, as the spread and the course of PD vary widely among individual patients, understanding more about these PD subtypes could improve patient treatment with existing therapies and help develop new treatments.

In drug discovery, subtyping chemical databases may help to understand the relationship between bioactivity classes of molecules, thus improving our understanding of the similarity (and distance) between drug- and chemical-induced phenotypic effects.

Therefore, subtyping is a general problem and in the following chapters, we will present our data mining scenario to search for subtypes in data. In this chapter we described three application areas of our subtyping scenario: in medical research on OA and PD, and in drug discovery. For each application, we introduced the domain and then we motivated why subtyping is interesting. Finally, we explained how the datasets were collected or generated and the type of data that we ran our analyses on.

Chapter 2

A Scenario for Subtype Discovery by Cluster Analysis

In this chapter, we present our subtyping scenario*. First, we discuss data processing issues when preparing the data before analysis. Next, we motivate our choice for a particular clustering method. Then, to select for a number of subtypes or a model, we describe a computational approach that repeats data modeling. Finally, we report on methods to characterize, compare and evaluate the most likely subtypes.*

2.1 Introduction

To identify homogeneous subtypes of complex diseases like Osteoarthritis (OA) and Parkinson's disease (PD) and to subtype chemical databases, we developed a scenario mimicking a cluster analysis process: from data preparation to cluster evaluation, see Figure 2.1 for an illustration of our scenario. It implements various data preparation techniques to facilitate the analysis given different data processing. It also features a computational approach that repeats data modeling in order to select for a number of subtypes or a type of model. Additionally, it defines a selection of methods to characterize, compare and evaluate the top ranking models.

The outline of the rest of the chapter is as follows. First, we describe data preparation issues with methods to answer them, as well as the clustering method. Second, we report methods to characterize, compare and evaluate cluster results. Illustrations of our scenario throughout this chapter are from medical research on OA and PD.

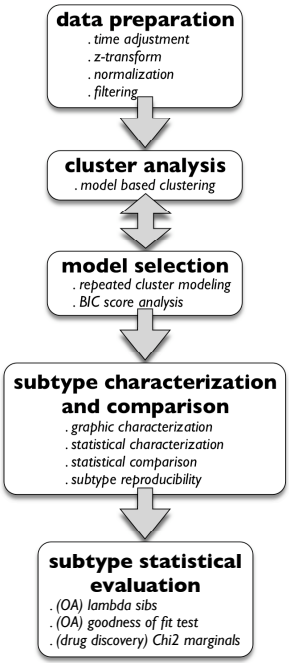

Figure 2.1: *Workflow of a subtype discovery analysis.*

2.2 Data preparation and clustering

We aim to identify homogeneous and *reliable* subtypes. Hence, cluster results should be reproducible and the clusters should characterize true underlying patterns, not the incidental ones. We discuss in this section the removal of the *time* dimension in the OA and PD datasets, the *reliability* and *validity* of cluster results and give a brief overview of model based clustering.

2.2.1 Data preparation

As data preparation can influence largely the result of data analysis, our scenario implements various methods to transform and process data, e.g. computing the z-scores of variables to obtain scale-invariant quantities, normalizing according to the Euclidean norm (L_2), the Manhattan distance (L_1), the maximum and centering with respect to the empirical mean, the median or the minimum.

As in the overal severity of OA and PD, respectively age or disease duration (thereafter, the *time*) are known to play a major role, we may want to remove their dimension in the data because we do not want to model clusters only characterized

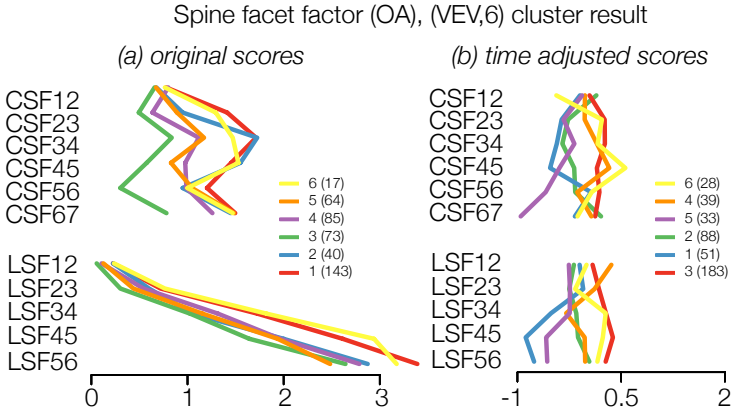

Figure 2.2: *For OA, we show results of two cluster analyses on the spine facets factor with a VEV model having six mixtures (the VEV model will be explained in section 2.2.3). In (a), the modeling is on the original ROA scores, i.e. between* $[0, 4]$ *and in (b), on the time adjusted scores, i.e. z-scores. This illustrates how the time influences the cluster results. The arrangement of the variables mimicks the disposition of the cervical and lumbar vertebrae, from top to bottom.*

by them. In Figure 2.2, we report a visualization of two cluster analyses on OA data: we conducted the clustering on the original scores and on the time adjusted scores; it shows how much the time influences the modeling. So, to remove the *time* dimension for the data, we first perform regression on the *time* for each variable and next, we conduct cluster analyses on the residual variance.

If we denote by α and β the estimated intercept and coefficient vectors of the regression, by the matrix $\mathbf{X}$ the data where x_{ij} refers to measurement j of observation i, then the regression is given by

$$x_{ij}(t_i) = \alpha_j + \beta_j g(t_i) + \varepsilon_{ij}, \tag{2.1}$$

$$\varepsilon_{ij} = x_{ij}(t_i) - \alpha_j - \beta_j g(t_i). \tag{2.2}$$

The ε_{ij} refer to the residual variation and $g(t) \in \left\{log(t), \sqrt{t}, t, t^2, exp(t)\right\}$ (the time effect is not necessarily linear). Additionally, residuals ε_{ij} should distribute normally around zero for each variable j, as illustrated in Figure 2.3.

2.2.2 The reliability and validity of a cluster result

In our data mining scenario for subtype discovery by cluster analysis, hierarchical clustering [Sne73] or k-means [Has01] did not match our expectations in terms of *reliability* and *validity* (see discussion below). Instead, we selected model

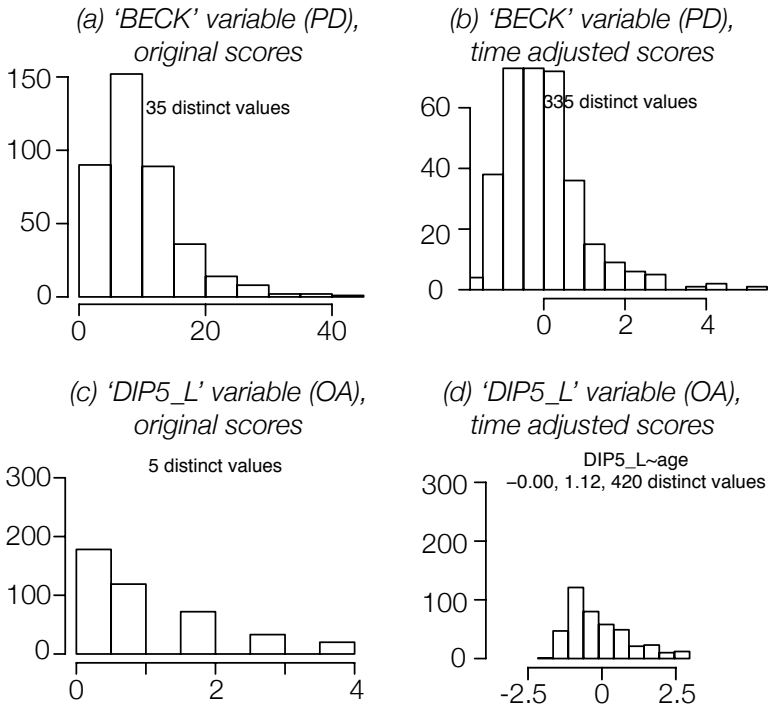

Figure 2.3: *These four figures illustrate the original and the time-adjusted data distributions of variables DIP5_L and beck, which respectively pertain to OA and PD analyses. Such histograms are obtained when plotting a dataset class (*`cdata`*) of the* R Subtype-Discovery *package. To be valid, the residuals ε_{ij} of the regression on the time should distribute normally around zero for each variable j.*

based clustering that relies on the EM-algorithm (Expectation Maximization) for parameter estimation [Fra99; Fra02b; Fra03; Fra06]. In the following two paragraphs, we discuss the *reliability* and *validity* of the k-means, the hierarchical and the model based clustering.

k-means and hierarchical clustering First, in terms of *reliability*, the cluster results should be consistent when we repeat the analysis. For example, when we repeat the k-means, solutions may differ because of the different starting values. Second, both the hierarchical clustering and the k-means clustering depend on distance measures which do not necessarily mimic the data distribution of the clusters; however, to be *valid*, the clusters should be understandable which is not evident when they are defined in terms of distances, especially for non-euclidean ones.

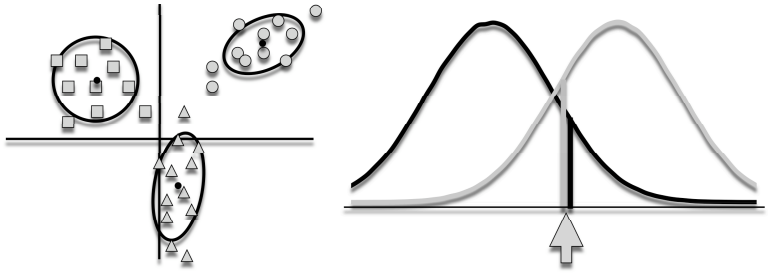

Figure 2.4: *On the left, we show a simple modeling with three mixtures in two dimensions which are defined by their center μ_k and their geometry Σ_k with $k = 1, 2, 3$. On the right, we illustrate two mixtures on a single dimension. Membership of the gray is most likely. Membership of the black is less likely.*

In fact, the clusters should also be distinguishable, which becomes an issue as the modeling takes place in higher dimensions because the distance-based algorithms are sensitive to the curse of dimensionality [Bey99]. And finally, another aspect that hampers especially the hierarchical clustering, concerns the numerous parameters that can only be set subjectively. The book [Sne73] gives a detailed description of all the possible parameters.

Model based clustering To be fair, *reliability* issues also exist for clustering by mixture of Gaussians because it relies on the EM-algorithm. To estimate model parameters, EM optimizes iteratively the model likelihood and as a matter of fact, different starting values for EM may lead to different cluster results. Therefore, an important issue concerns the sensibility to different starting values of the mixture modeling. In this regard, Fraley and Raftery decided to initiate systematically their EM-algorithm by a model based hierarchical clustering [Fra99]. This choice ensures the reproducibility of the cluster results because two repeats of the mixture modeling will initiate EM equally.

Concerning the *validity* issue, mixture modeling not only reports the estimated center of each mixture but also it estimates the covariance structure. Therefore, it also yields estimates of the cluster membership certainty. Further, as shown in [Ban93] and as illustrated with an example in Figure 2.4, the framework relies on the concept of reparameterization of the covariance matrix which enables to select and adapt the level of complexity of the covariance by controlling its geometry. Hence, the analysis offers a range of models that involve varying number of parameters to estimate. For instance, a particular model may set an equal data distribution for all mixtures, while another may discard the estimations of the covariates in the model.

2.2.3 Clustering by a mixture of Gaussians

In this subsection, as in [Fra99; Fra02b; Fra03; Fra06], we describe clustering by mixture modeling.

First, the likelihood function of a mixture of Gaussians is defined by

$$\mathcal{L}_{MIX}(\theta, \tau|\mathbf{x}) = \prod_{i=1}^{N} \sum_{k=1}^{G} \tau_k \phi_k(\mathbf{x}_i|\mu_k, \Sigma_k), \tag{2.3}$$

where $\mathbf{x}_i$ is the i^{th} of N observations, G is the number of components and τ_k the probability that an observation belongs to the k^{th} component (hence $\tau_k \geq 0$ and $\Sigma_{k=1}^{G} \tau_k = 1$). Then, the likelihood of an observation $\mathbf{x}_i$ to belong to the k^{th} component is given by

$$\phi_k(\mathbf{x}_i|\mu_k, \Sigma_k) = \frac{\exp\{-\frac{1}{2}(\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1}(\mathbf{x}_i - \mu_k)\}}{\sqrt{det(2\pi\Sigma_k)}}. \tag{2.4}$$

The reparameterization proceeds by eigenvalue decomposition of the covariance matrix Σ_k

$$\Sigma_k = D_k \Lambda_k D_k^T. \tag{2.5}$$

This decomposition depends on the diagonal matrix Λ_k of the eigenvalues and on the eigenvector matrix D_k which determines the orientation of the principal components. The matrix Λ_k can be decomposed further into

$$\Lambda_k = \lambda_k A_k \tag{2.6}$$

with A_k the geometrical shape and λ_k the largest eigenvalue.

In their framework, Fraley and Raftery control the structure of Σ_k using constraints on the three parameters λ_k, A_k and D_k. The constraints are expressed in letters $\{I, E, V\}$ which stand for identical, equal and variable respectively.

- λ_k refers to the relative size or the *scale* of the k^{th} mixture which may be equal for all mixtures (E) or vary (V).
- A_k specifies the *geometrical shape* which may limit the mixtures to spherical shapes (I), to equally elongated shapes for all mixtures (E), or to varying ones (V).
- D_k characterizes the principal orientations of the covariance which may simply coordinate along the axes (I) and therefore neglect estimation of the covariates; but when considering covariates, we may select an equal orientation for all mixtures (E) or a different one (V).

Hence, a constraint is expressed by three letters, one for each parameter. For example, the constraint VVI refers to a model where a diagonal covariance matrix will be estimated for each cluster; therefore, no covariate is estimated.

EM-algorithm For a given number of mixtures and a covariance model, the EM-algorithm is used to estimate the model parameters [Dem77]. It alternates iteratively between a step of *Expectation* to estimate for each observation its cluster membership likelihood, and a step of *Maximization* to identify the parameters that maximize the model likelihood. The iterative process stops as likelihood improvements become small.

An important concern for the EM algorithm is the dependency on the starting point. As mentioned before, Fraley and Raftery propose to systematically initialize EM with a model based hierarchical clustering. Though, a common strategy is to start EM from several random points and then to study the sensibility of the cluster results to these changes. We selected this second strategy for our data mining scenario.

2.3 Model selection

The larger the number of parameters, the more likely it is that our model may overfit the data which restricts its generality and comprehensiveness. In this section, we discuss a score that we use as a guidance to compare models involving different numbers of parameters and an approach to conduct a *valid* model selection.

2.3.1 A score to compare models

For model selection, Kass and Raftery [Kas95] prefer the Bayesian Information Criterion (BIC) to the Akaike Information Criterion (AIC) because it approximates the Bayes Factor. Therefore, our analyses also rely on the guidance provided by the BIC. It is defined by

$$BIC = -2 \log \mathcal{L}_{MIX} + \log \left(N \times \#params \right), \tag{2.7}$$

with $\mathcal{L}_{MIX}$ the Gaussian-mixture model likelihood, N the number of observations and $\#params$ the number of parameters of the model.

2.3.2 Valid model selection

In our data mining scenario, we found it inappropriate to conduct model selection on the basis of a single BIC value because it left several questions unanswered. We give some of them:

1. What is the statistical significance of BIC scores differences that are less than 5%?
2. If EM was initialized from different starting values, how reliable would the cluster results be?

3. Did the EM-algorithm end in a local or a global likelihood maximum?

For this reason, we decided to further validate our choice for a particular model by repeating the data modeling process for different starting values; our approach proceeds as follows:

1. Set an integer that fixes the starting point of the random number generator.
2. Draw from a uniform distribution a matrix of cluster membership probabilities.
3. Proceed to a *maximization* step (M-step) to identify the parameters of the most likely model.
4. Start EM-algorithm from its *expectation* step (E-step).

This way, by repeating EM initialization from many different starting points, we can select the most likely model and consider it as the *optimal* one.

Then, to conduct a valid model selection, we aggregate the BIC scores in a number of ways. In first place, we report the average rank of the model (respectively the average rank of the number of clusters) when a particular number of clusters (resp. a model-type) is chosen. These rankings may enable to select for a particular type of model and a number of clusters. We also report tables that characterize statistically the BIC scores in terms of the empirical *mean*, the standard deviation and different quantile statistics. Finally, two more tables present the starting values and the BIC scores of the most likely models for each combination.

2.4 Characterizing, comparing and evaluating cluster results

Because cluster models may take different spatial-shapes, we need methods to report their characteristics and to compare them. Further, when analysing data from the medical domain, we consider as important to evaluate the clinical relevance of the subtypes by some additional characteristics. Therefore, in this section, we present our techniques to address these different aspects.

2.4.1 Visualizing subtypes

To check the effect of changing the settings (the type of cluster model and the number of clusters), we need visualization tools to see the characteristics of the cluster results. Being influenced by Tukey [Tuk77] and Tufte [Tuf83; Tuf90] for scientific data visualization and by Brewer's suggestions for color selection in geography [Bre94], we selected three visual-aids to address this issue: *heatmaps* [Eis98], *parallel coordinates* plots [Ins85] and *dendrograms* [Sne73].

Heatmaps In the analysis of micro array data, heatmaps are often used to display and cluster data. However, as heatmaps depend on hierarchical clustering, there are many parameters that need to be set rather subjectively. Besides, as we do calculations with distance measures, the variables should be scale-free and comparable; this may be awkward when variables are not scale-homogeneous. On top of that, as variables are correlated, the distances will mostly reveal patterns in the principal component dimensions of the data.

For the OA data, we can illustrate this by considering a large joint factor that consists of hips and knees and another one that consists of the spine joints. Simply because there are only four variables in the first factor and about 20 in the second, the spine has a larger "contribution" than the large joints in the distance. So, simple distances lack sensitivity to manifest changes in the small principal component dimensions. We limit the use of heatmaps to report statistical patterns of the clusters, e.g. the mean, the median or quantiles.

Next, as hip left and right pertain to the hips in OA or as both urinary and cardiovascular problems reflect autonomic symptoms in PD, we can often group variables into main factors. Indeed, we may expect the variables to correlate in each factor; yet, standard heatmaps do not exploit the grouping of the variables, this makes the comprehesion of the cluster results more difficult.

Parallel coordinate plots In parallel coordinates plots, we can make use of this grouping information in factors to order the variables appropriately. For each cluster, we use a different color and, as Figure 2.2 illustrates OA data, we characterize each center (μ_k) by lines connecting the different variables (the parallel axis). In this Figure, we notice the particular ordering for the cervical and the lumbar spinal joints that reflects the natural ordering of these joints from top to bottom. An interesting additional property of this type of plots is that besides each cluster center (the mean pattern), we can also report quantile-statistics using connected lines of a different shape (e.g. the 2.5% and 97.5% patterns of a cluster).

Dendrograms Finally, in spite of the many disadvantages of hierarchical clustering, we find it a useful addition to the heatmaps and parallel coordinates because dendrograms can illustrate the similarity between the center patterns or between the variables. In fact, a dendrogram on the cluster centers can help to order the clusters by similarity, whereas a dendrogram on the variables can provide a rudimentary factor analysis. Therefore, both kinds of dendrograms are included and provide additional understanding.

2.4.2 Statistical characterization and comparison of subtypes

First, using the *log of the odds*, we report the main statistical characteristics of the clusters. Second, to cross-compare the cluster results, we rely on regular

association tables from which we estimate the usual χ^2 statistics. Next, we use further the χ^2 statistics to calculate a single measure in terms of the *Cramer's V* coefficient of nominal association. Finally, as a way to assess the *reproducibility* of cluster results, we estimate the generalization of the cluster result by training common machine learning algorithms on the clustered data.

Statistical characterization For each application domain, we group variables by main factor such as the main joint sites in OA (the spine facets, the spine lumbars, the hips, the knees, the distal and the proximal interphalengeal joints), the impairment domain in PD (the cognitive, the motricity and the autonomic disorders) and the class of molecular descriptors in drug discovery. Then, to characterize statistically the cluster results, we compute the *odd* of the cluster data distribution as compared to the one of the dataset; the data distribution is the sum of the scores in each group of variables (the factors).

In practice, one might refer to the *log of the odds* as the cross-product because we calculate it from tables similar to Table 2.1. We express the log of the odds of a cluster k on a factor l as

$$logodds_{kl} = \log \frac{A \times D}{B \times C}. \tag{2.8}$$

Table 2.1: *For each sum score l, we consider a middle value δ_l such as the dataset mean or median. For cells A and B, we use it to count how many observations i in the cluster S_k have a sum score above and below its value. For cells C and D, we proceed to a similar count but on the rest of the observations $i \in \{S - S_k\}$.*

	$x_i < \delta_l$	$x_i \geq \delta_l$
$i \in S_k$	A	B
$i \in \{S - S_k\}$	C	D

Statistical comparison of cluster results In order to compare cluster results, we report association tables that describe the joint distribution between the two cluster affectations of the observations (nominal variables). If the table has many empty cells, then the two cluster results are highly related. However, if the joint distribution over all cells is even, then the two cluster results are unrelated (independent).

Further, to summarize the association tables, we calculate the Cramer's V. Similarly to Pearson's correlation coefficient, the Cramer's V takes values in $[0, 1]$; one stands for completely correlated variables and zero for stochastically independent ones. The measure is symmetric and it is based on the χ^2 statistics of

nominal association. Therefore, the more unequal the marginals, the more V will be less than one. Alternatively, the measure can be regarded as a percentage of the maximum possible variation between two variables. It is defined by

$$V = \sqrt{\frac{\chi^2}{n \times m}}, \tag{2.9}$$

where n is the sample size and $m = min(rows, columns) - 1$.

In our table-charts, we will embed in the top left the joint distribution and in the lowest row the *Cramer's V* coefficient.

Estimating the cluster result reproducibility When performing unsupervised cluster analysis, it is important to know whether the cluster result generalizes, for instance to the total patient population in the case of medical research. Therefore, we chose to assess the cluster result *learnability* by training machine learning algorithms like the naive Bayes, the linear Support Vector Machines or, as a baseline, the one nearest neighbor classifier.

To evaluate these algorithms, we use the average classifier accuracy estimated by training ten times the classifiers on datasets splitted randomly into training (70%) and test set (30%). To split the data, we chose to preserve in every training and test set the cluster proportions from the original sample.

Stratifying the samples enables to reduce the variability of the accuracy estimates which is coherent with the practice in machine learning because we primarily aim to compare algorithms. However, in medical research, we might prefer to include the variability inherent to the cluster proportions in the estimation of the accuracy.

2.4.3 Statistical evaluation of subtypes

When conducting a subtype discovery analysis, a key concern is the evaluation of the clusters. For that purpose, we implemented a simple mechanism to add study-specific evaluation procedures of the clusters.

In OA for instance, as the study involves sibling pairs, we defined two statistical tests that assess the level of familial aggregation in each subtype and its significance. Our first test relies on a risk ratio which we refer to as the λ_{sibs}, whereas the second test makes use of a χ^2-test of goodness of fit.

In drug discovery, χ^2 cell-statistics between the human-defined classification and the one identified by the subtyping are reported; we search for χ^2 cell-statistics showing a large marginal.

The λ_{sibs} risk ratio in OA research First of all, we characterize each individual as *proband* or *sibling* depending on whether this individual was the first sibling involved in the study or not.

Then, this test quantifies the *risk* increases of the second sibling given the characteristics of the proband. For instance, a $\lambda_{sibs} = 1$ means that the risk does not increase and that the cluster membership of the proband does not influence the one of his sibling. On the other hand, if $\lambda_{sibs} = 2$, then the risk increase is two-fold. Finally, a λ_{sibs} is significant when the lower bound of the 95% confidence interval is above 1. In the following, we describe formally the λ_{sibs} and we derive its confidence interval analytically by the delta method.

Take two siblings s_1 and s_2 with s_1 being the proband. A proband is the first affected family member who calls for medical attention. We consider the probability of a sibling to belong to a group S_k as $P(s_i \in S_k)$ with $i \in \{1, 2\}$, or for short $P(s_i)$. Then, the conditional probability that the second sibling is in S_k given that the first sibling is also in S_k is referred to as $P(s_2|s_1)$. Therefore, the λ_{sibs} is expressed by

$$\lambda_{sibs}(S_k) = \frac{P(s_2|s_1)}{P(s_2)} = \frac{P(s_1, s_2)}{P(s_1)P(s_2)} = \frac{P(s_1, s_2)}{P(s)^2}. \tag{2.10}$$

where $P(s_1) = P(s_2) = P(s)$ if the population is considered to be infinite. Next, we derive a confidence interval by the delta method using

$$\lambda_{sibs} = \frac{\hat{\alpha}}{\hat{\beta}} \tag{2.11}$$

where $\hat{\alpha} = P(s_1, s_2)$, $\hat{\beta} = P(s)$ (the hat denotes quantities estimated from the data). Then, the variances and covariance of $\hat{\alpha}, \hat{\beta}$ have the form

$$\sigma_\alpha^2 = \frac{\hat{\alpha}(1 - \hat{\alpha})}{n_i}, \tag{2.12}$$

$$\sigma_\beta^2 = \frac{\hat{\beta}(1 - \hat{\beta})}{N}, \tag{2.13}$$

$$cov(\hat{\alpha}, \hat{\beta}) = \frac{\hat{\alpha}(1 - \hat{\beta})}{N}, \tag{2.14}$$

with n_i the sibship size and N the number of observations. The first order Taylor approximation of $f(\alpha, \beta)$ in $(\hat{\alpha}, \hat{\beta})$ is expressed by

$$f(\alpha, \beta) = f(\hat{\alpha}, \hat{\beta}) + \sum_{\delta=\alpha,\beta} (\delta - \hat{\delta}) \frac{\partial f(\hat{\alpha}, \hat{\beta})}{\partial \delta} + R_1. \tag{2.15}$$

If we move the zeroth derivative to the left and we raise everything to the square, then we obtain

$$\Big(f(\alpha, \beta) - f(\hat{\alpha}, \hat{\beta})\Big)^2 = \left((\alpha - \hat{\alpha}) \frac{\partial f(\hat{\alpha}, \hat{\beta})}{\partial \alpha} + (\beta - \hat{\beta}) \frac{\partial f(\hat{\alpha}, \hat{\beta})}{\partial \beta}\right)^2. \tag{2.16}$$

Provided that $\partial f(\hat{\alpha}, \hat{\beta})/\partial \alpha = 1/\hat{\beta}^2$ and $\partial f(\hat{\alpha}, \hat{\beta})/\beta = -2\hat{\alpha}/\hat{\beta}^3$, we obtain

$$\begin{aligned}\left(f(\alpha,\beta)-f(\hat{\alpha},\hat{\beta})\right)^2 = \quad & (\alpha-\hat{\alpha})^2\left(\tfrac{1}{\hat{\beta}^2}\right)^2 \\ & +(\beta-\hat{\beta})^2\left(\tfrac{-2\hat{\alpha}}{\hat{\beta}^3}\right)^2 \\ & +2(\alpha-\hat{\alpha})\left(\tfrac{1}{\hat{\beta}^2}\right)\left(\tfrac{-2\hat{\alpha}}{\hat{\beta}^3}\right).\end{aligned} \tag{2.17}$$

Finally, taking the expectation, the variance is expressed by

$$\sigma_\lambda^2 = \frac{1}{\hat{\beta}^4}\left(\sigma_\alpha^2 - 4cov(\hat{\alpha},\hat{\beta})\frac{\hat{\alpha}}{\hat{\beta}} + 4\sigma_\beta^2\frac{\hat{\alpha}^2}{\hat{\beta}}\right), \tag{2.18}$$

or equivalently

$$\sigma_\lambda^2 = \frac{1}{\hat{\beta}^4}\left(\sigma_\alpha^2 - 4cov(\hat{\alpha},\hat{\beta})\hat{\beta}\lambda + 4\sigma_\beta^2\lambda\right). \tag{2.19}$$

A χ^2-test of goodness of fit for OA research We also implemented a simple χ^2 test of goodness of fit to assess the level of familial aggregation in each cluster k.

This test counts the pairs of siblings in each group and compares them to the ones expected when cluster membership would be random. If we first define N as the number of individuals and if we let S be a random draw of size $|S|$, then the probability that an individual i belongs to S is

$$P(i \in S) = \frac{|S|}{N}. \tag{2.20}$$

Next, if we consider a second individual j which is independent of i, then the probability that both i and j belong to S is expressed by

$$P(i,j \in S) = P(i \in S)P(j \in S) = \left(\frac{|S|}{N}\right)^2. \tag{2.21}$$

Further, if we denote by $E(i,j \in S)$ the expected number of sibling pairs under random cluster membership which relies on the total number of pairs $(N/2)$, then

$$E(i,j \in S) = P(i,j \in S)^2\frac{N}{2}. \tag{2.22}$$

Finally, the Grand Total of the χ^2 test is

$$GrandTotal = \sum_{k=1}^{G}\frac{(O(i,j \in S_k) - E(i,j \in S_k))^2}{E(i,j \in S_k)} = \sum_{k=1}^{G}\chi_k^2, \tag{2.23}$$

where k indices over the different clusters and χ_k^2 refers to the separate χ^2 statistics of each cluster. The number of degrees of freedom of our test is

$$df = G - 1 \tag{2.24}$$

with G the number of clusters.

Association tables in drug discovery In order to better understand the relationship between the bioactivity classes, we decided to study the joint distribution between the subtypes and the bioactivity classes. For this purpose, the joint distribution between the cluster affectation and the bioactivity class is reported both in terms of cell-counts and χ^2 cell-statistics; we are interested in the cells with high χ^2-statistics.

2.5 Concluding remarks

We presented a data mining scenario that facilitates and enhances the search for subtypes with application to medical research and drug discovery. This scenario involves techniques to prepare data, a computational approach repeating data modeling to select for a number of clusters and a particular model, as well as other methods to characterize, compare and evaluate the most likely models. Therefore, our scenario does not solely cluster data but it also produces a set of results to conduct a subtype discovery analysis: from data preparation to subtype evaluation.

Chapter 3

Reliability of Cluster Results for Different Types of Time Adjustments

As age in Osteoarthritis (OA) and disease duration in Parkinson's disease (PD) are known to play a major role in the disease severity, not adjusting the data for their contribution would lead to subtypes essentially characterized by them. Yet, this adjustment can be done in a number of ways: depending on the variable, we can consider a linear, a logarithm and an exponential function for the age or the disease duration. As this choice may influence the result of a subtyping analysis, we consider two items: first, how to deal with the time dimension in the data and second, how reliable are the subtypes? In this chapter, we discuss these two issues and we propose a method to select the adjustment that will lead to the most reliable subtypes.

3.1 Introduction

In searching for disease subtypes by cluster analysis, we have to consider two key elements.

1. How to deal with the time dimension in the data?

2. How reliable are the cluster results?

First, adjusting for time helps reduce the variability in the data, hence increases the homogeneity of the model subtypes. However, no precise guidance exists that indicates whether we should reduce the variability according to, for example, a log, a square root, or simply a linear time effect and whether all variables should be adjusted for the same type of effect or not. Secondly, because we

expect to use these cluster models for clinical research, we want to assess the reliability of cluster results. In this chapter, we address these two issues by comparing different types of time-adjustments when altering the data by noise addition.

The outline of this chapter is as follows. We start by describing our method to assess the reliability of cluster results and then, we illustrate our results for OA and PD analyses.

3.2 Methods

In the following, we describe the sequence of steps to conduct our analysis: the data preparation, the noise-addition procedure, our mixture model, the *reliability* measure and finally our evaluation methodology.

Data and preparation For the analysis on PD data, we do not consider the longitudinal aspect of the data; we only analyse the 1152 complete profiles from the four years. For the analysis on OA data, we conducted experiments on the 422 complete profiles.

In the overal severity of OA and PD, respectively age or disease duration which we further refer to as the *time*, are known to play a major role. Therefore, to model for clusters without the *time* dimension, we first perform a regression on the *time* for each variable and then we conduct the cluster analysis on the residuals of the regression. Next, in order to manipulate scale invariant quantities, the residuals are further processed by computing their z-scores.

Simulating new data by noise addition To assess cluster result reliability, we add to the data a gausian noise $\epsilon_l \sim \mathcal{N}(0, \sigma_l)$. This way, we generate new datasets $\mathbf{Y}_l$ that are alterations of $\mathbf{X}$ with different amounts of noise:

$$\mathbf{Y}_l = \mathbf{X} + \epsilon_l, \tag{3.1}$$

where l indices over the "noise-widths" (σ_l) that are proportional (%) to the variance in $\mathbf{X}$; here, variances equal 1 because we take the z-score of variables. The proportions are of the form $1/2^q$, with $q = 1, ..., 10$ such that, in %,

$$\sigma_l \in \{50, 25, 12.5, 6.25, 3.12, 1.56, .78, .39, .19, .1\}. \tag{3.2}$$

Model types for cluster analysis We will present experimental results on models of type VVI for which we search five mixtures. Recall that VVI models estimate, for each mixture, the mean μ_k and the diagonal covariance matrix Σ_k with $k = 1, ..., 5$ (cf. Chapter 2).

Cluster results reliability We repeat cluster modeling on slightly differing datasets $\mathbf{Y}_l^s$ where l indicates the noise level σ_l and s is a random seed in $1, ..., 10$ that fixes the drawing process. Then, given these cluster results, we measure their two-by-two association by means of the Cramer's V, which leads for each σ_l to $10 \times (10-1)/2 = 45$ measures.

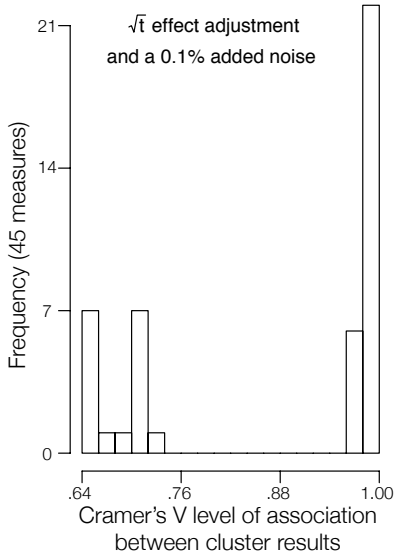

Figure 3.1: *Measures of association V do not distribute normally; therefore, summarizing by the empirical mean would not be reliable. Instead, we prefer to report the quantiles of V as illustrated in Figures 3.2 and 3.3.*

Evaluation methodology Denote by α and β the estimated intercept and coefficient vectors of the regression, by the matrix $\mathbf{X}$ the data where x_{ij} refers to measurement j of observation i, then the regression is given by

$$x_{ij}(t_i) = \alpha_j + \beta_j g(t_i) + \varepsilon_{ij}, \tag{3.3}$$

$$\varepsilon_{ij} = x_{ij}(t_i) - \alpha_j - \beta_j g(t_i). \tag{3.4}$$

The ε_{ij} refer to the residual variation and $g(t) \in \left\{log(t), \sqrt{t}, t, t^2, exp(t)\right\}$ (the time effect is not necessarily linear). We performed experiments on four different types of time-adjustments:

(a) $\varepsilon_{ij} + \epsilon_l = \epsilon_l + x_{ij}(t_i) - \alpha_j - \beta_j log(t_i)$,

(b) $\varepsilon_{ij} + \epsilon_l = \epsilon_l + x_{ij}(t_i) - \alpha_j - \beta_j\sqrt{t_i}$,

(c) $\varepsilon_{ij} + \epsilon_l = \epsilon_l + x_{ij}(t_i) - \alpha_j - \beta_j t_i$ and

(d) $\varepsilon_{ij} + \epsilon_l = \epsilon_l + max_{r^2;g}\{x_{ij}(t_i) - \alpha_j - \beta_j g(t_i)\}$.

In words, (a), (b) and (c) apply to all variables j the same type of time adjustment, i.e. either a log, a square root or a linear adjustment while (d) selects for each variable the adjustment that maximizes the variability explained by the linear regression (the r^2). Then, with respect to the cluster results *reliability*, we vary the noise levels in the data from minor levels σ_l like .1, .2 or 1.6% to substantial ones like 25 or 50%. For each σ_l, we measured the association levels of every pair of cluster results by the Cramer's V measure.

In Figure 3.1, we report a sample distribution of V when comparing cluster results on PD data adjusted for a square root time effect and altered by a .1% noise. Remarkably, the V values aggregate at levels .7 and 1. This particular distribution may indicate that EM stops its iterative process in equally likely (1) but substantially different end points (.7). However, because measurements of V are non-normally distributed, summarizing measurements by an average is meaningless. Therefore, we prefer to compare the cluster results by visualizing the quantiles of V.

In the color images of Figures 3.2 and 3.3, we illustrate side-by-side the quantiles for different noise levels σ_l. The color mapping is black when cluster results associate greatly (1) and white when they compare only fairly (.5). In particular, the Figure 3.2 exhibits narrow contour levels between the lines .75 and 1, which illustrates again the non-normality of the measurements V. More generally, in the four cases, the association levels V decrease when the noise levels σ_l increase. This is expected as for larger amounts of noise, the mixture modeling integrates the additional noise in the models.

3.3 Experimental results

First, comparing the results in Figure 3.2 (PD data), we notice that (b) presents overal high association levels V (black) whereas (c) presents the lowest ones. Then, concerning (a) and (d), (a) seems to show higher association levels than (d), but consistently lower than (b).

As a result, when ranking the adjustments for PD, we obtain the following ranking:

1. the square root (b),
2. the log (a),
3. the r^2-optimizing (d),
4. the linear-type (c).

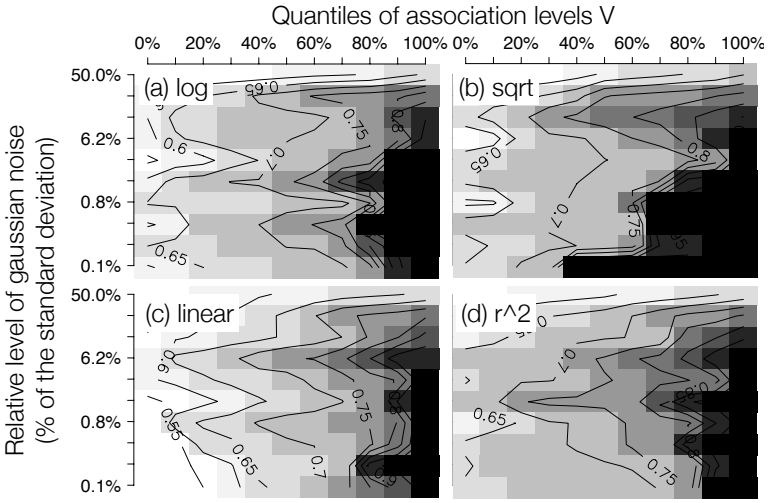

Figure 3.2: *For the PD dataset, the figure illustrates the quantiles of V (x-axis) for different relative levels of added noise σ_l (y-axis). The more dark the subfigure, the more the cluster results are highly related according to V; therefore, (b) shows better comparability than (a), (d) and (c).*

In fact, the square root and the log behave analogously as they both give more influence to the initial time values than to the large ones; the log can reduce more large values than the square root. Given that the markers are monitoring the activity of the disease and its level of severity, a linear relationship between the time and the markers would suggest severities that always increase. Yet, we may expect the severity to reach a maximum after a certain time, this would favour time effects of type square root or log.

In Figure 3.3 we report experimental results on OA data, which are in overal similar to those for the PD data. Still, we notice that the reliability results for OA data, are substantially less sensitive to noise addition than for PD data; Figure 3.3 shows larger black areas than Figure 3.2. Furthermore, for OA, the values of V drop as noise exceeds widths of 1.6%, whereas for PD, the levels of V are contrasted for noise widths that are higher than .1% because of the equally likely but different cluster results. Finally, although quite similar to PD, the ranking of the adjustments differs slightly; we obtain:

1. the log transformation (a),
2. the linear transformation (c),
3. the square root transformation (b) and finally,

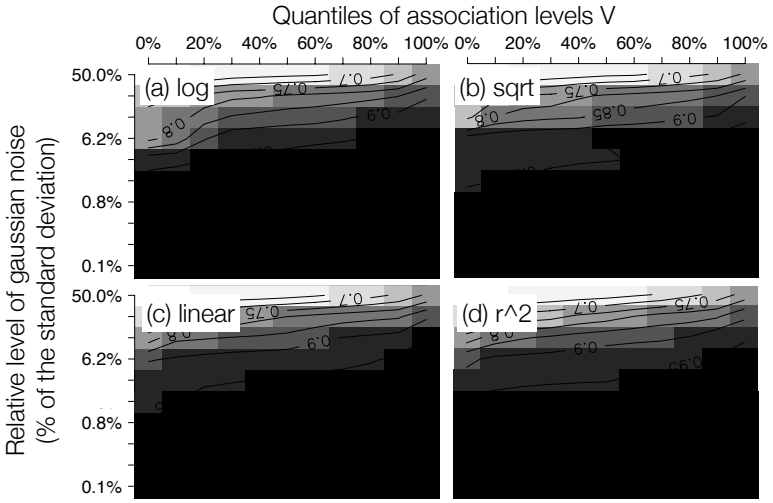

Figure 3.3: *For the OA dataset, the figure illustrates the quantiles of V (x-axis) for different relative levels of added noise σ_l (y-axis). The more dark the subfigure, the more there are of highly related clusterings according to V; therefore, (a) shows better comparability than (c), (b) and (d).*

4. the r^2-optimizing (d).

In fact the dataset properties of OA and PD differ substantially. For OA, the phenotype description of the participants has scores with values in $\{0, 1, 2, 3, 4\}$, while in PD the scales assessing the severity profile of participants are mixed. As a result, it is very likely that the scale-sensibility, and therefore the numerical complexity of the optimization (via EM-algorithm), explains most of the difference in terms of *reliability*.

In addition, as we clustered on PD data, the analyses lead to equally likely clustering end-points, as the V distribution illustrated particularly well in Figure 3.1. Yet, association levels exceeded .65 or .70, which means that the cluster results compare fairly well. Therefore, it seems that only a small subset of points is switching from a mixture to another between the different cluster results and gives rise to the drop in comparability V of the cluster results.

3.4 Why does optimizing the r^2 not boost the cluster *reliability*?

With respect to the r^2-optimization (d), as the regression fit improves (r^2), more variability is explained. Therefore, we would expect that the cluster results are

more *reliable*. In practice, this is not the case and cluster results are substantially less comparable.

To understand why the r^2-optimization is not an improvement, we first describe the procedure in more detail:

1. Do the time adjustments using the transformation that optimize the r^2.
2. Take the z-score.
3. Add noise by generating ten altered datasets.

Consequently, the r^2-optimization is performed on the same data meaning that the type of effect is uniquely selected for each variable. Therefore, we can not account the lower reliability to the noise addition procedure in the data.

When looking specifically to the type of effect for each variable that would exhibit the best fit, most types would be defined as square root or log; yet, for at least one variable, we noticed that the exponential was selected. As mentioned before, this seems particularly unlikely because we do not expect outcomes measuring disease severity to follow an exponential-like *time* effect but rather a log-, square root- or eventually linear-one.

In practice, not all variables are *time* dependent. Therefore, the procedure may have elected an effect-type with unsignificant r^2 differences. To tackle this issue, coefficients of the regression should be monitored for significance.

3.5 Concluding remarks

To prevent cluster analyses that model only the *time* dimension in the data, we presented a method that helps to select for a type of *time* adjustment by assessing the cluster results *reliability*.

Our method repeatedly clusters data to which a Gaussian noise is added. Next, to assess how cluster results compare, we use a χ^2-based measure of nominal association in terms of the Cramer's V.

Our results show that for OA and PD data, the sensibility of cluster results to noise addition depends on the type of effect chosen for adjustment. Next, searching for *reliable* cluster results, the best type of adjustment (in the set of possibilities we considered) is a square root of the disease duration for PD and a logarithm of the age for OA.

Chapter 4

Subtyping in Osteoarthritis, Parkinson's disease and Drug Discovery

We present subtyping results obtained using our data mining scenario. For Osteoarthritis (OA), we describe a sequence of steps that may enable to discover more homogeneous OA subtypes. In Parkinson's disease (PD) research, we did several analyses; we subtyped all available outcomes of PD severity, on motor disturbance outcomes and on the progression profile of PD patients. Finally, in the field of drug discovery, we looked for subtypes in a chemoinformatics dataset.

4.1 Introduction

In three sections, we present the result of subtyping analyses performed in medical research on Osteoarthritis (OA) and Parkinson's disease (PD) and in drug discovery.

For each subtyping analysis, we will recall briefly the data, give an outline of the analysis, motivate our choice for a subset of models and finally, characterize the subtypes of the most likely models. In OA and PD, these subtypes were further evaluated statistically: as part of our analysis, or through a *post-hoc* analysis, respectively.

4.2 Subtyping in Osteoarthritis

In this section, we report steps that may enable to subtype OA. We conducted our analyses on a dataset where the OA phenotype of patients is expressed in terms of Kelgren and Lawrence (K/L) scores [Kel57] that were determined from radiographic images (ROA) on 45 joint locations of the body. As some individuals had an incomplete ROA phenotype, they were discarded and we also decided to restrict our analysis to family sibship involving only two members (proband / sibling), we left out a total of 13 individuals. Therefore, for the analysis presented in this thesis, we analysed the ROA profile of 211 sibling pairs ($N = 422$ patients).

4.2.1 Outline of the analysis

We carry out the analysis on data where the severity of OA is described in terms of 45 ROA K/L scores, for details see Table 4.1. We prepared the data by standardizing each variable and by removing the variation due to the age as a linear effect.

Table 4.1: *Listing of the 45 joint locations where the individuals were measured.*

Main site	Joint location
Hips	Left and Right
Knees	Left and Right
Hands DIP+IP	Thumb (IP) and DIP 2, 3, 4, 5 on the Left and Right
Hands PIP	PIP 2, 3, 4, 5 on the Left and Right
Spine Discus	Cervical 23, 34, 45, 56, 67 and Lumbar 12, 23, 34, 45, 56
Spine Facets	Cervical 12, 23, 34, 45, 56, 67 and Lumbar 12, 23, 34, 45, 56

Next, we searched the data for clusters using model based clustering and we repeated this modeling 100 times given 100 different random starts. Given the small number of patients in the dataset, we limited the modeling to three, four or five clusters and models of type EII, VII, EEI, VEI or VVI (cf. Chapter 2).

Restricting our subsequent analyses to the optimal models in terms of BIC scores, on each subgroup of joint locations, we characterized the subtypes visually using parallel coordinates and statistically using the log of the odds.

Then, the clinical relevance of the subtypes may be evaluated using the λ_{sibs} risk ratio or a χ^2 test of goodness of fit that our scenario implements, but other perspectives could also be considered as, e.g. the number of joints affected or the mean body mass index of each subtype. The agreement between different types of models (consistency) was also assessed in terms of the Cramer's V coefficient of nominal association.

4.2.2 Model selection

As reported in Table 4.2, the most likely cluster result occurs for five clusters and the model VVI (1.1%). Yet, (VVI,4), (VVI,3) and (EVI,5) exhibit relative BIC score differences that are in average less than 5% lower than the best one for each random start (respectively 1.9%, 3.2% and 5%).

Table 4.2: *This table reports the average of the relative BIC score difference when comparing the scores with the best one for each random start. (VVI,5) is the most likely combination while (VVI,4), (VVI,3) and (EVI,5) exhibit a relative BIC score difference that is in average less than 5% lower than the best one.*

	EEI	EII	EVI	VEI	VII	VVI
3	9.1	9.6	6.4	6.3	7.5	3.2
4	8.8	9.4	5.4	6.0	7.3	1.9
5	8.6	9.3	5.0	5.6	7.2	1.1

As can be seen from Table 4.3 that describes the number of parameters for each model, there is a gain in BIC (model likelihood) for the model (VVI,5) compared to the models (VVI,4), (VVI,3) and (EVI,5), at the expense of more parameters. This was expected because we are in a trade-off situation between the number of parameters and the model fit.

Table 4.3: *Comparison of the number of parameters for the different types of gaussian mixtures having four clusters.*

Model	Number of parameters
VVI,5	$5 \times 45 + 5 \times 45 = 450$
VVI,4	$4 \times 45 + 4 \times 45 = 360$
VVI,3	$3 \times 45 + 3 \times 45 = 270$
EVI,5	$5 \times 45 + 1 + 5 \times (45 - 1) = 446$

Given the BIC scores, we would select (VVI,5) because, consistently, it exhibits higher BIC scores on the 100 repeats. However, because small BIC score differences may not be significant, we also decided to select those models whose average relative BIC score difference is less than 5% worst than the best one. Hence, we also considered the models (VVI,4), (VVI,3) and (EVI,5) for the subsequent analyses.

Finally, by repeating the cluster analysis 100 times on 100 random starts, we also know the starting points that yield the most likely models for each combination of number of clusters and model type. This enables to circumvent the issue of local optima when maximizing the model likelihood by EM-algorithm.

As described in Table 4.4, the best random starts are 6024, 6091, 6082 and 6023 for the models (VVI,5), (VVI,4), (VVI,3) and (EVI,5).

Table 4.4: *This table reports the random start that leads to the most likely model for each combination of number of cluster and model type. Model (VVI,5) gave the highest BIC score for the start initialized by a random seed set to 6024.*

	EEI	EII	EVI	VEI	VII	VVI
3	6059	6062	6045	6092	6013	6082
4	6097	6097	6058	6023	6075	6091
5	6072	6092	6023	6053	6094	6024

4.2.3 Subtype characteristics and evaluation

Here, as the OA subtypes are not yet published, we restrict ourselves in the analyses in Table 4.5 and Figure 4.1 to one joint location (the spine facet SF) which provides sufficient details for the purposes of this thesis.

Table 4.5: *This table describes in its top left area the joint distribution between the optimal cluster models (VVI, 5, 6024) and (VVI, 4, 6091). Each subtype is characterized statistically.*

	2	4	1	3	SF	λ_{sibs}	(95%)
3	100		13		- .4	1.3	(.5)
4	23		8	30	- .2	1.1	(1.0)
5	44	1	3	52	.5	1.4	(.7)
1	2		70		.9	2.3*	(1.2)
2	4	72			- .8	2.5*	(1.1)
SF (*log of the odds*)	- .6	- .9	1.0	.5	$p_{\chi^2} = 5 \times 10^{-4}$		
λ_{sibs} (95%)	1.2 (.3)	2.7* (1.2)	2.1* (.8)	1.5 (.8)	$(\Sigma_k \chi^2_k = 789.7)$ $V = 79\%$		

In first place, when we compare models (VVI,5) and (EVI,5) having the same number of subtypes, we observe very similar visual characteristics of the subtypes using both the parallel coordinates plots and heatmaps. Though not showing the complete subtype characteristics, a sample visual extract of the result is given in Figure 4.1 for (VVI,5). This figure shows that applying different models (here VVI and EVI) results in similar characteristics of the subtypes; this similarity can also be seen on other joint locations. Therefore, our subtyping scenario seems to show consistent results on the OA data.

To compare models with different number of subtypes, it is convenient to use tables like Table 4.5 which reports the joint distribution of the cluster allocation

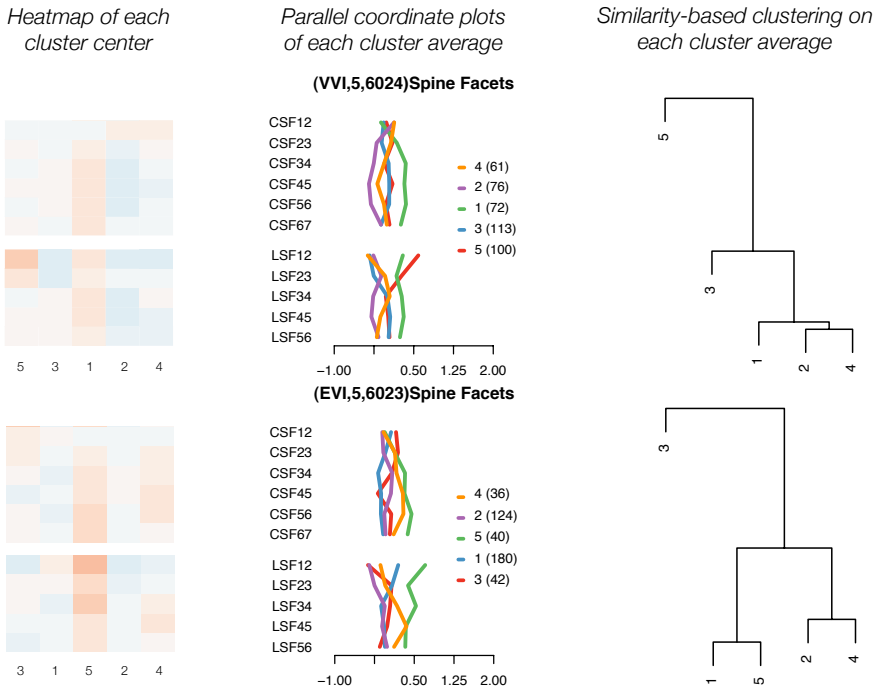

Figure 4.1: *Heatmaps, parallel coordinates and dendrograms are used to compare visually the characteristics of the optimal models (VVI, 5, 6024) and (EVI, 5, 6023). Because of the confidentiality of the results, only the spine facets are shown.*

for the models (VVI,4,6091) and (VVI,5,6024). In that case, we remark that the distribution shows a very strong association given the very low p-value of the χ^2-test of association ($p_{\chi^2} = 5 \times 10^{-4}$). This association is also reflected by the Cramer's V coefficient of nominal association (agreement) that is relatively high $V = 79\%$.

Next, we observe that subtypes (1, 2) of (VVI,5) are modeled by subtypes (4, 1) of (VVI,4) since most of the patients (72 and 70) distribute jointly to those subtypes. Yet, though subtype (1) of (VVI,5) is the main contributor of subtype (2) in (VVI,4) with 100 patients, additional patients are joining the subtype (2) in (VVI,4) from (4) and (5) of (VVI,5). Therefore, four and five clusters solutions with the VVI model differentiate mostly on subtypes (4) and (5) from (VVI,5). This illustrates the validity of the results since most differences can be understood in terms of merging and splitting operations on the clusters.

Table 4.5 also characterizes each subtype of the spinal facet (SF) factor by the *log of the odds*. Individual scores are summed into the SF factor and the score distribution in each cluster is compared to the one of the population by the

odd-measure. In Table 4.5, the SF factor does not characterize specifically any of the subtypes; other factors do but they are not reported here.

Finally, when appropriate subtyping has been obtained, there may be numerous ways to further characterize the subtypes in order to boost follow-up research. Here, since the OA study consists of siblings and that its main goal is to assess the genetic factor, we consider a score (the λ_{sibs} risk ratio) to further address the characteristics of the subtypes. As reported in Table 4.5 for (VVI,5), subtypes (1, 2) present a high risk of familial aggregation of (2.3, 2.5); a sibling from one of those two groups will have a more than twice higher chance to share the same pattern of OA than his brother/sister (the proband) as compared to random expectation (the population distribution). In fact, we also observe that the λ_{sibs} characteristics of subtypes (4, 1) of (VVI,4) are similar to those of subtypes (1, 2) of (VVI,5); further, the joint distribution shows that these subtypes are involving essentially the same individuals.

4.3 Subtyping in Parkinson's disease

In this section, we report subtyping results of PD on measures of the disease severity. These results are submitted for publication [Roo08a]. Again, we conducted the analysis using our scenario.

Table 4.6: *List of the 13 scores of PD severity that are used in the subtyping analysis. The additional score* disease duration *is not included in the cluster analyses; it is used for the initial data processing.*

Slowness of movement
Stiffness
Trembling
Rise, postural instability, gait
Freezing, speech, swallowing
Cognition sumscore
Autonomic sumscore
Motor fluctuations
Dyskinesias
Psychotic symptoms
Depression
Sleep
Daytime sleepiness
Disease duration

4.3.1 Outline of the analysis

First, we prepared the data by standardizing each variable and by removing the variation due to the disease duration. As described in Table 4.6, we selected 13 outcomes to model the severity of PD when performing the subtype discovery analysis. Next, we searched the data for clusters by model based clustering and we repeated the modeling 50 times given 50 different random starts.

Confining our subsequent analyses to the optimal models, we first characterized them visually (heatmaps) and then, statistically. Subsequently, subtypes were evaluated on the prior probabilities as well as on the agreement: between different types of models (for the consistency) and between year one and two (for the reproducibility).

4.3.2 Model selection

Given the small number of patients in the dataset, we limited the modeling to three, four or five clusters and models of type EII, VII, EEI, VEI or VVI. Then, as reported in Table 4.7, the most likely cluster result occured for five clusters and the model VVI. Yet, most other results gave relative BIC score differences that are in average less than 5% lower.

Table 4.7: *For year one, this table reports the average of the relative BIC score differences when comparing the scores with the best one for each random start. The model (VVI,5) is the most likely combination while (EII,4) has relative BIC scores that are in average for each random start, 3.63% lower than the best one.*

	EII	VII	EEI	VEI	VVI
3 clusters	3.86	2.79	4.05	2.75	0.99
4 clusters	3.63	2.49	4.25	2.48	0.21
5 clusters	3.56	2.63	3.75	2.46	0

Table 4.8: *Comparison of the total number of parameters for the different types of gaussian mixtures having four clusters.*

Model	Number of parameters
EII	$4 \times 13 = 52$
VII	$4 \times 13 + 4 = 56$
EEI	$4 \times 13 + 13 = 65$
VEI	$4 \times 13 + 4 + (13 - 1) = 68$
VVI	$4 \times 13 + 4 \times 13 = 104$

Further, as reported in Table 4.8, the model EII is the simplest in terms of number of parameters to estimate. On top of that, as Tables 4.9 and 4.10

illustrate, the agreement (reproducibility) was high for EII with four clusters. For these reasons, we decided to select model (EII,4).

Table 4.9: *Agreement between year one and year two of the four cluster solutions.*

Model	agreement (in %)	agreement (in %), clustering certainty above 95%
EII	66	86
VII	57	88
EEI	70	82
VEI	59	80
VVI	68	84

Table 4.10: *Agreement between models of the four cluster solutions (agreement when clustering certainty was* $> 95\%$*).*

	EII (in %)	VII (in %)	EEI (in %)	VEI (in %)
VII	58 (81)			
EEI	91 (100)	54 (75)		
VEI	61 (82)	78 (97)	57 (76)	
VVI	47 (60)	58 (75)	47 (59)	62 (73)

4.3.3 Subtype characteristics

We used heatmaps to visualize each subtype through its center obtained by taking averages in each dimension. We see that for four subtypes, the visual characteristics of the different models are very alike. Therefore, regardless of the model type, consistent results seem to emerge from the subtyping analyses. This is an additional reason why we decided to focus on model (EII, 4) which is the simplest of all those models; Figure 4.2 illustrates the heatmap of (EII,4).

We identify a subtype (1) mainly characterized by severe symptoms on most of the impairments. A second subtype (4) shows mild symptoms on all impairments. Another one (2) is especially characterized by high severity on the variables 1, 2, 3, 5, 6, 7 and 9 and low severity on the rest. A last subtype (3) displays intermediate severity on all variables.

4.3.4 Outline of the *post hoc*-analysis

Given the cluster results of the SubtypeDiscovery, we performed a discriminant analysis to evaluate which variables contributed to the cluster allocation. Next, we characterized the subtypes on demographics and disease related variables and

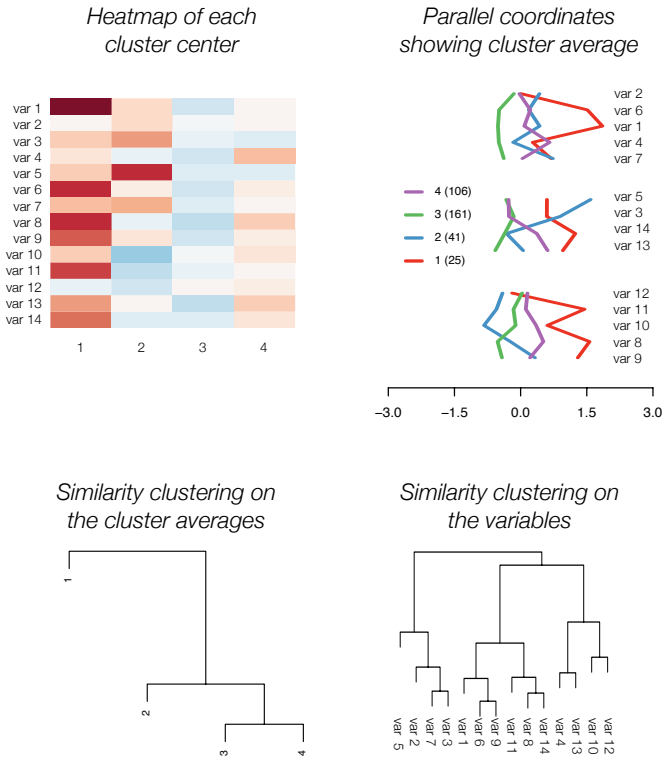

Figure 4.2: *Heatmap, parallel coordinates and dendrograms for the model (EII, 4, 6052). Because of the confidentiality of the results, names of variables are artificial names.*

we differentiated the subtypes using ANOVA, Kruskall-Wallis and χ^2 statistics. These analyses were performed in SPSS 14.0 (SPSS Inc., Chicago IL) [SPS05].

4.4 Subtyping in drug discovery

In this section, we present subtyping in drug discovery employing a public dataset. This dataset is called `wada2008` and it refers to the list of prohibited doping agents that the World Anti-Doping Agency maintains yearly.

Here, we perform our subtyping analyses on the 2008 version which lists $N = 3037$ different molecules. From this list, the molecular properties were determined on a selection of 98 descriptors, see Table 4.11. Next, we carried out the subtyping on the principal component dimensions of this dataset which explain 95% of the variability.

Table 4.11: *2D molecular properties that we selected to describe and characterize the databases of molecules.*

Atom and bond counts (ABC)	a_aro, a_count, a_heavy, a_IC, a_ICM, a_nB, a_nBr, a_nC, a_nCl, a_nF, a_nH, a_nI, a_nN, a_nO, a_nP, a_nS, b_1rotN, b_1rotR, b_ar, b_count, b_double, b_heavy, b_rotN, b_rotR, b_single, b_triple, chiral, chiral_u, lip_acc, lip_don, lip_druglike, lip_violation, nmol, opr_brigid, opr_leadlike, opr_nring, opr_nrot, opr_violation, rings, VAdjEq, VAdjMa, VDistEq, VDistMa
Adjacency and distance matrix descriptors (ADDM)	balabanJ, diameter, petitjean, petitjeanSC, radius, weinerPath, weinerPol
Kier and Hall connectivity and kappa shape indices (KH)	KierFlex, zagreb
Partial charge descriptors (PCD)	PC., PC..1, Q_PC., Q_PC..1, Q_RPC., Q_RPC..1, Q_VSA_FHYD, Q_VSA_FNEG, Q_VSA_FPNEG, Q_VSA_FPOL, Q_VSA_FPOS, Q_VSA_FPPOS, Q_VSA_HYD, Q_VSA_NEG, Q_VSA_PNEG, Q_VSA_POL, Q_VSA_POS, Q_VSA_PPOS, RPC., RPC..1
Pharmacophore feature descriptors (PFD)	a_acc, a_acid, a_base, a_don, a_hyd, vsa_acc, vsa_acid, vsa_base, vsa_don, vsa_hyd, vsa_other, vsa_pol
Physical properties (PP)	apol, bpol, density, FCharge, logP.o.w., logS, mr, reactive, SlogP, SMR, TPSA, vdw_area, vdw_vol, Weight

For this `wada2008` dataset, the intention is to describe additional outputs (not used in the other two applications) which our subtyping scenario can calculate in the course of an analysis. The outputs can contribute to the decision making process in terms of model selection and hence, of subtype discovery. In terms of drug discovery, it helps to understand the relationships between different chemical bioactivity classes.

4.4.1 Outline of the analysis

We performed our subtype analysis on 98 features that describe the properties of the prohibited molecules. These features are listed in Table 4.11 and further detailed in Tables that are given in Appendix A. In terms of data preparation, molecular descriptors were all standardized in order to remove the scale effect. Next, as the `wada2008` shows a relatively high number of dimensions (98) and because the descriptors are highly correlated, we performed a principal component analysis and the scores of each molecules on the main dimensions were extracted. We considered those dimensions that explained together 95% of the variability.

We searched the data for clusters using model based clustering and we repeated the modeling 50 times on 50 different random starts. We limited our cluster analyses to combinations of three, four, five and six clusters with models of type EII, EEI, VII, VEI, VVI. In this analysis, as the purpose of the `wada2008` dataset is to illustrate our subtyping scenario on a real example, we particularly focused on the additional outputs (not used in previous two applications) that can contribute to the model selection and the discovery of subtypes.

In that regard, besides the classical table aggregating the relative BIC scores, we also report several rankings to help the selection of a particular model and number of clusters. We will also illustrate the characteristics of the most likely subtypes using a heatmap, several parallel coordinate plots and a dendrogram. Finally, we report the main cross comparison table between the two most likely models.

4.4.2 Model selection

As reported in Table 4.12, the most likely cluster result occurs for six clusters and model VVI (3.4%). This number means that in average the BIC scores of model (VVI,6) are 3.4% lower than the best model: (VVI,6) initialized with a random seed of 6022 as reported in Table 4.13. Furthermore, this number also illustrates that the models (VVI,6) tend, in average, to be more likely in terms of BIC scores than the other combinations of model type and number of clusters. For instance, Table 4.12 shows that, as compared to the top-ranking model, the models (VVI,5) and (EII,6) are in average 7.7% and 84.6% less likely in terms of BIC score. In fact, as we usually consider the combinations of number of clusters and model type that are in average worst than 5% lower than the best one, here we do not consider any alternative model for further analysis.

Table 4.12: *This table reports the average of the relative BIC score difference when comparing the scores with the best one for each random start. (VVI,6) is the most likely combination. Other models have a BIC score that is in average more than* 5% *lower than the best one.*

	EII	EEI	VII	VEI	VVI
3	84.1	84.3	46.3	28.1	21.
4	84.3	84.5	41.1	23.1	12.
5	84.5	84.6	38.3	18.5	7.7
6	84.6	84.8	34.7	13.8	3.4

Table 4.13: *This table reports the most likely combination of model type, number of clusters and initialization.*

1	VVI	6	6022
2	VVI	6	6060
3	VVI	6	6033

Next, in Tables 4.14 and 4.15, we report the average rank of each model type or number of clusters as we fix respectively the number of clusters and the model-type. This way, we observe whether a particular model appears consistently as top-ranking for all numbers of clusters or whether some number of clusters shows as top-ranking when a particular model is chosen.

Table 4.14: *This table reports the average rank of each model type as the number of cluster is fixed. In that case, for three clusters, the most likely model type is in average VVI, then VEI, VII, EII and EEI.*

	3	4	5	6
EII	4	4	4	4
VII	3	3	3	3
EEI	5	5	5	5
VEI	2	2	2	2
VVI	1	1	1	1

In Table 4.15, we notice that depending on the model-type, the ranking of the number of cluster varies. Interestingly, models with equal variance across all clusters (EII and EEI) tend to favor a small number of clusters. On the other hand, those models estimating variance parameters particular to each model (VII, VEI, and VVI), tend to favor larger number of clusters. We do not yet have an interpretation of this result.

Table 4.15: *This table reports the average rank of each number of cluster as the model type is varied. In that case, for model type VII, the most likely number of cluster is in average six, then five, four and three.*

	EII	VII	EEI	VEI	VVI
3	1.	4.	1.	4.	4.
4	2.	2.8	2.	3.	3.
5	3.	2.1	3.	2.	2.
6	4.	1.1	4.	1.	1.

4.4.3 Subtype characteristics

To characterize the most likely subtyping results, we rely on both visualization and statistical measures. These characteristics are illustrated in Figure 4.3 and in Table 4.16.

In Figure 4.3, for both the heatmaps and the parallel coordinate plots, we first remark the subtype number four in blue of (VVI, 6, 6022) which shows a very high profile on most of the variables.

Second, the subtypes number six (red) and three (green) are characterized by a similar low profile on most of the variables, see both the heatmap (six and three) and the parallel coordinate plots (red and green). Yet, these two subtypes are different on `chiral_u`, `b_double` and on the `density`, see the parallel coordinate plots of the Atom and Bond Counts (ABC 1) factor and of the Physical Properties (PP) factor.

Third, there is the subtype one (orange) which is especially characterized by the variables `b_triple` of the factor Atom and Bond Counts (ABC 1) and `reactive` of the factor Physical Properties (PP). For the rest of the variables, this subtype exhibits an "average" profile. However, we also note that the modeling involves essentially the principal component dimensions 16 and 17, see parallel coordinate plots (PCA).

Finally, subtypes five (purple) and two (yellow) are made of a large number of molecules (1017) and (1051). However, when compared to subtypes one, three, four and six, they do not exhibit any particular characteristics. In fact, if we look at the parallel coordinates plot of the principal component analysis (PCA), we notice that these two subtypes are almost centered on all principal component dimensions whereas the other subtypes show some characteristics on at least one of the dimensions.

In fact, when looking at Table 4.16 row-wise for model (VVI, 6, 6022), we further notice that, between models (VVI, 6, 6022) and (VVI, 6, 6033) which are two top-ranking cluster results, the joint distribution differs particularly on subtypes two and five. Indeed, the molecules of these two subtypes make up most of the disagreement between the two results. On the other hand, subtypes one, three, four and six are discovered rather consistently by the two models since

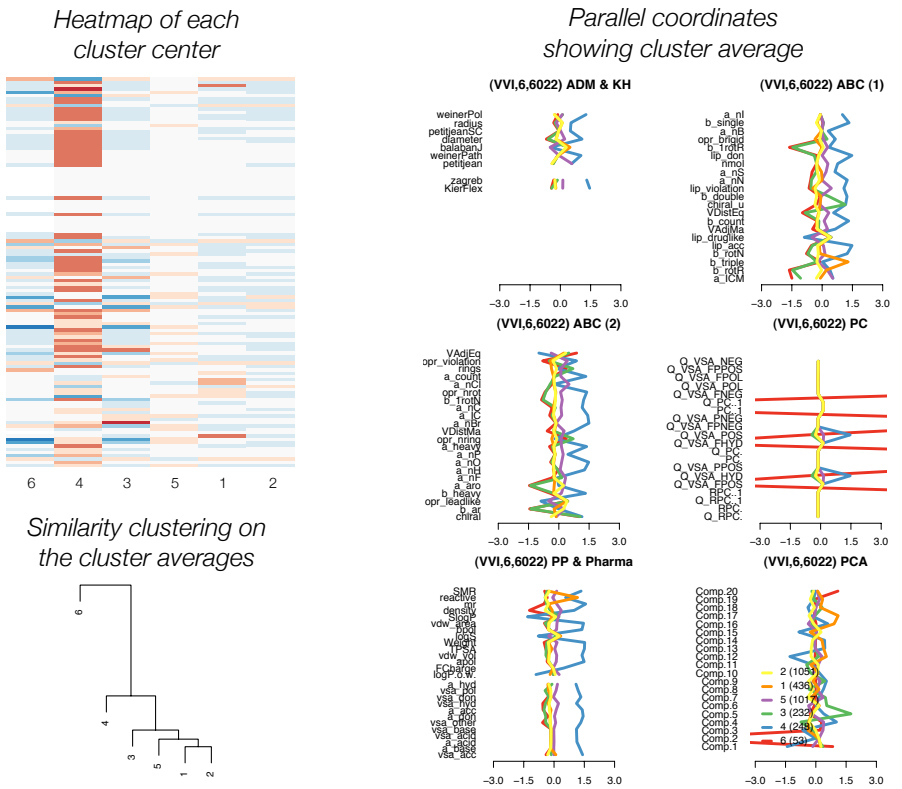

Figure 4.3: *This Figure illustrates the six average patterns of (VVI, 6, 6022). It also characterizes the different subtypes on all variables grouped by factors. The scales on the parallel coordinate plots refer to the z-scores with 95% of the values that should fit within* $[-2, 2]$. *In this Figure, the blue subtype with (248) molecules displays an especially high profile for most descriptors, the red (53) and green (232) subtypes show comparatively low profiles. In particular, these two subtypes differ on the partial charge descriptors (PC). We may account the "zigzag" of the red subtype to the numerical type of these variables which are counts.*

there is much less scattering. These results mean that subtypes two and five show poor homogeneity, whereas subtypes one, three, four and six show important and characteristic profiles. This uncertainty in the modeling is summarized in terms of the Cramer's V measure which shows a level of agreement of 76%.

The aim of the log of the odds computed on the sum score of each important factor is to summarize each subtype. However, in this domain, as the characteristics of the discovered subtypes one (orange), six (red) and three (green) are on a limited set of variables, making a sum score on a large number of variables hinders the characterization. In that case, only the subtype 2 (blue) shows high log of the odds on all factors because as mentioned previously, it exhibits generally high scores on all variables. Yet, our SubtypeDiscovery package (to be presented in the next chapter) can be configured to calculate as many summary statistics as necessary.

Table 4.16: *This table describes in its top left area the joint distribution between the optimal cluster models (VVI, 6, 6022) in rows and (VVI, 6, 6033) in columns. Each subtype is characterized by the log of the odds on the main factors: ABC, ADM, KH, PC, PP and Pharma, as well as on the principal components (here we only report 5 of them out of 20). Finally, in the lower right corner, we show the degree of agreement between the two cluster results (V) and the χ^2 statistics.*

	1	6	2	3	4	5	ABC	ADM	KH	PC	PP	Pharma
2	829	220	1	1			- .82	- .44	-1.26	-2.63	-1.44	-1.61
5	5	631	272	9	64	36	1.46	.84	1.68	.51	1.39	1.50
1			401	35			- .89	- .25	- .94	-3.28	- .01	- .84
3				232			-Inf	-3.76	-Inf	-Inf	-Inf	-Inf
4			7		193	48	1.60	.85	1.53	2.08	2.09	1.98
6						53	-Inf	-Inf	-Inf	Inf	-Inf	-Inf
ABC	-1.07	1.06	- .06	-4.03	1.35	.85						
ADM	- .56	.40	.33	-3.25	.39	1.25						
KH	-1.55	1.09	.00	-4.02	1.33	1.16						
PC	-3.19	- .38	- .94	-Inf	1.04	Inf						
PP	-1.68	.64	.50	-2.00	1.92	.93						
Pharma	-1.95	.72	.09	-Inf	1.78	1.51						
Comp.1	1.15	-1.05	.06	4.09	-1.04	-1.09						
Comp.2	Inf	3.09	Inf	- .52	-1.22	-6.67						
Comp.3	- .01	.69	- .27	-1.43	- .10	.45						
Comp.4	-Inf	1.39	- .02	-Inf	2.02	.50						
Comp.5	- .11	- .26	-1.04	Inf	.65	.02						

$$p_{\chi^2} = 5 \times 10^{-4}$$
$$(\Sigma_k \chi^2_k = 8682.7)$$
$$V = 76\%$$

4.5 Concluding remarks

We presented the results of subtype analyses in three different domains: in medical research on OA and PD and in drug discovery. For each domain, we first discussed the data used, then we gave an outline of the analysis, we motivated our selection for a small subset of models and finally, we characterized the subtypes of the most likely models. In OA and PD, these subtypes were further evaluated statistically: as part of our analysis, or through a *post-hoc* analysis.

For each application, we reported a selection of the output generated by our data mining scenario. In OA and PD, the model selection and the selection of graphics and table statistics were determined by the research team. In drug discovery, because they were not yet illustrated in previous two applications, we decided to show and interpret additional elements like tables ranking the model-type or the number of clusters.

To conclude, we showed in this chapter how our subtyping scenario could enhance the search for homogeneous subtypes in data. This data mining scenario repeats cluster modeling, reports visual characteristics and calculates a number of statistics on the subtypes. With each domain, based on the set of results generated, we also showed a slightly different way to conduct the statistical inference on the subtypes in data.

Chapter 5

Scenario Implementation as the R SubtypeDiscovery Package

To enable reproducibility *of our analyses and to abstract from the application domains, we implemented in the R SubtypeDiscovery package our data mining scenario to search for homogeneous subtypes in data by cluster analysis. We present the implementation in this chapter.*

5.1 Introduction

We previously introduced a *data mining scenario* to facilitate and enhance the task of discovering subtypes in data. For different application domains, we also presented some results of our subtyping analyses. In order to enable reproducibility of our analyses, we decided to make our scenario available as an R package. In this chapter, we present its implementation.

The R project for statistical computing is an initiative to provide as public domain software, an integrated suite of software facilities for data manipulation, calculation and graphical display [rla08]. R refers both to the computing environment and to the R language. In itself, R is very alike to the S environment and language. However, R is a public domain software under the GNU General Public Licence and it can be installed on a number of different operating systems such as Windows, MacOS and Unix.

Our data mining scenario consists of five sequential steps: the data preparation, the cluster modeling, the model selection, the characterization of the subtypes, their comparison and their evaluation. In Figure 5.1, we present the implementation of our scenario as the R SubtypeDiscovery package. It involves three classes: the dataset container (`cdata`), the cluster model (`cmodel`) and

the set of cluster results (`cresult`). This last structure stores the outcomes of the SubtypeDiscovery analyses: a dataset `cdata` that takes as input the data and some `settings` defining the way the data should be prepared and interpreted, along with the different models (`cmodel`) calculated while repeating the cluster analysis.

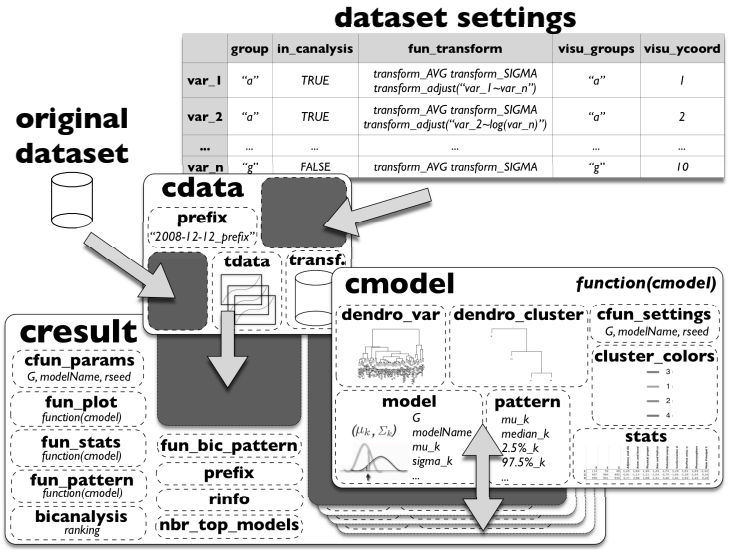

Figure 5.1: *We illustrate the main three classes of our package: the data* `cdata`*, the cluster model* `cmodel` *and the set of cluster results* `cresult`*. In particular, the dataset preparation (*`cdata`*) uses* `set_cdata()` *which takes as input the raw data and the* `settings` *that describe the data transformation.*

The outline of this chapter is as follows. We start by presenting the design of the implementation: the data preparation methods, the dataset class, the cluster result class, and the methods to characterize, compare and evaluate the cluster results. Next, we show two pieces of code that perform a typical SubtypeDiscovery analysis.

5.2 Design of the scenario implementation

We start by explaining the design of our implementation for preparing data. Then, we describe the constructor methods, their helper-methods, and the generic functions for both the `cdata` and the `cresult` classes. Finally, we discuss the methods to characterize, compare and evaluate the subtypes.

5.2.1 Methods for data preparation and data specific settings

As Figure 5.2 illustrates, the data preparation can be described in terms of five sequential steps. The first two steps aim to define a `settings` configuration file that tells how the data should be prepared; this is the stage of user-interaction. Next, as the `cdata` constructor is called (`set_cdata`), a method (`transform_cdata`) will parse the `fun_transform` column of the `settings` file. It enables to call the appropriate methods that will perform the data transformation. In fact, these transformations are defined in terms of two elements: the modeling procedure (e.g. the mean or the standard deviation) and its operating mode on the data (e.g. to substract or to divide). Therefore, there is a low level method (`transform_ALL`) that takes as parameter these two elements along with the data and it will process the data accordingly.

In the following, we describe the different methods.

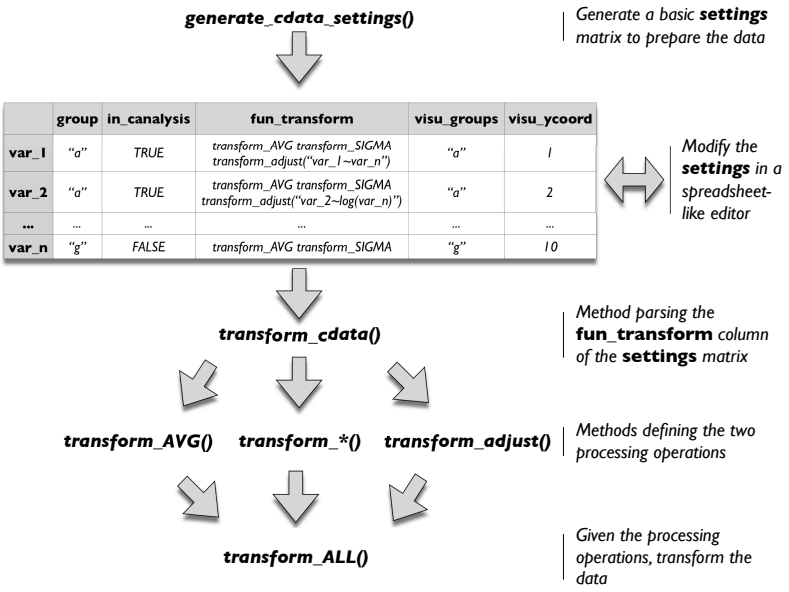

Figure 5.2: *Flow of operation to prepare the data.*

First, the user can rely on a helper-method to generate a `settings` matrix with default values (`generate_cdata_settings`). That matrix is necessary for a SubtypeDiscovery analysis because it is where the user describes how the data should be prepared, which variable should be involved in the cluster modeling (`in_canalysis`), how the variables should be represented graphically (e.g. in the parallel coordinate plots and the heatmaps) and characterized statistically (in the

log of the odds). Typically, the user will prepare the `settings` matrix within a spreadsheet editor. The `cdata` objects are datasets for SubtypeDiscovery analyses, they embed this configuration file within the data structure in the `settings` argument.

Upon construction of a dataset (`cdata`) for a SubtypeDiscovery analysis, the method `transform_cdata` is called. It parses the column `fun_transform` of the `settings` matrix sequentially from left to right and from top to bottom. Lower level methods are then called to process the data (e.g. `transform_AVG`, `transform_SIGMA`, etc.). The parsing uses commas and spaces first, and second, the parenthesis because the method `transform_adjust` can take as parameter a linear model formula in parenthesis. When data-preparing, we store in `tdata` structures the computed models and estimates. This enables to transform additional data given previous models and estimates.

To remove for each variable the variability explained by a given factor, e.g. the time, the method `transform_adjust` is used. It outputs a transformed vector and a `tdata` storing the regression model and the estimates. With a similar output, the methods `transform_ABSMAX`, `transform_L1`, `transform_L2`, `transform_MAX`, `transform_SIGMA` (respectively `transform_AVG`, `transform_MEDIAN`, `transform_MIN`) can normalize (resp. center) the values of the variables.

At the lowest level, the processing of the data is done by `transform_ALL`. Given two operations, it will either estimate the relevant statistics (e.g. the mean or the median) and use these to transform the data (e.g. by substracting or dividing), or use a previously computed estimate (in `tdata`) and apply it to the data. The method returns a transformed vector and a `tdata` structure where the estimators and the models are stored.

5.2.2 The dataset class (`cdata`) and its generic methods

In Figure 5.4, we illustrate the construction via the method `set_cdata` of a `cdata` structure that will contain the dataset of a SubtypeDiscovery analysis. First, the constructor copies the original dataset into `data_o`, second, it applies the initialization and filtering methods and next, it processes the data by the method `transform_cdata` given the `settings` matrix. In the following, we describe the default initialization procedure and the generic plotting method of a `cdata` structure.

In the initialization, the method `init_data_cc` is called. Via the column `in_canalysis` of the `settings` matrix, it limits the SubtypeDiscovery analysis to the observations showing complete records on the variables selected for the cluster modeling. This is because model based clustering can not process datasets having missing values. However, by subsetting on `in_canalysis`, we do not drop unnecessarily observations having missing values on variables not in the cluster analysis.

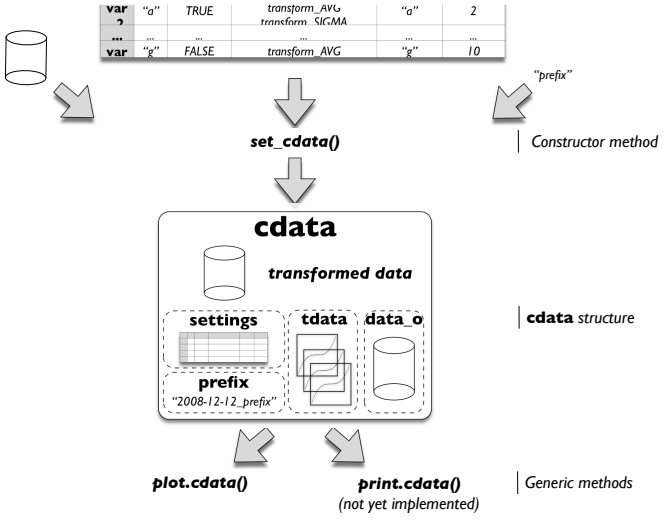

Figure 5.3: *A* `cdata` *stores information about the dataset for a subtype discovery analysis. This diagram illustrates the different methods to construct and access a* `cdata` *structure.*

Finally, the method `plot.cdata` reports boxplots and histograms for each of the variables of the dataset. Besides the histograms, additional information is reported as text over the estimators of the transformations (e.g. the mean or the standard deviation).

5.2.3 The cluster result class (`cresult`) and its generic methods

In Figure 5.1, we show a `cresult` data structure that can be constructed by the method `set_cresult`. It contains all the information of a SubtypeDiscovery analysis, i.e. the data (`cdata`), the experimental settings, all the models of the repeated cluster analyses (`cmodel`) and statistics such as the set of Bayesian Information Criterion (BIC) scores (`bicanalysis`). In the following paragraphs, we first describe the parameters of the constructor and then we discuss, together with their helper-methods, the generic methods associated to the `cresult` data structure.

First, the constructor takes as parameter a dataset (`cdata`), a cluster modeling method (`cfun`) that can be parameterized via (`cfun_params`), a set of methods to characterize and evaluate both graphically (`fun_plot`) and statistically (`fun_stats`) the subtypes, a number that indicates how many top-ranking

models we consider for the cross-comparison (`nbr_top_models`) and some methods (e.g. the mean or some other quantile statistics) to summarize the subtypes (`fun_pattern`) or the BIC scores (`fun_bic_pattern`).

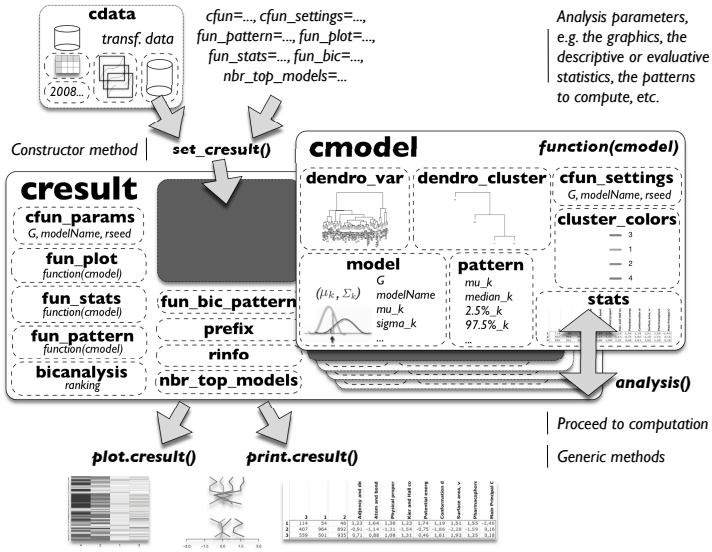

Figure 5.4: *A* `cresult` *stores all the data computed in the course of a Subtype Discovery analysis: the dataset* `cdata`*, the set of cluster models* `cmodel`*, the analysis settings, and some additional operating system/computing environment information* `rinfo`*. This diagram illustrates the different methods to construct and access a* `cresult` *structure.*

Next, we implemented a generic plotting method (`plot.cresult`) to report the visual characteristics of a `cresult` object. A first parameter (`device`) enables to define whether the graphical output should be redirected to a postscript file or a series of png pictures. A second parameter (`query`) can limit the plotting to one particular model among those listed by the command `names(a_cresult)`.

We also implemented the method `print.cresult` that can describe a `cresult` by a series of tables aggregating the BIC scores in a number of ways, e.g. by various rankings and some summary statistics such as the mean or a quantile statistics. It also reports tables where the top-ranking models are cross-compared.

Finally, the method `get_plot_fun` returns a function that, upon execution with a `cdata` argument will return another *unexecuted* plotting function. This new function takes a `cmodel` as parameter. In fact, by storing the unevalu-

ated plotting functions within the `cresult`, we can redraw independently the plots. Yet, in the course of a regular SubtypeDiscovery analysis via the method (`analysis`), all cluster models are by default graphically characterized.

5.2.4 Statistical methods to characterize, compare and evaluate subtypes

In the following, we present procedures that report statistical measures and summaries of cluster models. Some of these procedures are unevaluated function calls taking as argument a cluster model (`cmodel`); they are retrieved by the helper-method (`get_fun_stats`) when the constructor `set_cresult` is called. Then, we present additional methods that can calculate statistical patterns on the subtypes (`fun_pattern`) or the BIC scores (`fun_bic_pattern`). Last, we present the method that enables to cross-compare cluster results.

Statistical characterization of subtypes We first implemented `stats_logodds` that can calculate summary statistics of the subtypes based on an odd-ratio statistics. This method enables to identify the main characteristics of the subtypes on a number of factors; the factors are used to calculate sum-scores on groups of variables (`group`) defined by the user in the `settings` matrix. In practice, the odd ratios are calculated by comparing the distribution of the sum-scores in the cluster with the one in the whole dataset.

We also implemented a method (`stats_auuc`) to summarize the average level of uncertainty to cluster the observations for the current model. We refer to this average as the area under the curve of the clustering uncertainty (`auuc`).

Finally, in order to assess the reproducibility of the clustering result, we prepared a method (`stats_generalization`) to evaluate the classification accuracy of different machine learning algorithms trained on the clustered data. This procedure repeats the training of classifiers a number of times (by default 10) given a random training-test split (stratified) with 70% of the observations in the training set and 30% in the test set. At this moment, our package features three classification algorithms the naive Bayes, the k nearest neighbors classifier and the linear Support Vector Machines.

Statistical evaluation Because our research on OA involves a cohort study made of sibling pairs, we implemented two statistical tests that can assess the level of familial aggregation of the cluster models. One test quantifies the *risk* increases of the second sibling given the characteristics of the proband; it is referred to as the λ_{sibs} risk ratio (`stats_lambdasibs`). The other test counts the pairs of siblings in each cluster and it compares them to the the expected counts if observations were affected randomly to the clusters; the statistic is the one of a χ^2-test of goodness of fit.

In drug discovery research, we chose to report the joint distribution between the subtypes and the bioactivity classes in terms of cell counts. Second, in order

to illustrate how unequal the marginals of this distribution are, we report the χ^2 values (i.e. the deviation to the counts expected at random). We are especially interested in the cells that exhibit a high χ^2 value.

Statistical patterns To show the characteristic patterns of each subtype, we rely on summary statistics like the 2.5% quantile, the maximum, the mean, the median, the standard deviation and the 97.5% quantile of a numeric vector. These statistics are estimated by methods that discard by default the missing values (`patternLowquant`, `patternMax`, `patternMean`, `patternMedian`, `patternSd`, `patternUpquant`). When the constructor `set_cresult` is called, there are two parameters that define the different patterns to calculate: first, on each data subtype (`fun_pattern`) and second, on the set of BIC scores (`fun_bic_pattern`).

Cross-comparison of subtypes In the course of a SubtypeDiscovery analysis, we use the `compare_cresult` method to draw comparisons between the top-ranking cluster results. However, this method can also be called independently if two cluster models to compare are provided as argument. The function returns a list of tables where the models are cross-compared; there are two kinds of tables, those with the original values and those designed to be visualized (e.g. on a HTML report). The method can also store the original tables into comma separated value files.

5.2.5 Other methods

The method `analysis` defines a sample workflow for subtype discovery. It generates graphics on the data preparation, it performs the cluster analysis, it computes the subtype's characteristics and evaluate them, and it cross-compares the top-ranking results.

The function `fun_mbc_em` does a model based clustering on the data (a data matrix), given the model (`modelName`) and the number of clusters (`G`). The initialization is particular in that it draws at random a cluster membership probability matrix (`z`) of dimension ($N \times G$), with N the number of observations. Then, from that vector, it estimates the corresponding model by an M-step which is further used to start EM given the model.

5.3 Sample analyses

The purpose of this section is to show two fragments of code to give the reader an idea about the code needed for an analysis. For this purpose, we use a public chemoinformatics dataset that we embed in our package.

The outline of this section is as follows. First, we show the piece of code to conduct a typical SubtypeDiscovery analysis. Yet, as molecular descriptors tend

to correlate highly, we show a second analysis performed on the scores from the principal components; we select the dimensions that explain 95% of the variability.

5.3.1 Analysis on the original scores

In the following, we describe and then report a piece of code to perform a subtyping analysis on the `wada2008` dataset.

First, it is necessary to load the SubtypeDiscovery package via the method `library`. Then, we load the `wada2008` dataset and its predefined settings `wada2008_settings` which our package embeds. The next step is to prepare the dataset for a subtyping analysis (`cdata`). For this purpose, we use the constructor `set_cdata` that takes the dataset, the settings and a short name describing the data. In a similar way, we will prepare a `cresult` using the constructor `set_cresult`; many parameters are set by default. Finally, the subtyping analysis is carried out using the method `analysis`. It will repeat the cluster modeling, perform an analysis on the BIC scores, characterize graphically and statistically the cluster results and save the computed models.

```
library(SubtypeDiscovery)
# LOAD dataset
data(wada2008)
data(wada2008_settings)
# PREPARE CDATA
cdata1  <- set_cdata(data=wada2008
   , prefix="WADA2008_Sample_Analysis"
   , settings=wada2008_settings)
# PREPARE THE SET OF RESULTS FOR CLUSTER modeling
x <- set_cresult(cdata=cdata1,
   fun_stats=list(oddratios=get_fun_stats(
      fun_name="oddratios", fun_midthreshold=mean)),
   nbr_top_models=5,
   cfun=fun_mbc_em,
   cfun_settings=list(
      modelName=c("EII", "VII","EEI","VEI","VVI"),
      G=3:6,
      rseed=6013:6063))
# PROCEED TO THE ANALYSIS:
cresult_set <- analysis(x)
```

5.3.2 Analysis on the principal components

Here, we present a second subtyping analysis on the `wada2008` dataset. The difference with previous analysis is in the preparation of the data because here we decide to repeat the cluster modeling on the principal component dimensions. For

this purpose, we rely on the method `get_cdata_princomp` that will estimate the principal components of the dataset. This method will update the dataset structure `cdata1` such that the subtyping analysis is performed on the principal components of the data. The remainder of the code is the same than in previous analysis.

```
library(SubtypeDiscovery)
# LOAD dataset
data(wada2008)
data(wada2008_settings)
# PREPARE CDATA
cdata1 <- set_cdata(data=wada2008,
   prefix="WADA2008_Sample_Analysis",
   settings=wada2008_settings)
# PREPARE NEW CDATA FOR CANALYSIS ON PRINCOMP
cdata2 <- get_cdata_princomp(cdata1)
# PREPARE THE SET OF RESULTS FOR CLUSTER modeling
x <- set_cresult(cdata=cdata2,
   fun_stats=list(oddratios=get_fun_stats(
      fun_name="oddratios", fun_midthreshold=mean)),
   nbr_top_models=5,
   cfun=fun_mbc_em,
   cfun_settings=list(
      modelName=c("EII", "VII","EEI","VEI","VVI"),
      G=3:6,
      rseed=6013:6063),
   fun_pattern=list(mean=patternMean))
# PROCEED TO THE ANALYSIS:
x <- analysis(x)
```

5.4 Concluding remarks

We initially prototyped our subtyping methodology for OA in collaboration with the MOLecular EPIdemiology department (MOLEPI) of the Leiden University Medical Center. Then, following the interest of the Neurology department (LUMC) working on PD, we prepared it as the R SubtypeDiscovery package because this would enable reproducibility and reliability of our analyses. More recently, we collaborated with the Pharma-IT platform of the Leiden University to apply subtyping to the field of drug discovery. As our primary users are from biology, we simplified the scenario's design until it relied solely on a spreadsheet-like description of the data. This particular effort is to match the user's demand who, in the end, is expected to carry out his / her own analyses.

Yet, *usability* of our package could be further improved by constructing a basic graphical front-end. It would certainly reduce the amount of time needed to define

the dataset settings: by limiting the possibility for spelling mystakes, by making the software features more accessible, by removing the need to exchange between R and an editor, and by removing the manual initialization of the analysis via a R script file.

Besides, the software design could be more *robust* if we used the MLInterfaces package of BioConductor to determine generalization estimates of the classification algorithms on the cluster models. Similarly, using the Sweave package would enable us to separate the report-making process from the data generation. At this moment we still use our own Machine Learning and report-making methods.

Finally, a more extensive use of R object-oriented programming in our software design would further increase both the *usability*, *robustness* and *reliability*.

Our R package can be found at:

- `http://www.grano-salis.net/SubtypeDiscovery/`.

Part II

Automatic Text Classification

Chapter 6

A Scenario for the Comparison of Algorithms in Text Classification

In this chapter, we describe a data mining scenario for the comparison of algorithms in text classification. We start by introducing the problem of automatically classifying text documents into categories. Then, we consider the problem of doing fair *classifier's comparisons. Next, we describe the three algorithms that we compare: the k nearest neighbors classifier, naive Bayes and the Support Vector Machines. Last, we define the settings of our scenario and the data on which we performed our experiments on.*

6.1 Introduction

The aim of text categorization is to build systems which are able to automatically classify documents into categories.

To build text classification systems, the *bag of words* representation is the most often used feature space. Its popularity comes from its wide use in the field of information retrieval and from the simplicity of its implementation. Yet, as in the *bag of words* representation each dimension corresponds to the number of occurrences of the words in a document, the task of classifying text documents into categories is difficult because the size of the feature space is very high. In typical problems, it commonly exceeds tens of thousands of words. Another aspect that hampers this task is the fact that the number of training documents is several orders of magnitude smaller than the size of the feature space.

Among the algorithms applied to text classification, the most prominent one is *linear* Support Vector Machines. First introduced to text categorization by Joachims [Joa98], Support Vector Machines (SVM) were systematically included

in subsequent comparative studies [Dum98; Yan99b; Zha01; Yan03]. Their conclusions suggest that SVM is an outstanding method for text classification. In his large scale study, Forman also confirmed SVM as outperforming other techniques for text classification [For03].

So should we just not bother about other classification algorithms and opt always for SVM?

As shown in [Dae03], set-up parameters can have a more important effect on the performance than the individual choice of a particular learning algorithm. Indeed, classification tasks are often highly unbalanced and the way training documents are sampled has a large impact on performance. In fact, in several large studies, SVM did not systematically outperform other classifiers. For example the work of [Liu05] showed the difficulty to extend SVM to large scale taxonomies. Others showed that depending on experimental conditions, the k nearest neighbors classifier or naive Bayes can achieve better performance [Dav04; Sch06]. We also reported such results [Col06b]. On top of that, selecting the right parameters of the SVM, e.g. the upper bound of the Lagrange multipliers (C), the kernel, the tolerance of the optimizer (ϵ) and the right implementation are non-trivial issues which are seldomly investigated thoroughly.

In this thesis, considering globally the task of classifying text documents, we present a data mining scenario (together with its results when applied) to *compare text classification algorithms*, see Figure 6.1.

6.2 Conducting *fair* classifier comparisons

Although naive Bayes and the k nearest neighbors classifier are multi-class classifiers, the SVM are by default binary classifiers. Then, to handle multi-class problems, SVM usually relies on a *one versus all* strategy where as many binary classifiers as there are classes are trained. For instance, in the case of a classification problem with n-classes, n *one versus the rest* binary classifiers are trained.

Therefore, when running experiments on complex classification tasks involving more than two-classes, we are actually comparing n SVM classifiers (for n classes) to single multi-class naive Bayes or k nearest neighbors classifier. We consider this *unfair*.

Moreover, Fürnkranz has shown that a *round robin* approach using a set of *one against one* binary classifiers performs at least as well as a *one versus all* approach [Für02]. Therefore, we do not limit the generality of the results by studying only *one against one* classification problems.

In addition, to observe and compare the *behaviors* of the classifiers when experimental conditions are varying, these conditions must be controled precisely. Indeed, the properties of the training set can influence largely the learning abilities

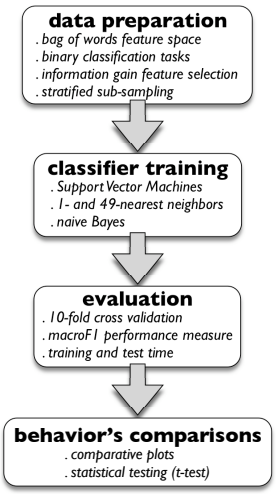

Figure 6.1: *A data mining scenario to compare algorithms in the field of text classification.*

of the classifiers, while in multi-class problems, it can be difficult to understand the particular influence of each class on the classifier's behaviors.

Therefore, in our scenario, we focus on problems with only two-classes. First, it enables us to discard the influence of the multi-class aggregating algorithm in the case of SVM and thus, to compare SVM more *fairly* with naive Bayes and the k nearest neighbor classifier. Second, it also gives the possibility to control more carefully the properties of the training set. In that regard, in order to give to both classes the same chance to be learned as well, we only studied situations where the number of training instances is the same in each class. Last, as binary problems are smaller than multi-class problems, they are usually easier and faster to learn, thus facilitating the conduction of experiments.

6.3 Classification algorithms

Because of their simplicity and their generally good performance reported in text categorization, we compare the SVM with two well known classifiers, namely the k nearest neighbors classifier and naive Bayes. In the following, we first introduce some general notations and then, we introduce the three classifiers formally.

Consider a database of instances $\boldsymbol{x}_i$ and class membership y_i, $i = 1, ..., N$ and d the dimension of the feature space, i.e. the dimension of $\boldsymbol{x}_i$. Denote by a function Φ the mapping in the database between each instance and its class mem-

bership such that $y_i = \Phi(\boldsymbol{x}_i)$. Considering only binary classification problems, this mapping takes values in $\mathcal{C} = \{-1, +1\}$. Then, a classification algorithm can learn this mapping by training and we denote the estimated classification function by $\hat{\Phi}(\boldsymbol{x}_i)$.

6.3.1 k Nearest Neighbors.

Given a test point $\boldsymbol{x}'$ and a predefined similarity metric (sim) that can order the training points by their similarity to $\boldsymbol{x}'$, a k nearest neighbor classification rule will assign to $\boldsymbol{x}'$ the class having the highest similarity score. These scores are calculated by summing up the similarities of the k nearest neighbors in each class. The classification rule compares these scores and return the class having the highest, it is defined as

$$\hat{\Phi}(\boldsymbol{x}') = argmax_{y' \in \mathcal{C}} \sum_{k=1}^{K} \delta(y', \Phi(\boldsymbol{x}_k)) sim(\boldsymbol{x}_k, \boldsymbol{x}'), \tag{6.1}$$

with K the number of nearest neighbors and $\delta(y', \Phi(\boldsymbol{x}_k)) = 1$ if $\Phi(\boldsymbol{x}_k) = y'$, 0 otherwise.

6.3.2 Naive Bayes

For $y' \in \mathcal{C}$, let $P(y')$ be the prior probability of each class. For x_{ij} (feature j of an instance $\boldsymbol{x}_i$), let $P(x_{.j}|y')$ be the probability to observe the feature value $x_{.j}$ conditionally to y'. Then, given a test point $\boldsymbol{x}'$ whose feature values are $(x'_1, ..., x'_d)$, the naive Bayes classification function is expressed by

$$\hat{\Phi}(\boldsymbol{x}') = argmax_{y' \in \mathcal{C}} P(y') \prod_{j=1}^{d} P(x'_j|y'). \tag{6.2}$$

6.3.3 Support Vector Machines

The SVM are based on statistical learning theory [Vap95]. Its theoretical foundations together with the results obtained in various fields makes it a popular algorithm in machine learning.

The SVM classification function of a test point $\boldsymbol{x}'$ is given by

$$\hat{\Phi}(\boldsymbol{x}') = sign(\langle \mathbf{w}.\boldsymbol{x}' \rangle + b) \tag{6.3}$$

with $\mathbf{w}$ and the scalar b, the coordinates of the separating hyperplane and the bias to the origin. The particularity of this hyperplane $\mathbf{w}$ is that it is the one separating the points of the two classes with the maximum distance when these

are linearly separable. This concept of maximum separating distance is formalized by the *geometrical margin* which is defined as

$$\gamma = \frac{1}{2||\mathbf{w}||_2}. \tag{6.4}$$

Therefore, the SVM problem resides in searching the maximum of γ or, alternatively, the minimum of $||\mathbf{w}||_2$ given the constraints. To identify this $\mathbf{w}$, an optimization problem must be solved. Its primal form is expressed by

$$\begin{aligned} &minimize_{\mathbf{w},b} && \tfrac{1}{2}\langle \mathbf{w}.\mathbf{w} \rangle, \\ &subject\ to && y_i(\langle \mathbf{w}.\boldsymbol{x}_i \rangle + b) \geqslant 1, i = 1, ..., N, \end{aligned} \tag{6.5}$$

where the $\boldsymbol{x}_i$ are the training instances in the database. Yet, as the number of training documents is typically several orders smaller than the number of features in text classification, it is usually preferred to operate the SVM in the dual (that depends on the number of training documents). The dual form can be obtained by posing and deriving the Lagrangian according to $\mathbf{w}$ and b:

$$\begin{aligned} &maximize_{\alpha} && \textstyle\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j \langle \boldsymbol{x}_i.\boldsymbol{x}_j \rangle, \\ &subject\ to && \textstyle\sum_{i=1}^{N} y_i \alpha_i = 0 \\ & && 0 \leqslant \alpha_i \leqslant C, i = 1, ..., N. \end{aligned} \tag{6.6}$$

In order to limit values of Lagrange coefficients α_i, an upper bound C is introduced so that each training instance has a maximum contribution when classes are not linearly separable. This type of SVM is referred to as soft-margin SVM.

Concerning the kernel function, even though problems may not always be separable in text classification, a *linear* kernel is commonly regarded as yielding the same performance as non-linear kernels in our text classification domain [Yan99b]. For this reason, we only considered linear kernels in our scenario.

Next, recall that the hyperplane $\mathbf{w}$ is defined by

$$\mathbf{w} = \sum_{i=1}^{N} y_i \alpha_i \boldsymbol{x}_i. \tag{6.7}$$

Then, upper-bounding the Lagrange multipliers gives the constraints

$$0 \leqslant \alpha_i < C. \tag{6.8}$$

Observe that the norm of the hyperplane tends to vanish as C goes to zero

$$\lim_{C \to 0} ||\mathbf{w}||_2 = 0. \tag{6.9}$$

This implies that the geometrical margin goes to infinity ($||\mathbf{w}||_2 > 0$)

$$\lim_{C \to 0} \gamma = +\infty. \tag{6.10}$$

Consequently, lowering C to very small values will eventually lead to a SVM solution where all training instances are within the margin. We further discuss the quality of this SVM solution in coming sections.

Interpreting the SVM solution Recall that a set of points is *convex* if the line segment between any two of its points stays within the set [Str86], and consider the smallest of the convex hulls (the smallest convex set). Then the solution of SVM in a binary classification problem is made of the training points on the smallest convex hulls of the two classes. We can regard these particular points as defining the boundary of the two classes.

In the solution of SVM, the Lagrange multipliers α_i quantify the contribution of each training point to the positioning of the hyperplane. The higher the α_i, the more force the point exerts on the position of the hyperplane. Thus, the points within the smallest convex hull of their respective classes are set *inactive* with $\alpha_i = 0$. Those other points $\boldsymbol{x}_i$ for which $\alpha_i > 0$ are on the convex hulls. They are considered as *active* and referred to as support vectors (SV).

If the two classes are linearly separable, the solution of linear SVM will be the hyperplane in *force equilibrium* between the two convex hulls. This constraint is further discussed in the following paragraph *Settings of text classification*. However, when classes are not linearly separable, points may exert high pressure on the hyperplane without ever being on the right side of it. Consequently, some multipliers may be very large compared to others or even infinite. In order to limit the individual contribution of the multipliers, the so-called soft margin was introduced. In a soft margin SVM solution, the multiplier values are upper bounded by a parameter C, that is the maximal cost that we are ready to pay to classify a training point well. There are four types of training points. We list and characterize them by their distance and their contribution to the hyperplane in Table 6.1.

Table 6.1: *The different types of training instances composing an SVM solution.*

	Distance	Contribution	Active?	Well classified?
(1)	$y_i(\langle \mathbf{w}.\boldsymbol{x}_i \rangle + b) \geqslant 1$	$\alpha_i = 0$	no	yes
(2)	$y_i(\langle \mathbf{w}.\boldsymbol{x}_i \rangle + b) = 1$	$0 < \alpha_i < C$	yes, *in* bound	yes
(3)	$0 < y_i(\langle \mathbf{w}.\boldsymbol{x}_i \rangle + b) < 1$	$\alpha_i = C$	yes, *at* bound	yes
(4)	$y_i(\langle \mathbf{w}.\boldsymbol{x}_i \rangle + b) < 0$	$\alpha_i = C$	yes, *at* bound	no

Recall that the concept of sparsity aims at finding the most parsimonious representation for a problem. Then, in a sparse SVM solution, most of the training

points are set inactive (1) and the ones that are active (2,3,4) define the smallest convex hull of the two classes. The active points are expected to represent the "boundaries" of the two classes.

In linearly separable problems, there are only training points of types (1) and (2) (without the bound C). However, many problems are not linearly separable, which means that the linear separation surface will misclassify part of the training points. Thus, in soft margin SVM, the more a solution has bounded SV's (3 and 4), the less linearly separable the problem is. In addition, we remark that only the bounded SV of type (4) are misclassified training points, in contrast to the bounded SV of type (3) that are well classified.

Large proportions of bounded SV are not desirable because it shows the non-linear separability of the problem. If training points of distinct classes are at the same location in the feature space, no surface of any complexity can separate well those overlapping training points. Therefore, using non-linear kernels would not show any improvement.

Settings of text classification In addition to large number of features, the *bag of words* feature space exhibits high levels of sparsity. The majority of the word occurrences are zero. As the dimensionality of the problem increases, there will be more training points on the smallest convex hull of the two classes. As an example, more than 75% of the dual variables are non-zero in the SVM solutions of [Rou06] in text classification. We will also illustrate this phenomenon through our experiments.

Next, consider the *force equilibrium* between the two classes formalized by the constraint

$$\sum_{y_i \in \{+1\}} \alpha_i = \sum_{y_j \in \{-1\}} \alpha_j, \tag{6.11}$$

where the sum of the individual training point forces should remain equal for the two classes. Then, a specific SVM solution is the one where all the training points are equally weighted. We refer to it as the *nearest mean classifier* solution.

Our experiments suggest that the best performing SVM solutions in large *bag of words* feature spaces are solutions that are similar to the *nearest mean classifier* because most of the training points have equal weight.

Setting the parameters of SVM First, although we tried several values for the parameter ϵ which controls the tolerance of the stopping criterion of SVM, we selected $\epsilon = 0.1$. In fact, while no effect on the performance was observed for other settings, this setting significantly reduced the training time. Second, concerning the C parameter, it is seldom optimized in text categorization and our scenario will investigate its effect in Chapter 8, both on the performance and on the type of SVM solutions.

6.3.4 Implementation of the algorithms

For naive Bayes and the k nearest neighbors classifier, we used the `libbow` library [McC96]. With respect to SVM, we used both the Platt's SMO algorithm in `libbow` and the `libsvm` implementation of [Cha01]. Therefore, in this thesis, our conclusions on SVM do not relate to a particular implementation because we could reproduce them for two different implementations.

6.4 Definition of the scenario

In this section we describe our data mining scenario for the comparison of algorithms in text classification.

The remainder of the section structures as follows. First, in order to compare the classifier's behaviors, we describe our evaluation methodology and its measures. Second, we present the different dimensions of our experimental set-up.

6.4.1 Evaluation methodology and measures

To improve the reliability of our comparative experiments between classification algorithms, we chose an evaluation methodology that, for each experimental condition, repeats the training of the classifiers a number of times. Then, under each experimental condition, we took a set of measurements to picture the classifier's behaviors and finally, we calculated aggregates of these measurements (e.g. the empirical mean).

In the following, we first describe our evaluation methodology and then, our measures.

Evaluation methodology We adopted the 10-fold cross-validation methodology to evaluate the classifier's behaviors. It proceeds as follows. First, the complete database of instances is separated in 10 folds. Second, by a mechanism similar to a rotation, each fold is successively considered as test set whereas the remaining of the instances composes the set of instances *available* for a training set. Indeed, we will show in the next section that the set of *available* training instances will be *sub-sampled*.

Measures We are interested in the ability of the classification model to predict correctly the class of an instance. To assess this, as illustrated in Table 6.2, we group the errors made by the classifiers using a confusion matrix.

From this confusion matrix, we measure the *precision* of the classification model that is, the accuracy to predict a specific class. Considering the target class as A, then the precision is defined by

$$Precision_A = \frac{tp_A}{tp_A + e_{BA}} \tag{6.12}$$

Table 6.2: *Confusion matrix in a two class problem.*

		Predicted class A	Predicted class B
Known class	A	tp_A	e_{AB}
	B	e_{BA}	tp_B

The *recall* is a measure of the ability of a classification model to retrieve the appropriate instances of a certain class in a dataset. Again, considering A as the target class, it is defined by the formula

$$Recall_A = \frac{tp_A}{tp_A + e_{AB}} \tag{6.13}$$

Similarly to [Yan99b], we adopt for our experiments the macro averaged F_1 measure which is defined by

$$maF_1 = \frac{2 \times maPrecision \times maRecall}{maPrecision + maRecall} \tag{6.14}$$

where $maPrecision = (Precision_A + Precision_B)/2$ and $maRecall = (Recall_A + Recall_B)/2$. In words, the maF_1 measure relates the precision and the recall computed in two confusion matrices, interchanging the definition of the target class in the two-class problem (either A or B). We calculate the mean of maF_1 over the ten measures from the cross validation.

Further, we decided to characterize the solutions of SVM in terms of the number of SV *within* and *at* bound. For this purpose, we measured these numbers for all experiments whereas, for a selection of experimental conditions, we also measured *all* the Lagrange multiplier values α_i in order to compare different types of SVM solutions on their quantiles of α_i. Regarding the number of SV *at* and *within* bounds, their empirical mean is estimated.

Last, the global processing time in seconds is recorded in the course of our experiments. This processing time includes both the training and the test time. We used two types of computers in our experiments: Pentium III 1Ghz with 1GB of RAM in Chapter 7 and AMD Opteron dual 2.6Ghz with 4GB of RAM in Chapter 8. All these computers were using the Linux operating system where the graphical interface was disabled.

6.4.2 Dimensions of experimentation

As classifiers are influenced by the number of training documents and by the features choosen, we decided to examine these issues in detail and we compared the classifiers when varying both dimensions.

In the remainder, we first present our strategy to prepare and sub-sample the training sets. Second, we describe how we reduce the size of the feature space.

Then we present our third dimension of experimentation for SVM. Finally, we report which series of values we use to do the experimentations in two (Chapter 7) or three dimensional (Chapter 8) spaces of experimentation.

Number of training documents For each classification task, the procedure to prepare the training sets can be structured as follows.

1. The database of instances is separated by class.
2. For each class, the instances are ordered randomly.
3. For each class, the instances are separated into ten folds.
4. For each class and each test fold, the set of instances *available* for the training set is the merging of the remaining folds.
5. For each class and each test fold, the trainsets are sub-sampled into sets of increasing size.
6. For each sub-sample, the instances from both classes are merged into one of the ten folds.
7. For each sub-sample, the instances are re-ordered randomly.

To study the behaviors of the classifiers when the number of training documents was increasing, the set of *available* training instances is sub-sampled. The sub-sampling creates training sets of increasing size but with equal number of cases from each class. Thus, both classes were given the same chance to be learned as well. For instance, a training set of size 90 would have 45 instances of both classes. Further, when varying the experimental condition from 90 training instances to 128, we preserved the 90 first and grew the training set of 38 new instances, i.e. 19 from each class.

Following this procedure enabled us to reduce the variability in our experiments, in particular due to our sampling strategy.

Number of features To study the influence of varying the size of the feature space on the classifiers, we chose the *information gain* heuristic as a means to select a subset of features. Our choice stems from the good overal performance of this heuristic as well as its simplicity [Yan97; Rog02]. In the following, we first introduce some notations and then, the *information gain*.

Denote by $p_y = P(Y = y)$ the probability to observe an instance from class $y \in \mathcal{C}$. Recall that d is the dimension of the *bag of words*. Denote by $X = x_{.j}$ with $j = 1, ..., d$ the distribution of the presence / absence of a word j in a set of instances.

First, a result from *information theory* states that to classify instances in $\mathcal{C}$, the optimal algorithm coding the class needs an average number of bits given by

$$H(Y) = -\Sigma_{y \in \mathcal{C}} p_y \log_2 p_y. \tag{6.15}$$

This quantity $H(Y)$ is referred to as the *entropy* of the distribution Y. The *entropy* is high if the distribution of Y is even over all values and low if the distribution is varied.

Second, we can define the average conditional entropy of a distribution X given the one of Y by

$$H(X|Y) = \Sigma_{y \in \mathcal{C}} P(Y = y) H(X|Y = y). \tag{6.16}$$

This second quantity estimates the average number of bits to code the values of X if the class Y is known. Knowing more about the problem, here the class, may help to identify a more efficient coding scheme that exhibits a lower entropy.

Finally, the *information gain* is defined as a function of these two quantities by

$$IG(X = x_{.j}|Y) = H(X = x_{.j}) - H(X = x_{.j}|Y). \tag{6.17}$$

It calculates the average number of bits that could be saved when predicting the occurence of a word $x_{.j}$ if the distribution of Y was known. The higher is the *information gain*, the more the class Y associates with the occurence of the word $x_{.j}$.

Therefore, when applying the feature selection heuristic, we search for words in the *bag of words* feature space that exhibit the highest *information gain* in the training set. Finally, recall that features are ranked and selected by information gain at the start of *each* experiment.

Dimensions of experimentation We do measurements in two (Chapter 7) or three dimensional (Chapter 8) spaces. These axis are the number of training documents, the number of features and the parameter C of the SVM classifier. The conditions of experimentation are determined by series of exponential values on each axis because the phenomenas that we study are not linear.

For both the number of training documents and of features, the values follow the series given by $2^{\frac{i}{2}+b}$ with $i = 1, 2, 3...$ and $b \in \mathbf{N}$, e.g. the series starts with $\{90, 128, 181, 256, ...\}$ when $b = 6$.

Concerning the values of C, they follow the series 10^i, e.g. $\{0.0001, 0.001, ..., 1000\}$.

6.5 Experimental data

This section outlines as follows. First, we describe the text classification datasets used in the comparison of algorithms in Chapter 7. Second, we present the datasets used in the study of Chapter 8 where we investigate SVM's scale-up in large *bag of words* feature spaces.

6.5.1 To study the behaviors of the classifiers

For our experiments we used two well known datasets: `20newsgroups` and `ohsumed-all`. The `libbow` library was used to process the text data [McC96].

1. The `20newsgroups` dataset is composed of 20000 newsgroup emails [Het99]. We removed the headers of the emails and no stemming[1] was performed.

2. The `ohsumed-all` dataset[2] is taken from the Ohsumed corpus which was initially compiled by William Hersh (`ftp://medir.ohsu.edu/pub/oshumed/`). The dataset is made of 50216 medical abstracts categorized into 23 cardio vascular disease categories. Although Joachims used only the first 10000 medical abstracts for training and the second 10000 for testing [Joa98], in our experiments we use all 50216 documents for the cross validation. To be consistent with our previous processing of the `20newsgroups` dataset, we did not perform stemming.

On these datasets, we chose to study the set of *one against one* binary classification tasks. Thus, the total number of classification tasks on `20newsgroups` was

$$\frac{20(20-1)}{2} = 190. \tag{6.18}$$

For `ohsumed-all`, because of computing time considerations, we decided to limit our analysis to the first 162 classification tasks taken in alphabetical order out of the

$$\frac{23(23-1)}{2} = 253 \tag{6.19}$$

possible binary problems.

Therefore, we performed experiments on a total of 352 classification tasks.

6.5.2 To study the scale-up of SVM in large bag of words feature spaces

From the previous study which involved 352 tasks, we selected two binary text classification problems where SVM exhibited a performance drop. In order to further validate our conclusions, we also performed additional experiments on the Reuter Corpus Version 1 dataset (`RCV1`) [Lew04].

This last dataset interests us particularly because it is available in its processed form. The features were extracted and the word frequencies were *tfidf*-transformed. See [Lew04] for a detailed description of the dataset processing. Because in our previous study, we used the word frequencies to represent the classification tasks, these additional experiments on `RCV1` will enable to show the influence of the feature space transformation on the performance and the nature of SVM's solutions.

[1] Process of reducing words to their root.

[2] `http://dit.unitn.it/~moschitt/corpora/ohsumed-all-docs.tar.gz`.

In the remaining of this thesis, we will refer to the three classification tasks by the following acronyms:

20ng "alt.atheism / talk.religion.misc" from the `20newsgroups` dataset.

C01-C21 "Bacterial Infections and Mycoses / Disorders of Environmental Origin" from the `Ohsumed-all` dataset.

RCV1 Reuters Corpus Version 1.

6.6 Concluding remarks

We investigate the problem of automatically classifying text documents into categories which relies on standard machine learning algorithms. These algorithms, given a set of training examples, can learn a classification rule in order to further categorize new text documents automatically.

Among the algorithms suggested for use in text classification, the most prominent one is Support Vector Machines and repeatedly, it was shown to outperform other techniques. Yet, we consider that some of the previous comparative experiments of algorithms were not fairly conducted (see the discussion in the section *Related work* of chapter 7). In fact, other studies [Dav04; Sch06; Col06b] have shown that in some situations, other algorithms like naive Bayes or the k nearest neighbors classifier give better results than SVM.

Therefore, we first introduced the problem of classifying text documents into categories. Next, with respect to previous comparative studies, we discussed fairness issues when comparing algorithms. Then, given this focus, we described our data mining scenario that aims to compare as fairly as possible classification algorithms. It will help us to better understand the problem of classifying text documents into categories.

Chapter 7

Comparison of Classifiers

In this chapter, on a large number of binary text classification tasks, we describe comparative experiments between the Support Vector Machines (SVM), the k nearest neighbors classifier and the naive Bayes classifier. First, as some algorithms like the SVM and the k nearest neighbor classifier can accept different parameters, we perform experiments in order to limit the subsequent study to a selection of parameter-optimized classifiers. Second, using these optimized versions of the classifiers, we report comparative experiments on the behaviors of the classifiers when the number of training documents and the feature space size are increased.

7.1 Introduction

In this comparative study based on binary classification tasks, we seek answers to the following questions.

1. Should we still consider "old" classification algorithms like the naive Bayes and the k nearest neighbors classifier in text categorization or opt systematically for Support Vector Machines (SVM) classifiers?

2. What are the strengths and weaknesses of these algorightms on a set of binary text classification problems?

3. Are there some parameter optimization results transferable from one problem to another?

Before answering the above questions, our parameter optimization results are presented. The optimized versions of the classifiers are then used in the subsequent comparative study.

7.2 Experimental data

For the experiments, we use the data introduced in Chapter 6 (section 6.5.1): the `20newsgroups` dataset composed of 20000 newsgroup emails classified into 20 categories and the `ohsumed-all` dataset composed of 50216 medical abstracts classified into 23 categories.

7.3 Parameter optimization

In our comparative study, we will compare classification algorithms when the feature space size and the number of training documents is varying on a large number of binary text classification problems. As some algorithms like the SVM and the k nearest neighbor classifier can accept different parameters, we decided to perform some experiments in order to limit the subsequent comparative study to a selection of parameter-optimized classifiers.

We ran these experiments on three classification tasks from the `20newsgroups` dataset [Het99]:

1. "alt.atheism / talk.religion.misc".

2. "comp.sys.ibm.pc.hardware / comp.sys.mac.hardware".

3. "talk.politics.guns / talk.politics.misc".

In these experiments, we used a 10-fold cross validation. For each fold, the training set included 1800 documents and the test set 200. Regarding the feature space, all features were used.

In the following subsections, we report our parameter optimization results for SVM and the k nearest neighbors classifier.

7.3.1 Support Vector Machines

Various parameters of SVM can be considered in the attempt to optimize the performance of this algorithm. The parameter C (relative importance of the complexity of the model and the error) was varied and various kernel functions were tried as well; in these settings (i.e. high feature space), none of those lead to interesting improvements in terms of performance (maF_1, cf. Chapter 6, section 6.4.1) or processing time. So, the default value $C = 200$ and a *linear* kernel are used. In particular, this choice for a *linear* kernel is consistent with previous results [Yan99b; Zha01].

We also varied the ϵ parameter which controls the accepted error of the SVM optimization problem solver. For different values of ϵ, Figure 7.1 (a) shows the dependence of the processing time of SVM on the size of the feature space. Figure 7.1 (b) is similar but shows the dependence of the processing time on the number of documents. We have found that ϵ has no influence on maF_1 as long as its value was smaller or equal to 0.1. However, when the largest value of ϵ was used, the processing time could be reduced by a factor of four in the best case.

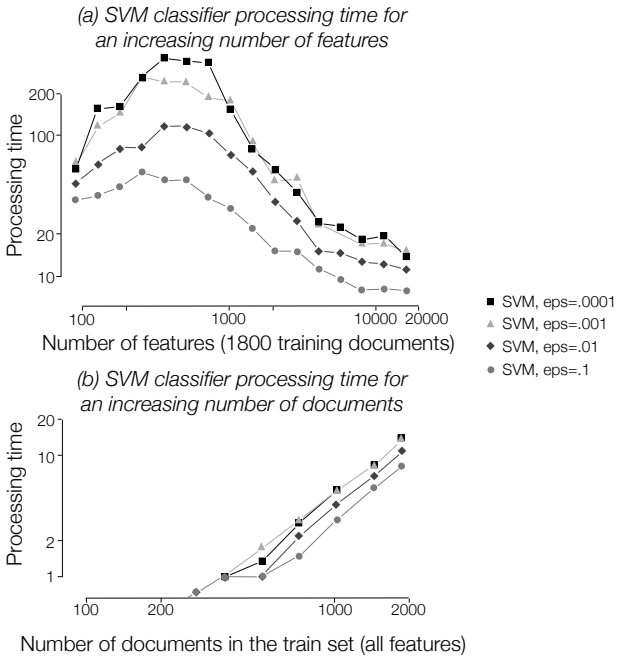

Figure 7.1: *SVM processing time (in seconds) for several values of ϵ for an increasing number of features (a) and an increasing number of documents in the training set (b). Experiments were performed on* `alt.atheism` *vs.* `talk.religion.misc` *from the* `20newsgroups` *dataset.*

Therefore, when relaxing the error constraint ϵ of the solver for the set of acceptable hyperplanes, we reduce the time necessary to conduct the optimization. Thus, for larger ϵ, the solver will find more rapidly an hyperplane that matches the stopping criterion. Further, as varying ϵ has nearly no effect on the performance, we may assert that a "rough" solver precision is sufficient to train SVM

on `20newsgroups` and eventually, by extension, in text classification problems characterized by high dimensions.

7.3.2 k Nearest Neighbors

For the k nearest neighbor classifier we performed experiments to select the best number of neighbors k and the best feature space transformation.

Best number of nearest neighbors To compare the generalization of the nearest neighbors classifier for different k, we evaluated each setting on the three tasks using 10-fold cross validation. As a result, each setting was characterized by a series of 30 maF_1 measures. We compared these sets of measures via a pairwise t-test to assess whether a k gives statistically a better performance than other ones. We implemented a voting scheme that attributed a "victory point" to the best k when the p-value was lower than 5%. When the difference was not significant, no point was given. Finally, we selected the best k by counting the number of wins, see Figure 7.2.

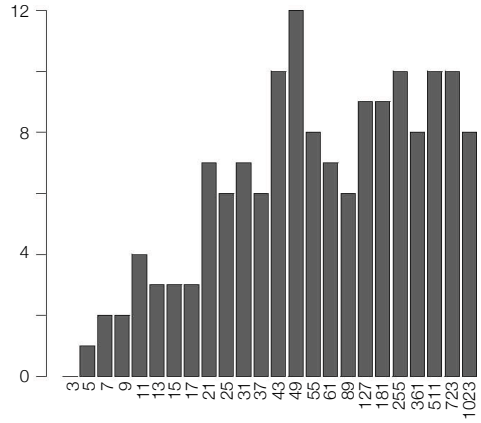

Figure 7.2: *Count of pairwise wins for each number of nearest neighbors. Note that for $k = 49$, the nearest neighbors classifier exhibits the largest count; yet, as illustrated by their high counts, large values of k tend to perform also well.*

In our experiments, we selected $k = 49$ to be the best number of nearest neighbors and therefore, the subsequent comparative study is based on the 49 nearest neighbors. Interestingly, this optimal k value (49) is quite close to the one

in [Yan99b], who suggested $k = 45$. Yet, experimental settings differed substantially as first, Yang performed her experiments on the `Reuters-21578` dataset [Het99] and second, the classification tasks were essentially multi-class.

In addition, most large values of k gave also good performance. To explain this, we have considered the similarity measure used in the k nearest neighbors classifier. In its calculation, the class-contribution of each neighbor is weighted relatively to its similarity to the test point. Therefore, for large k, the additional neighbors hardly contribute anything to the class score because they are too dissimilar to the test point. Yet, the more neighbors, the larger the computing time. Therefore, we selected the first best performing number of nearest neighbors (49) and besides the 49 nearest neighbors classifier, we also included the 1 nearest neighbor classifier as a baseline.

Best transform To achieve better performance with the k nearest neighbor classifier, we can try to transform the feature space. Such a feature space transformation (ϕ) involves three types of data:

1. The number of occurences of the i^{th} term (tf_i).

2. The *inverse document frequency* (idf) which is the ratio between the total number of documents N and the number of documents in the database that contain the j^{th} term (df_j).

3. A normalization constant (κ) making $\|\phi\|_2 = 1$.

To compare the different feature space transformations, we apply a similar evaluation procedure as for identifying the best k. We report the results in Appendix B, Figures B.1, B.2 and B.3.

Any transformation was found suitable except for the *binary* one, which reduces the term frequencies to binary variables for the presence and absence of a word in a document. This particular result is consistent with a previous study where it was also found that the *binary* transformation performed worst [McC98]. Concerning the *inverse document frequency*, it is regarded necessary because it decreases the importance of common words occurring in numerous documents. Yet, our experiments show that those transformations improved *only* slightly the performance. We may attribute this to the fact that in the considered classification tasks (`20newsgroups`), the email data has rather short texts, thus limiting the potential influence of the *inverse document frequency*. Finally, concerning the normalization, we could not identify any effect on the performance.

In all subsequent comparative experiments, we have adopted the `ntn.lnc` transformation because it achieved the best results. In this case, the feature

space transformation for the documents in the training set is

$$\phi_{\texttt{ntn}}(x_{ij}) = tf(x_{ij})\ log\left(\frac{N}{df_j}\right) \tag{7.1}$$

whereas the transformation for the documents in the test set is

$$\phi_{\texttt{lnc}}(x_{ij}) = \frac{log(tf(x_{ij}))}{\kappa}. \tag{7.2}$$

7.4 Comparisons for increasing document and feature sizes

The aim of our experiments is to examine the classifier learning abilities for an increasing number of documents in the training set, preserving the balance between documents of both classes; *all the features* were selected.

We also consider how the performance is affected by the size of the feature space; then, the training set contains all available documents with as many documents of both classes.

On the influence of experimental set-up We observed that the parameters related to the experimental set-up (sample selection, feature space, feature subset selection, classifier parameters) had a *larger* impact on the performance than the choice of individual classifiers. In fact, if suitable parameters of the set-up are chosen and if the parameter settings of the classifiers get correctly optimized, then the performance of the algorithms hardly differ. This is illustrated in Figure 7.3 (b) for an increasing training set size where the maF_1 performance of the 49 nearest neighbors classifier, naive Bayes and the SVM perform very similar. This result is *typical* of what we observed on other 352 classification tasks.

Comparative performance behavior Figure 7.4 illustrates that the 49 nearest neighbors classifier and naive Bayes often start with an advantage on SVM when the training sets are composed of a small number of documents. However, as the number of documents increases, this difference diminishes. Most of the time, when the whole training set is used, the performance of SVM is very similar to the one of 49 nearest neighbors and naive Bayes. Yet, for the larger training sets, it is rare to see SVM performing better than the two other classifiers.

With respect to the number of features, 49 nearest neighbor and naive Bayes tend to reach the best performance on a medium sized feature space. Most of the time, the performance of the classifier remains at the top, or increases *very slightly* for any larger number of features. But it does also occur that an increasing number of features leads to a drop of performance.

Comparative processing time behavior The SVM is in a clear disadvantage when we consider the processing time; this is not only much higher than for the other

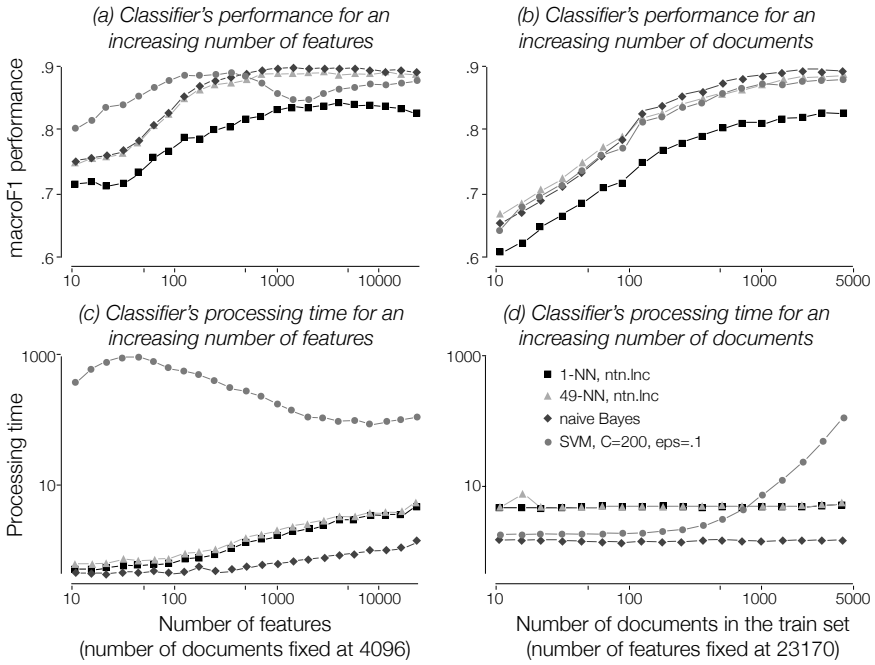

Figure 7.3: *Classifier's performance (a, b) and processing time (c, d) on C01-C21 (`Ohsumed-All`) given an increasing number of features (a, c) and of documents (b, d).*

algorithms but, as Figure 7.3 (d) illustrates, super linear in the number of training documents. However, the processing times of naive Bayes, the 49 and 1 nearest neighbors classifiers depend only on the size of the test set. We also notice that when the number of documents increases, the processing time of these classifiers remains the same, see Figure 7.3 (d). Yet, for the same number of training documents, if we compare classification tasks having different total number of documents (and accordingly different test set sizes), processing time differences are observed.

Furthermore, as Figure 7.3 (c) illustrates, both naive Bayes and the k nearest neighbors classifiers are affected by the number of features. Comparatively, the training time of the SVM is particularly high, especially for small feature spaces. This result may be due to the solver whose task in small feature space is harder than in larger ones. In fact, as the dimensionality increases then the classification

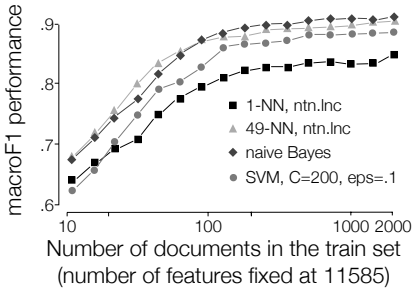

Figure 7.4: *Classifier's performance given an increasing number of documents in the training set on the task C02-C13 (`Ohsumed-All`).*

problem becomes more linearly separable, which tends to ease the task of finding a proper separating hyperplane. Therefore, the training time will be longer in small feature spaces than in larger ones.

A performance drop for SVM As Figure 7.3 (a) illustrates, we observe on many classification tasks a *wave pattern* on SVM performance when the feature space size is varied. As part of this *pattern*, large feature spaces do not necessarily lead to best performance. In fact, on those tasks, small feature space SVM classifiers would, first, exhibit performances that compare with the best ones shown by the 49 nearest neighbors classifier and naive Bayes, and second, perform better than large feature space SVM. Furthermore, for particular small feature space sizes, SVM outperforms other classifiers with an advantage that could be as high as 25% on some tasks.

These results are somewhat surprising, since SVM is often regarded as an algorithm that deals well with very large number of features; here, it appears that naive Bayes and the 49 nearest neighbors classifier do this better.

We explain part of the phenomena by recalling that the condition for SVM to identify an optimal separating hyperplane is only met when the number of documents in the training set is sufficiently large. Thus, this would explain why SVM is outperformed for small training set sizes and why, for small feature spaces with large training sets, it performs that well. Yet, it remains unclear why small feature space SVM can perform better than large feature space SVM.

7.5 Related work

Our results differ somewhat from previous comparative studies.

For example in [Dum98], Platt's SVM SMO algorithm was presented to outperform the naive Bayes; yet, only 50 features were selected for the naive Bayes which we consider to be conservative as in our experiments the naive Bayes performed best with mid-sized feature spaces (few thousands features). On the contrary, SVM was used with 300 features which may not be far from its optimal setting: few features, many documents. Indeed, we have also shown via our experiments that SVM generally outperforms other classifiers in small feature spaces.

Other studies found the naive Bayes to perform worse than SVM and the k nearest neighbors [Yan99b; Zha01]. In [Yan99b], the features space size appears consistent with our results (2000 for the naive Bayes, 2415 for the k nearest neighbors classifier and 10000 for SVM), however the experimental set-up differ substantially as the `Reuters-21578` dataset was used [Het99].

We did not consider `Reuters-21578` for our experiments because in that dataset, the document frequency per class varies widely, with about 33% of the categories having less than 10 documents and the top two having more than 2000 documents each. Therefore, we would not be able to study learning curves when varying the training set size because the inequal class distribution would disallow to build balanced training sets.

Moreover, comparisons were done on multi-class classification tasks [Yan99b; Zha01] or on the averaged performance of the set of *one against all* classification tasks [Dum98; Zha01]. But as explained earlier, comparing a single multi-class naive Bayes or k nearest neighbors classifier to n SVM classifiers, with n the number of categories, is definitively not fair for the naive Bayes and the k nearest neighbors classifier. Besides, comparing the aggregated performance of classifiers does not satistfy us as it may hinder our precise understanding of the classification problem.

7.6 Concluding remarks

When investigating the best parameter settings for the SVM, the *linear* kernel was found to be the best choice, which is consistent with previous work. Besides, a large value of ϵ was shown to conduct equally performing, yet SVM classifiers were faster to train. With respect to the best feature space size, SVM exhibited generally good performance for small or medium sizes, which surprises us as SVM is commonly said to best perform in very large feature spaces. Finally, regarding the k nearest neighbors classifiers, the optimal number k of neighbors was very similar to the one published previously.

In terms of overal performance, the k nearest neighbors classifier, the naive Bayes and the SVM perform similarly if suitable parameter settings are used. These results are in agreement with a study [Dae03] showing that the set-up pa-

rameters influence more the performance than the individual choice of a particular learning technique. Therefore, one should keep considering the k nearest neighbors classifier and the naive Bayes classifier as possible options because they are fast, simple and well understood. Regarding SVM, perhaps it can handle better complex classification tasks, but it remains to be seen how we can identify them; moreover, it is costly to train SVM.

Results depend on the evaluation methodology and we have focused here on binary classification tasks. New experiments should be carried out to explain why the naive Bayes behave so well on *one against one* classification tasks in contrast to its behavior on *one against all* tasks. We are also interested to understand more precisely SVM behavior as it exhibited an uncommon performance pattern shaped as a *wave* when the size of the feature space increases. Finally, to recommend a classifier with suitable parameter settings, a way to characterize classification tasks should be investigated, eventually via the use of a meta-learning strategy.

Chapter 8

Does SVM Really Scale up to Large Bag of Words Feature Spaces?

In this chapter, we aim at developing better understanding of Support Vector Machines (SVM) in text categorization problems represented by sparse bag of words feature spaces. First, we identify those experimental settings that lead to best performing SVM solutions. Second, we discuss the nature of the solutions in these situations. Then, we describe a performance drop for SVM that occurs for particular combination of number of documents and of features. Next, we propose to relate the performance drop to classification noise in the data and we validate this hypothesis by additional experiments. Finally, before concluding, we discuss related work.

8.1 Introduction

We are concerned with the problem of learning classification rules in text categorization. Many authors presented Support Vector Machines (SVM) as the leading classification method [Joa98; Dum98; Yan99b; Zha01; Yan03; For03]. A number of studies, however, pointed out that in some situations SVM is outperformed by simpler methods such as naive Bayes or the nearest-neighbor rule [Dav04; Sch06; Col06b]. In this chapter, we study in detail the performance of SVM and the number of support vectors when varying the training set size, the number of features and, unlike existing studies, also the SVM free parameter C, which is the upper bound of SVM's dual representation (Lagrange multipliers).

In Chapter 7, we only searched for optimized versions of SVM in large *bag of words* feature spaces and we used subsequently these SVM for a comparative analysis when both the number of training documents and of features were varied.

This new study is different because we decided to also vary the parameter C and to study in detail the nature of the SVM solutions in terms of the number of Support Vectors (SV).

If in Chapter 7, we did not find C to influence the performance, here we show that tightly constrained SVM solutions with small C are high performers. However, most training instances are then bounded support vectors in these SVM solutions, which means that instances tend to be equally weighted with $\alpha_i = C$. Yet, an SVM solution where all the training instances share an equal weight is similar to the one of a nearest mean classifier. Because for such SVM solutions no training is necessary, it raises an interesting question on the merits of SVM in sparse *bag of words* feature spaces. In our experiments, we also report that SVM suffer from performance deterioration for particular training set size/number of features combinations.

The outline of this chapter is as follows. First, we identify the experimental settings that lead to the best performing SVM solutions. Second, we study in detail the nature of the SVM solutions for those experimental settings. Next, we report results that show a performance drop for SVM and we try to explain these results. Finally, we discuss related work and we conclude.

8.2 Experimental data

For the experiments, we use the data introduced in Chapter 6 (section 6.5.2): `C01-C21` from the `Ohsumed-all` datasets, the `20ng` from the `20newsgroups` datasets and the `RCV1` (Reuters Corpus Version 1).

8.3 Best performing SVM

In Figure 8.1 (a) and (c), we illustrate the performance behavior of SVM solutions when experimental settings are varying. In Table 8.3

Table 8.1: *Summary of the experimental conditions of Figures 8.1 (a) and (c).*

task	train (d)	test	features (f)	C	plotted data
(a) `20ng`	$d = 1448$	200	$f = 2^{\frac{i}{2}+b}, b = 3$	10^i	$maF_1(f, C)$
(c) `C01-C21`	$d = 4096$	547	$f = 2^{\frac{i}{2}+b}, b = 3$	10^i	$\frac{maF_1(f,C)}{max_C\ maF_1(f,C)}$

In (a), SVM performance is illustrated by classical learning curves. The size of the feature space is increasing and we choose different values of C. Similarly, the performance pattern of SVM on `C01-C21` is illustrated in (c) but this time, in order to discard the performance variability associated to the size of the *bag of words*, performance is normalised relative to the best C setting given a number of features. Thus, the contour line labelled 0.94 should be understood as 6% below

the performance of the best performing C setting ($C = 0.01$), whose contour line equals 1. In Table 8.2 we describe the experimental settings for Figure 8.1 (b) and (d).

Table 8.2: *Summary of the experimental conditions of Figures 8.1 (b) and (d).*

task	train (d)	test	features (f)	C	plotted data
(b) `C01-C21`	$d = 2^{\frac{i}{2}+b}, b = 3$	547	$f = 2^{\frac{i}{2}+b}, b = 3$	10^{-2}	$maF_1(f, d)$
(d) `C01-C21`	$d = 2^{\frac{i}{2}+b}, b = 6$	547	$f = 2^{\frac{i}{2}+b}, b = 3$	10^{2}	$maF_1(f, d)$

In (b) and (d), the performance differences between solutions with small (0.01) and large (100) C are illustrated through the contour lines; the number of features and the size of the training set are varying.

In Figure 8.1 (a) and (c), we illustrate how C affects the performance of SVM as the number of features is changing. First, for small feature spaces, i.e. 90-256 in (a) and 11-362 in (c), most of the C settings produce equally performing SVM solutions. However, as the feature space is further enlarged, performance varies widely from one setting to another. In particular, some C settings show a performance drop that will be discussed in a following section. In large feature spaces on (a) and (c), very small C values $\{0.01, 0.001\}$ are the best performing. The difference between the best C setting and others may be relatively small as in (c), i.e. 2-4% (between contour lines 0.96 and 0.98), but this depends on the classification problem. A final remark based on (c) relates to the setting $C = 10^{-4}$ that yields systematically lower performance. As will be shown in the following section, setting C to a very small value like 10^{-4} yields an SVM solution where all training documents are equally weighted with 100% bounded SV. This SVM solution becomes similar to the one of the *nearest mean classifier*.

More generally, we illustrate through (a) and (c) that C is a parameter that should be tuned in order to obtain the best performance of SVM. However, most of the studies in text categorization use default values of the implementation and thus neglect to tune C when training SVM[1]. Additionally, provided that C is tuned, we also illustrate that the best performances are achieved for the largest feature spaces. This confirms that domain knowledge in the form of feature selection is unnecessary for SVM. Finally, (b) and (d) show that more training documents yield better performance, as this is expected.

8.4 Nature of SVM solutions

In Figure 8.2 (a) and (b), the proportion of bounded SV, denoted by $\%(\alpha_i = C)$, is characterized by the contour lines; the feature space size and C vary. In (c),

[1]SVMLight sets the default value of C according to a heuristic. However, most other SVM implementations have fixed C default values (e.g. WEKA, libbow and libsvm).

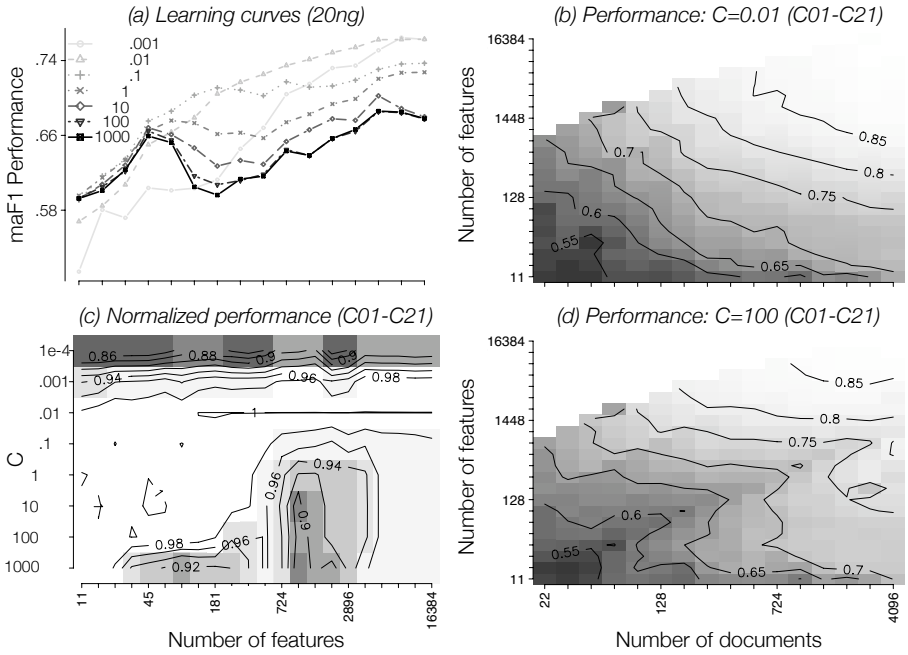

Figure 8.1: *Figure (a), (b) and (d) illustrate that SVM performance generally increases as the feature space and the training set are increasing. Depending on the C, performance may show a drop as in (a) around 724 features (`20ng`) and in (c) around 1448 features (`C01-C21`). In addition, (d) illustrates the linear dependency of the drop to the number of training documents and of features. Finally, (c) shows that the range of best performing C (in white, contour line* 1*) reduces towards small C values as the feature space is increasing.*

the solutions for different C are described when there are 11585 features. We only show it for this number of features because through Figure 8.1 we show that feature selection is not needed for SVM. Multiplier values α_i of the SVM solutions from the 10-fold cross validation are normalised by C and in the subfigure (c), we report the quantiles of the $\frac{\alpha_i(C)}{C}$ ordered by increasing value along the x-axis.

Finally, (d) reports the variation of the cross validated mean of the total number of SV (Support Vector) / in bound SV / at bound SV, when the feature space size varies and $C = 100$. We especially look in detail at these SVM solution because in Figure 8.1 (c), SVM displays a performance drop when the number of features matches the number of training documents per class, i.e. 2048 for a training set of 4096 documents.

Table 8.3: *Summary of the experimental conditions of Figure 8.2.*

task	train (d)	test	features (f)	C	plotted data
(a) `RCV1`	$d = 16384$	2024	$f = 2^{\frac{i}{2}+b}, b = 3$	10^i	$\%(\alpha_i = C)$
(b) `C01-C21`	$d = 4096$	547	$f = 2^{\frac{i}{2}+b}, b = 3$	10^i	$\%(\alpha_i = C)$
(c) `C01-C21`	$d = 4096$	547	$f = 11585$	10^i	$Q\left(\frac{\alpha_i(C)}{C}\right)$
(d) `C01-C21`	$d = 4096$	547	$f = 2^{\frac{i}{2}+b}, b = 3$	10^i	$\#(\alpha_i), C = 100$ $\#(0 <= \alpha_i < C)$ $\#(\alpha_i = C)$

Small feature space SVM's To better understand the behavior of SVM in small feature spaces (11-362 features), we put Figures 8.1 (c) and 8.2 (b) next to each other. For 8.1 (c), we first remark that any C setting performs equally. Then, in 8.2 (b), we notice that for those settings, SVM solutions exhibit large proportions (30 to 100%) of bounded SV, which have $\alpha_i = C$.

In fact, as the *bag of word* feature space is discrete and sparse, training points are located in a finite number of positions. Then, when representing the problem with only few features, it is likely that *training points* can belong to distinct classes; yet, no hyperplane can separate these overlapping points and as a result, their α_i in SVM will tend to infinity unless a soft margin C bounds them ($\alpha_i = C$). Remarkably, we also noticed that at constant feature space size, the number of overlaps will increase along with the number of training documents.

To conclude, it seems that *small feature space SVM's* are mostly defined in terms of bounded SV because of the points from distinct classes which overlap and are non-separable. *Small feature space SVM's* are therefore very similar in terms of proportions of inactive training points / bounded and unbounded SV, which may explain why they yield the same performance.

Large feature space SVM's As illustrated in Figure 8.2 (a), (b) and (d), in contrast to the small feature space SVM which are mostly characterized by bounded SV, the *large feature space SVM* are mostly based on unbounded SV. In fact, the Figure 8.2 (d) shows that the number of unbounded SV raises along with feature space.

This demonstrates that every training point tends to define its own local class boundary as feature space increases and correspondingly, that the definition of the convex hull of each class requires more training points in higher dimensions.

In Figure 8.2 (d), the number of bounded SV stagnates as the feature space increases. These residual SV are *outliers*, e.g. due to a mis-labelling, that would lie within the other class concept, thus avoiding any possible good classification by a linear hyperplane. Furthermore, we interprete the low performance of *large*

feature space SVM having high C in Figure 8.1 by the influence of those outliers whose $\alpha_i = C$ is likely to contribute too much to the final solutions.

Tightly constrained SVM's In Figures 8.2 (a) and (c), as C become very small, the proportion of bounded SV in SVM solutions tends to 100% for all feature space sizes; here, we are especially concerned with the *tightly constrained SVM* ($C = 10^{-4}$) that exhibits 100% of bounded SV.

In section 6.3.3, we introduced the *geometrical margin* and we expressed its limiting value to infinity as C would near zero. Then, if a sufficiently small value of C is taken, the *geometrical margin* will enclose all training points leading to SVM solutions solely defined in terms of bounded SV ($\alpha_i = C$). As all training points have equal weight, these solutions reduce to a simple *nearest mean classifier*.

8.5 A performance drop for SVM

In the study discussed in Chapter 7, we observed a "drop" in performance in a number of classification tasks. This means that the drop does not relate to a particular task, although we do think that its importance does. Similarly, by repeating our findings given two different SVM implementations, we could not relate the drop to the algorithm nor to its implementation. Here, we further analyse the SVM solutions in order to better understand the cause of the performance drop.

In Figure 8.1 (a), we observe a performance drop for 724 features when $C \in \{1, 10, 100, 1000\}$; on a second classification task illustrated in Figure 8.1 (c), a drop also occurs when $C \in \{1, 10, 100, 1000\}$ but for (1024-2048) features. Further, Figure 8.1 (d), which reports SVM performance in terms of color / contour lines, illustrates particularly well that the performance drop occurs consistently for particular combinations of the number of training documents and the number of features.

In fact, it seems that this drop occurs when there are as many features as there are training documents per class; thus, 724 for `20ng` in Figure 8.1 (a) and 2048 for `C01-C21` in Figure 8.1 (c). Moreover, if we compare Figures 8.1 (c) (when $C = 100$) and 8.2 (a), we even notice that the performance drop matches with the drop of the total number of SV. In that case, the number of bounded SV declines faster than the number of *within* bounds SV.

Yet, lowering C influences the nature of the solutions (inactive, bounded, unbounded SV) such that solutions are mostly defined in terms of bounded SV, eventually reducing the SVM to a *nearest mean classifier* if C is set very small. In solutions where there are many bounded SV, these SV do not transform into unbounded which may explain why the learning curves of SVM with small C are not subject to a performance drop.

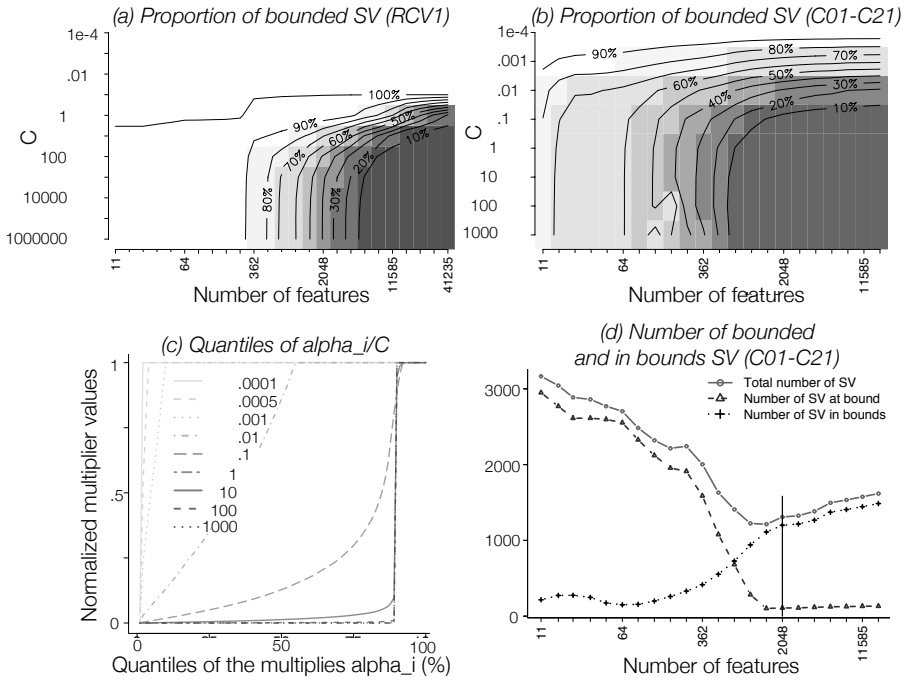

Figure 8.2: *In (a), the proportion of bounded SV ($\alpha_i/C = 1$) increases as C diminishes, and there are only bounded SV in SVM solutions for very small $C \in \{10^{-4}, 5\times10^{-4}\}$. In (b), the proportion of bounded SV is high for all C (11-362 features), but it decreases for higher feature space sizes as shown in (c). First, the total of SV follows the decrease in bounded SV (11-362). Then, the decrease is partially compensated by the appearance of unbounded SV (362-16384). The total of SV exhibits a drop slightly before 2048 features. In (b), for very small $C = 10^{-4}$, solutions consist only of bounded SV (100%) for any feature space size.*

8.6 Relating the performance drop to *outliers* in the data

In previous section, we showed that SVM is subject to a performance drop for particular combination of feature space size and number of training documents. Here, we conduct additional experiments on a classification task (`RCV1`, [Lew04]) in order to further validate the existence of the performance drop.

Performance drop on `RCV1` If we consider Figures 8.1 (c) and 8.3 (a), we notice that 8.3 (a) also exhibits performance drops for different C values $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{3}, 10^{4}, 10^{5}\}$. Similarly, we see in 8.3 (a) that the best performing C values (where the contour lines equal 1) tend to lower as the feature space size increases.

However, the picture differs substantially for *tightly constrained SVM* (100% of bounded SV as illustrated in Figure 8.2 (a)) which are clearly affected by *several* performance drops. We may explain the many drops by the change in feature space (compared to our previous experiments). Here, we use the `RCV1` dataset, which features were transformed by *tfidf* weighing scheme as explained in [Lew04], whereas in our previous experiments, we used the raw frequencies.

In addition, by processing the features, the `RCV1` classification task differs significantly from `C01-C21` and `20ng` because of the change in geometry of the space. Accordingly, as C values nearing zero can control the width of the geometrical margin, the range of C values that conducts to *tightly constrained SVM* is influenced.

Influence of misclassified data on the performance drop To assess how misclassified data influences the performance drop, we duplicated 740 of the positive documents in the `RCV1` dataset and then, we labelled them as negative; this results in about 7% of intentionally misclassified and overlapping documents in the negative class. We report our results in Figures 8.3 (a) and (c), where (c) describes the performance on the data with misclassified documents.

In Figure 8.3 (c) the contour lines drop till .85 for 16384 features with $C = 10^6$, whereas in (a) the performance only drops till .95, the drop is more extreme in (c) than in (a). Therefore, our experiment confirms that for those feature spaces, along with large C values, the amount of misclassified or unseparable (e.g. overlapping) training points controls the importance of the performance drop. This also shows the *outlying* nature of the bounded SV residual in the SVM solutions illustrated in Figure 8.2 (a), (b) and (d) (`C01-C21` and `RCV1`).

Stability of SVM solutions In Figures 8.3 (b) and (d) we report the 10-fold cross-validated standard deviation of the performance for different C values and different feature space sizes.

First, we remark that *small feature space SVM's* is generally much less stable than *large feature space SVM's*. Further, as classification noise is added to the

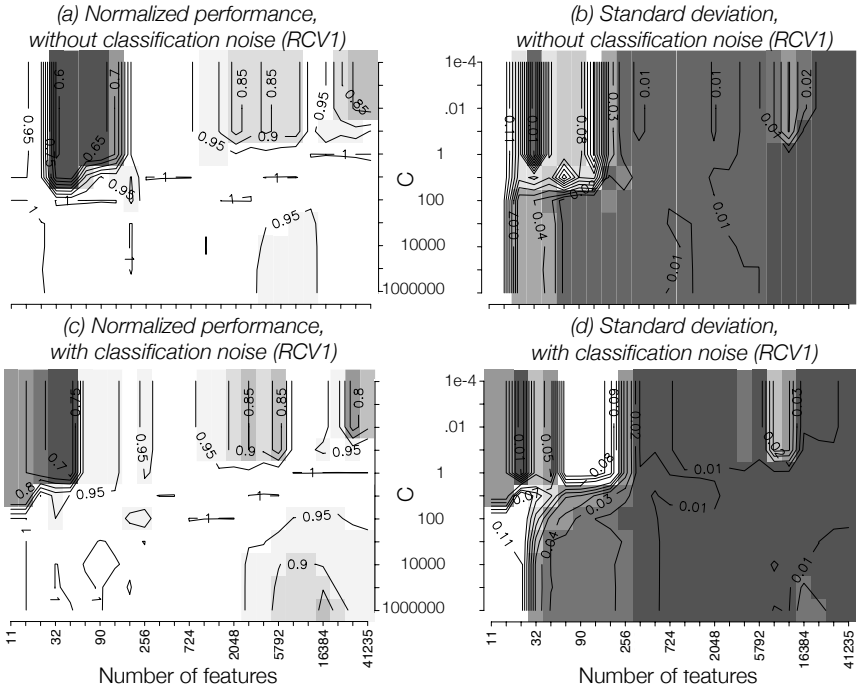

Figure 8.3: *In (c) and (d) experiments are the same as in (a) and (b) but with 740 positive documents duplicated into the negative class. As in Figure 8.1 (c), (a) and (c) detail SVM normalised performance whereas (b) and (d) report the performance standard deviation. Here, we could reproduce SVM performance drop and in view of (a) and (c), we further relate its importance to the level of classification noise in the data. Yet,* several *other performance drop occur for* tightly constrained SVM. *They may relate to the feature space geometry as here with* `RCV1`*, we operate in a* tfidf*-transformed space. Finally, (b) and (d) show that* small feature space SVM *are consistently less stable than* larger ones.

data in 8.3 (d), *small feature space SVM's* (11-256 features) are consistently less stable for all C values. Furthermore, in accordance to the several performance drops observed for small C values (illustrated in Figures 8.3 (a) and (c)), the *tightly constrained SVM's* can be less stable at identical feature space size than solutions with larger C.

8.7 Related work

In [Cri00], it is mentioned that *although the maximal margin classifier does not attempt to control the number of support vectors, [...] in practice there are frequently very few support vectors*. However, in this empirical study, we illustrate that SVM solutions are not sparse in large feature spaces. A similar observation in text classifiction was made in [Rou06] where it was observed that more than 75% of the dual variables were non-zero ($\alpha_i > 0$). In [Bur99] and [Rif99], SVM solution properties are analysed in terms of uniqueness and of degeneracy. In particular, the conditions for which all dual variables are at the bound are described, they can be referred to as "degenerate". In our experiments, we also gave experimental settings for which every training point is a bounded SV and we explained this gives trivial SVM solutions, i.e. *nearest mean classifier*. Furthermore, the study of [Mla04] raises the question whether sparsity of the feature space is a more reliable parameter than the number of features in predicting the performance of a classifier. Our experiments confirmed the specificity of the *bag of words* and its discrete nature.

8.8 Concluding remarks

Based on experiments, we systematically described the nature of SVM solutions in text classification problems when the number of training documents, the number of features and the SVM constraint parameter C are varying. In order to study SVM performance on equal grounds with other classification methods (e.g. the k nearest neighbors classifier and naive Bayes), training data was balanced and only binary classification tasks were considered.

In our experiments, we saw that SVM is consistently subject to a performance drop for particular combination of feature space size and number of training documents. This performance drop especially occurs when large C values are taken but we also observed *several* drops for small C values as we operated SVM on a classification task whose feature space was *tfidf*-transformed.

Further, we showed that this drop is a consequence of misclassified and overlapping data in the training set. In fact, as the SVM optimizer is unable to well-classify these "confusing" training points, their Lagrange multipliers (α_i) run-away until they reach the soft-margin upper bound C. Yet, although those points are useless for classification (because they overlap, they belong to distinct

classes and they counter-balance) they influence largely the final solution by having very large α_i values.

As in previous experiments we reproduced on a number of tasks a drop, it is very likely that those SVM solutions with large C values have a number of bounded SV that confuse the classification. We foresee two complementary ways to address this issue. First, the data should be cleaned before training the SVM and second, the SVM should implement a means to reduce the influence of the documents from the classification noise in the final solution.

Finally, SVM is often presented as an outstanding method in text classification; it is expected to find *sparse solutions* which would enable to classify quickly large amount of test documents at run time since SVM execution time depends on its sparsity. Yet, in our experiments, the sparsity of the solution reduces as larger feature space sizes are taken, which somewhat contradicts the common belief that all features should be used with SVM.

Conclusions

Conclusions

In this thesis, we presented two data mining *scenarios*: one for *subtyping* and one for the *comparison of algorithms*.

The rest of this conclusion chapter is structured as follows. For each *scenario*, we summarize our conclusions chapter by chapter, then we give some general conclusions, and finally we identify future work.

Subtype Discovery by Cluster Analysis

In the first part of this thesis, we presented a data mining scenario to identify *homogeneous subtypes in data by cluster analysis*. We applied this *scenario* to medical research on Osteoarthritis (OA) and Parkinson's disease (PD) and to a dataset in drug discovery. For each application, we illustrated our scenario with subtyping results. Besides, as we aimed for reproducible *subtyping* analyses and as we wanted to abstract from the application areas, we implemented our scenario as the R SubtypeDiscovery package.

In the following, we summarize our conclusions chapter by chapter.

Chapter 1 We presented the three application areas for our first scenario: medical research on OA and PD and drug discovery. We briefly described the domains, we motivated why subtyping is interesting and we gave details about the datasets used later in the thesis.

Chapter 2 We described our data mining scenario to facilitate and enhance the search for homogeneous *subtypes* in data. This scenario involves techniques to prepare data, an approach that repeatedly models data in order to select for a number of subtypes and the type of model, along with methods to characterize, compare and evaluate the most likely models. In particular, the design of this

scenario aimed at infering subtypes from data using statistical, machine learning and visualization techniques.

Chapter 3 We were concerned with two issues: *how to deal with the time dimension in the data?* and *how to assess the reliability of the discovered subtypes?* Indeed, if we do not appropriately prepare the data, identified clusters will mostly model the time dimension in the data. For this purpose, data must be adjusted for the effect of the time. Yet, the adjustment can be done in a number of ways. For this reason, we proposed to select for an adjustment on the basis of the cluster results *reliability*.

First, our experiments showed that both the type of time adjustment and the numerical type of the data (a five points scoring system for OA and mixed-scales for PD) have an influence on the *reliability* (consistency) of the subtyping. Second, we found that, in the set of possibilities we considered, a logarithm of the age in OA and a square root of the disease duration in PD gave the most *reliable* subtyping results.

Chapter 4 We described results obtained using our subtyping scenario. For OA research, we presented OA subtypes and we further assessed them for familial aggregation. For PD research, we reported on subtypes of PD and inferences made by the neurologists of the LUMC. In drug discovery, we applied our scenario to a public chemoinformatics dataset.

Chapter 5 To enable *reproducibility* of the *subtyping* analyses and to abstract from the applications we have done up to now, we made our data mining scenario available as the R SubtypeDiscovery package. In this chapter, we presented its implementation. Packaging the scenario in R helps these users, in particular from biology, to conduct analyses on their own.

Conclusions for chapters 1 to 5 In subtyping, we paid a special attention to the logical sequence of steps used to infer patterns from the data. The focus was on the *validity* of the discovered subtypes, on their *reliability* and on their *clinical relevance* in the case of OA and PD. We considered subtypes defined in terms of a Gaussian model and a subtype-selection based on a repeated modeling of the data. Further, we assessed the subtypes for their *reliability* because subtypes should show consistency as data changed slightly. Finally, we evaluated subtypes for their *clinical relevance* in terms of the familial aggregation for OA or the reproducibility of the subtyping from year one to year two for PD.

Eventually, our subtyping scenario will enhance the search for homogeneous subtypes in data; in PD research, the first subtyping results are submitted for publication [Roo08a; Roo08b]. By implementing our scenario as an R package, we abstracted from the applications we did up to now, and we enabled faster and more diverse subtyping analyses.

Future work in subtyping We identified a number of ways to improve the *subtyping* scenario. First, there are several software engineering issues. To improve the robustness of the software, the existing machine learning interface (MLInterface) of R should be used when conducting experiments on the reproducibility of the subtypes and their generalization to the total population of patients. Besides, we should separate the report generation from the data processing and for this purpose the Sweave R package could be used. In addition, relying more consistently on the object-orientation of R would improve the robustness of our package.

Concerning our subtyping results, the application in drug discovery showed an additional challenge as the variables are highly correlated. For this application, we performed the subtyping analysis on the scores of the first few principal components in order to address the correlation problem. We should conduct additional analyses in order to gain confidence in interpreting this type of subtyping analyses. Finally, to further validate our *subtyping scenario*, we want to apply it to more problem domains.

Automatic Text Classification

In the second part of this thesis, we presented a scenario for the *comparison of algorithms* in text classification. We first discussed and described a set of considerations to conduct *fair* comparative experiments. Then, we reported on experiments where we compared the behaviors of algorithms in text classification. We focused on SVM in order to develop a better understanding of its behavior in text classification.

In the following, we summarize our conclusions chapter by chapter.

Chapter 6 We introduced the problem of classifying text documents into categories and then we discussed fairness issues for comparing algorithms. Next, we described our scenario that aims to compare as fairly as possible algorithms in text classification. It should help us to better understand the problem of classifying text documents in categories.

Chapter 7 We investigated whether we should always opt for the SVM, discarding classification algorithms like the k nearest neighbors or the naive Bayes.

Our results show that all the classifiers achieved comparable performance on most problems. It is surprising that SVM is not a clear winner, despite its quite good overall performance. If a suitable preprocessing was used with the k nearest neighbors, it achieved good results and did scale-up well with the number of documents; this was not the case for the SVM. Naive Bayes also achieved good performance.

Chapter 8 We systematically described the nature of SVM solutions in text classification, when the number of training documents, the number of features and the SVM constraint parameter (C) are varying.

We showed that the SVM was subject to a performance drop for particular combination of feature space size and number of training documents. Further, we remarked that this drop especially occurs for large C values (the classification task used raw frequencies). Yet, on another classification task (*tfidf*-transformed), we also observed *several* performance drops for small C values.

The performance drop with large C is due to the occurence of overlapping data points in the training set that is, points that belong to two or more classes. In that regard, additional experiments showed that the performance drop was getting worse as more overlapping training points were artificially introduced in the classification task. Hence, either this points should be discarded from the computation by *data cleaning*, or the SVM optimizer should discover and handle these points.

Finally, we also commented on the sparsity of the SVM solutions in large *bag of words* feature space. Sparsity is a desirable property in industrial applications because of the shorter test time. Yet, we remarked that, as we involved more features in the classification problem, the sparsity of the solutions reduced. This contradicts the common understanding that all features should be used with the SVM in text classification.

Conclusions for chapters 6, 7 and 8 In text classification, the design of our scenario stemmed from our desire to develop a better understanding on typical classification tasks. This lead to a data mining scenario that aimed for more *fair* comparative experiments between algorithms than what was done up to now.

First, we focused our scenario on the binary classification problems because we recognized that the multi-class aggregating algorithm could influence greatly the comparison of algorithms. Besides, we also evaluated classifiers on training and test sets, which we sampled in *strata* taken equally from each class. Finally, we analyzed a selection of classifiers on a wide range of classification tasks and experimental settings.

To conclude, our *comparison of algorithms* data mining scenario offers a new view on the problem of classifying text documents into categories. This focus enabled to show that SVM was not systematically outperforming regular classifiers like the naive Bayes and the k nearest neighbors. Further and most importantly, we also showed that the SVM was consistently subject to performance deterioration for particular combination of number of features and documents. This performance drop can be due to classification noise in the data; SVM solutions tend to give a to great importance to the documents from the classification noise. In comparison, the naive Bayes and the k nearest neighbors do not exhibit such a performance drop. Besides, these two classifiers are simple, well-understood and fast. Therefore, the question whether traditional algorithms should still be used

in text classification can be answered with yes and especially in larger industrial applications.

Future work in comparison of algorithms The most interesting result in our opinion is that the performance of the SVM is affected by the presence of classification noise in the data. We see two complementary ways to address this issue. First, the data should be cleaned before training the SVM. Second, the SVM should implement a mean to reduce the influence of the documents from the classification noise in the final solution. Finally, we also noted the possible influence of the feature space transformation on the occurence of the performance drop; further investigations are needed.

Appendix A

Two Dimensional Molecular Descriptors

In Tables A.1, A.2, A.3, A.4, A.5, A.6, we list and give a description of the different molecular properties on which we performed our SubtypeDiscovery analyses in the chemoinformatics domain.

Table A.1: *Atom and bond counts (ABC).*

a_aro	Number of aromatic atoms.
a_count	Number of atoms (including implicit hydrogens). This is calculated as the sum of $(1 + h_i)$ over all non-trivial atoms i.
a_heavy	Number of heavy atoms $\#Zi \vert Zi > 1$
a_IC	Atom information content (total). This is a_ICM times n (as defined in the definition of a_ICM).
a_ICM	Atom information content (mean). This is the entropy of the element distribution in the molecule (including implicit hydrogens but not lone pair pseudo-atoms). Let n_i be the number of occurrences of atomic number i in the molecule. Let $p_i = n_i/n$ where n is the sum of the n_i. The value of a_ICM is the negative of the sum over all i of $p_i log p_i$.
a_nB	Number of boron atoms: $\#Zi \vert Zi = 5$
a_nBr	Number of bromine atoms: $\#Zi \vert Zi = 35$
a_nC	Number of carbon atoms: $\#Zi \vert Zi = 6$
a_nCl	Number of chlorine atoms: $\#Zi \vert Zi = 17$
a_nF	Number of fluorine atoms: $\#Zi \vert Zi = 9$

a_nH	Number of hydrogen atoms (including implicit hydrogens). This is calculated as the sum of h_i over all non-trivial atoms i plus the number of non-trivial hydrogen atoms.
a_nI	Number of iodine atoms: $\#Zi \vert Zi = 53$
a_nN	Number of nitrogen atoms: $\#Zi \vert Zi = 7$
a_nO	Number of oxygen atoms: $\#Zi \vert Zi = 8$
a_nP	Number of phosphorus atoms: $\#Zi \vert Zi = 15$
a_nS	Number of sulfur atoms: $\#Zi \vert Zi = 16$
b_1rotN	Number of rotatable single bonds. A bond is rotatable if it is not in a ring, and neither atom of the bond is such that $(d_i + h_i) < 2$.
b_1rotR	Fraction of rotatable single bonds: b_1rotN divided by b_count.
b_ar	Number of aromatic bonds.
b_count	Number of bonds (including implicit hydrogens). This is calculated as the sum of $(d_i/2 + h_i)$ over all non-trivial atoms i.
b_double	Number of double bonds. Aromatic bonds are not considered to be double bonds.
b_heavy	Number of bonds between heavy atoms.
b_rotN	Number of rotatable bonds. A bond is rotatable if it is not in a ring, and neither atom of the bond is such that $(d_i + h_i) < 2$.
b_rotR	Fraction of rotatable bonds: b_rotN divided by b_count.
b_single	Number of single bonds (including implicit hydrogens). Aromatic bonds are not considered to be single bonds.
b_triple	Number of triple bonds. Aromatic bonds are not considered to be triple bonds.
chiral	The number of chiral centers.
chiral_u	The number of unconstrained chiral centers.
lip_acc	The number of O and N atoms.
lip_don	The number of OH and NH atoms.
lip_druglike	One if and only if lip_violation < 2 otherwise zero.
lip_violation	The number of violations of Lipinski's Rule of Five.
nmol	The number of molecules (connected components).
opr_brigid	The number of rigid bonds bonds.
opr_leadlike	One if and only if opr_violation ¡ 2 otherwise zero.
opr_nring	The number of rings bonds.
opr_nrot	The number of rotatable bonds.
opr_violation	The number of violations of Oprea's lead-like test.
rings	The number of rings.

VAdjEq	Vertex adjacency information (equality): $-(1-f)log_2(1-f) - flog_2f$ where $f = (n^2-m)/n^2$, n is the number of heavy atoms and m is the number of heavy-heavy bonds. If f is not in the open interval (0,1), then 0 is returned.
VAdjMa	Vertex adjacency information (magnitude): $1 + log_2m$ where m is the number of heavy-heavy bonds. If m is zero, then zero is returned.
VDistEq	If m is the sum of the distance matrix entries then VdistEq is defined to be the sum of $log_2m - p_ilog_2p_i/m$ where p_i is the number of distance matrix entries equal to i.
VDistMa	If m is the sum of the distance matrix entries then VDistMa is defined to be the sum of $log_2m - D_{ij}log_2D_{ij}/m$ over all i and j.

Table A.2: *Adjacency and distance matrix descriptors (ADDM).*

balabanJ	Balaban's connectivity topological index.
diameter	Largest value in the distance matrix.
petitjean	Value of (diameter - radius) / diameter.
petitjeanSC	Petitjean graph Shape Coefficient: (diameter - radius) / radius.
radius	If r_i is the largest matrix entry in row i of the distance matrix D, then the radius is defined as the smallest of the r_i.
weinerPath	Wiener path number: half the sum of all the distance matrix entries.
weinerPol	Wiener polarity number: half the sum of all the distance matrix entries with a value of 3.

Table A.3: *Kier and Hall connectivity and kappa shape indices (KH).*

KierFlex	Kier molecular flexibility index: $(KierA1)(KierA2)/n$
zagreb	Zagreb index: the sum of d_i^2 over all heavy atoms i.

Table A.4: *Partial charge descriptors (PCD).*

Q_PC.	Total positive partial charge: the sum of the positive q_i. Q_PC+ is identical to PC+ which has been retained for compatibility.
Q_PC..1	Total negative partial charge: the sum of the negative q_i. Q_PC- is identical to PC- which has been retained for compatibility.
Q_RPC.	Relative positive partial charge: the largest positive q_i divided by the sum of the positive q_i. Q_RPC+ is identical to RPC+ which has been retained for compatibility.
Q_RPC..1	Relative negative partial charge: the smallest negative q_i divided by the sum of the negative q_i. Q_RPC- is identical to RPC- which has been retained for compatibility.
Q_VSA_FHYD	Fractional hydrophobic van der Waals surface area. This is the sum of the v_i such that $\|q_i\|$ is less than or equal to 0.2 divided by the total surface area. The vi are calculated using a connection table approximation.
Q_VSA_FNEG	Fractional negative van der Waals surface area. This is the sum of the v_i such that q_i is negative divided by the total surface area. The v_i are calculated using a connection table approximation.
Q_VSA_FPNEG	Fractional negative polar van der Waals surface area. This is the sum of the v_i such that q_i is less than -0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
Q_VSA_FPOL	Fractional polar van der Waals surface area. This is the sum of the v_i such that $\|qi\|$ is greater than 0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.
Q_VSA_FPOS	Fractional positive van der Waals surface area. This is the sum of the v_i such that q_i is non-negative divided by the total surface area. The v_i are calculated using a connection table approximation.
Q_VSA_FPPOS	Fractional positive polar van der Waals surface area. This is the sum of the v_i such that q_i is greater than 0.2 divided by the total surface area. The v_i are calculated using a connection table approximation.

Q_VSA_HYD	Total hydrophobic van der Waals surface area. This is the sum of the v_i such that $\|q_i\|$ is less than or equal to 0.2. The v_i are calculated using a connection table approximation.
Q_VSA_NEG	Total negative van der Waals surface area. This is the sum of the v_i such that q_i is negative. The v_i are calculated using a connection table approximation.
Q_VSA_PNEG	Total negative polar van der Waals surface area. This is the sum of the v_i such that q_i is less than -0.2. The v_i are calculated using a connection table approximation.
Q_VSA_POL	Total polar van der Waals surface area. This is the sum of the v_i such that $\|q_i\|$ is greater than 0.2. The v_i are calculated using a connection table approximation.
Q_VSA_POS	Total positive van der Waals surface area. This is the sum of the v_i such that q_i is non-negative. The v_i are calculated using a connection table approximation.
Q_VSA_PPOS	Total positive polar van der Waals surface area. This is the sum of the v_i such that q_i is greater than 0.2. The v_i are calculated using a connection table approximation.

Table A.5: *Pharmacophore feature descriptors (PFD).*

a_acc	Number of hydrogen bond acceptor atoms (not counting acidic atoms but counting atoms that are both hydrogen bond donors and acceptors such as -OH).
a_acid	Number of acidic atoms.
a_base	Number of basic atoms.
a_don	Number of hydrogen bond donor atoms (not counting basic atoms but counting atoms that are both hydrogen bond donors and acceptors such as -OH).
a_hyd	Number of hydrophobic atoms.
vsa_acc	Approximation to the sum of VDW surface areas of pure hydrogen bond acceptors (not counting acidic atoms and atoms that are both hydrogen bond donors and acceptors such as -OH).
vsa_acid	Approximation to the sum of VDW surface areas of acidic atoms.
vsa_base	Approximation to the sum of VDW surface areas of basic atoms.
vsa_don	Approximation to the sum of VDW surface areas of pure hydrogen bond donors (not counting basic atoms and atoms that are both hydrogen bond donors and acceptors such as -OH).
vsa_hyd	Approximation to the sum of VDW surface areas of hydrophobic atoms.
vsa_other	Approximation to the sum of VDW surface areas of atoms typed as "other".
vsa_pol	Approximation to the sum of VDW surface areas of polar (both hydrogen bond donors and acceptors) atoms (such as -OH).

Table A.6: *Physical properties (PP).*

apol	Sum of the atomic polarizabilities (including implicit hydrogens) with polarizabilities.
bpol	Sum of the absolute value of the difference between atomic polarizabilities of all bonded atoms in the molecule (including implicit hydrogens) with polarizabilities.
density	Molecular mass density: Weight divided by vdw_vol.
FCharge	Total charge of the molecule (sum of formal charges).
logP.o.w.	Log of the octanol/water partition coefficient (including implicit hydrogens). This property is calculated from a linear atom type model with $r^2 = 0.931$, $RMSE = 0.393$ on 1,847 molecules.
logS	Log of the aqueous solubility (mol/L). This property is calculated from an atom contribution linear atom type model with $r^2 = 0.90$, 1,200 molecules.
mr	Molecular refractivity (including implicit hydrogens). This property is calculated from an 11 descriptor linear model with $r^2 = 0.997$, $RMSE = 0.168$ on 1,947 small molecules.
reactive	
SlogP	Log of the octanol/water partition coefficient (including implicit hydrogens). This property is an atomic contribution model that calculates logP from the given structure; i.e., the correct protonation state (washed structures). Results may vary from the $logP(o/w)$ descriptor. The training set for SlogP was 7000 structures.
SMR	Molecular refractivity (including implicit hydrogens). This property is an atomic contribution model that assumes the correct protonation state (washed structures). The model was trained on 7000 structures and results may vary from the mr descriptor.
TPSA	Polar surface area (A^2) calculated using group contributions to approximate the polar surface area from connection table information only. The parameterization is that of Ertl et al.
vdw_area	Area of van der Waals surface calculated using a connection table approximation.
vdw_vol	van der Waals volume calculated using a connection table approximation.
Weight	Molecular weight (including implicit hydrogens) with atomic weights.

Appendix B

Additional Results in Text Classification

For the nearest neighbors algorithm, a number of feature space transformations is possible. The k nearest neighbors classifier implemented in the `libbow` library [McC96] defines these transformations by two sets of three letters; for a detailed description of the letters, see the Table B.1. In Figures B.1, B.2 and B.3 we report histograms of the count of pairwise wins for each combination of the feature space transformations.

We remark that binary transformations (`b__`) tend to perform worse. As well, the inverse document frequency (`_t_`) does not show as crucial as we could expect given its wide use in information retrieval. Further, normalizing the scores (`__c`) did not show any improvement. The rest of the transformations seem to perform equally well to the exception of the `_tc`-transformation. Then, as a `_tc`-transformation are applied on the training set, we observe generally underperforming nearest neighbor classifiers; a possible explanation would be a software-issue while normalizing the scores of the training set. In our analyses, we avoided this type of transformations.

Table B.1: *The feature space transformations are defined in the* `libbow` *library by combinations of three letters that refer to the term frequency, the inverse document frequency and the normalization. Recall that x_{ij} is the frequency of the word j in the document i. This Table summarizes the different combinations.*

		Term frequency (*tf*)	
n	*none*	Raw frequencies	$tf(x_{ij}) = x_{ij}$
b	*binary*	Binarize the raw frequencies	$tf(x_{ij}) = \begin{cases} 1 & \text{if } tf(x_{ij}) \geq 1 \\ 0 & \text{otherwise} \end{cases}$
m	*max-norm*	Normalize x_{ij} relatively to the maximum term frequency observed in a document i	$tf(x_{ij}) = \frac{x_{ij}}{max_j x_{ij}}$
a	*augmented norm*	Similar to the *max-norm* but with $\frac{1}{2}$ added	$tf(x_{ij}) = \frac{1}{2} + \frac{x_{ij}}{2max_i x_{ij}}$
l	*log*	Logarithm of the term frequency	$tf(x_{ij}) = 1 + log(x_{ij})$
		Inverse document frequency (*idf*)	
n	*none*	*idf* is not used	$idf(x_{ij}) = 1$
t	*idf*	Inverse of the frequency of the term x_{ij} in the database which has N documents	$idf(x_{ij}) = log\left(\frac{N}{df(x_{ij})}\right)$
		Normalization	
n	*none*	Normalization is not used	$\phi(x_{ij}) = tf(x_{ij})idf(x_{ij})$
c	*cosine*	Apply a cosine normalization	$\phi(x_{ij}) = \sqrt{\frac{tf(x_{ij})idf(x_{ij})}{\Sigma_j (tf(x_{ij})idf(x_{ij}))^2}}$

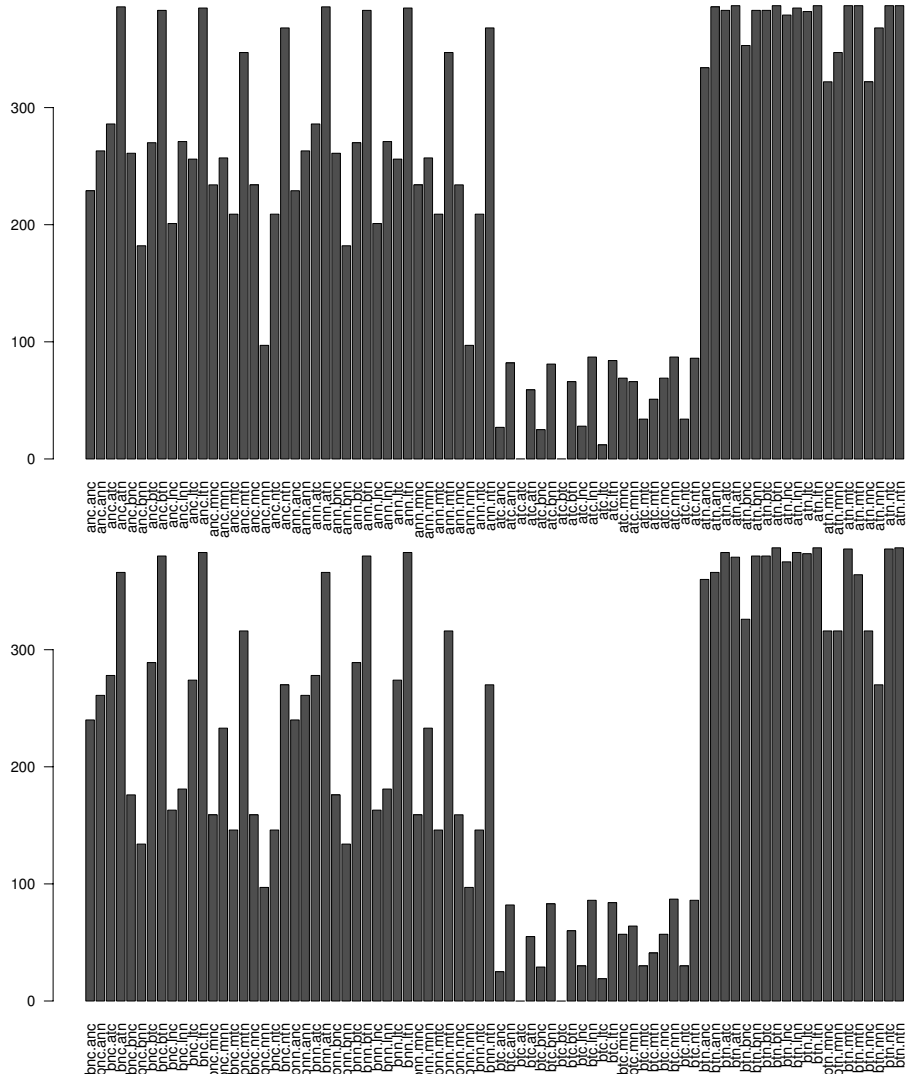

Figure B.1: *Counts of pairwise wins for each transformation, from* `ann.anc` *to* `btn.ntn`.

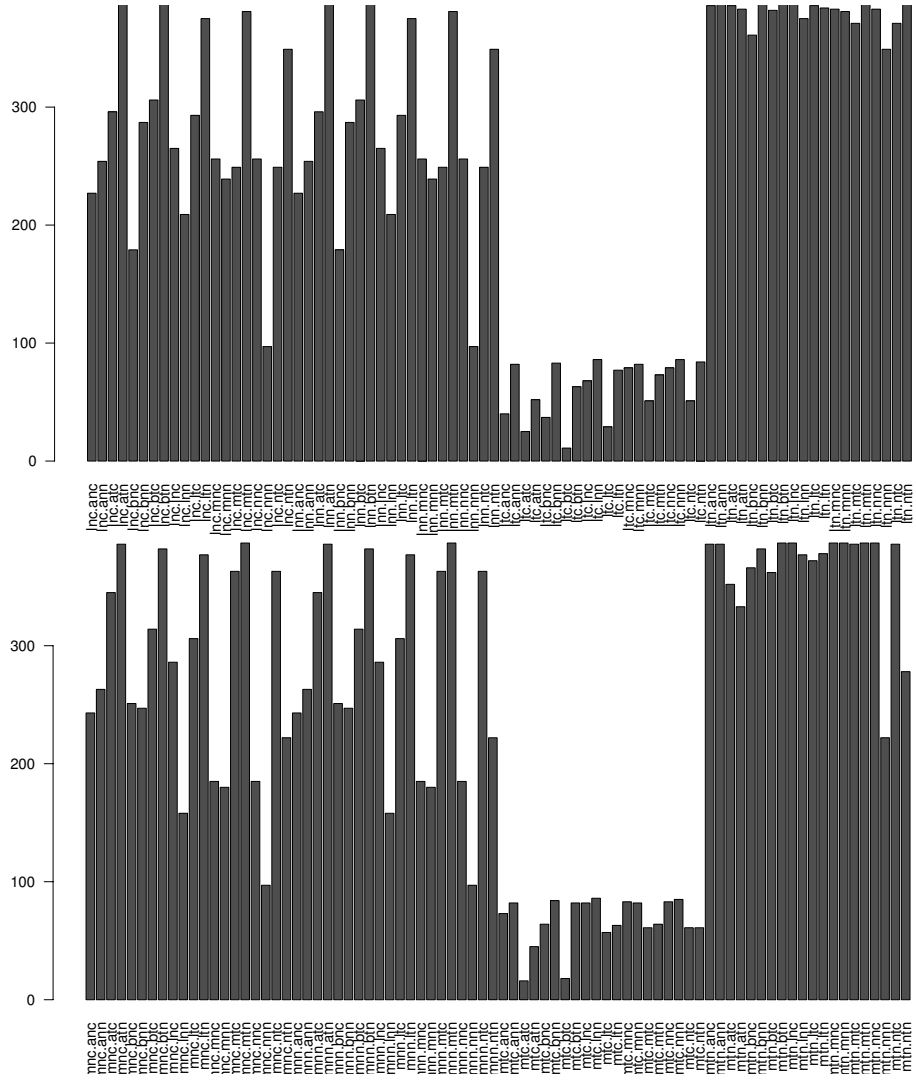

Figure B.2: *Counts of pairwise wins for each transformation, from* `lnc.anc` *to* `mtn.ntn`.

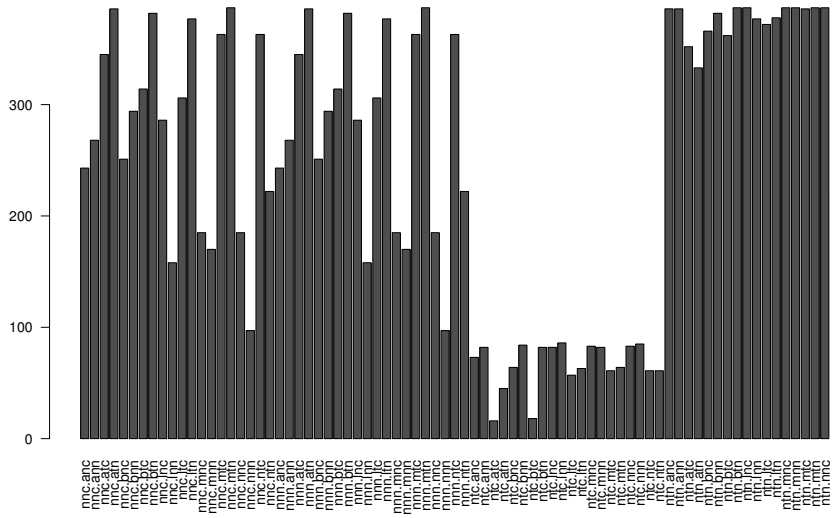

Figure B.3: *Counts of pairwise wins for each transformation, from* `nnc.anc` *to* `ntn.ntn`.

Bibliography

[Ban93] J. D. Banfield and A. E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, vol. 49:pp. 803–821, 1993.

[Bey99] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? *Lecture Notes in Computer Science*, vol. 1540:pp. 217–235, 1999.

[Bre94] C. A. Brewer. *Color Use Guidelines for Mapping and Visualization*, chap. 7, pp. 123–147. Elsevier Science, Tarrytown, NY, 1994.

[Bur99] C. Burges and D. Crisp. Uniqueness of the svm solution. In *Proceedings of the 12th Conference on Neural Information Processing Systems.*, pp. 223–229. MIT Press, Cambridge, MA, 1999.

[Can06a] E. O. Cannon, A. Bender, D. S. Palmer, and J. B. O. Mitchell. Chemoinformatics-based classification of prohibited substances employed for doping in sport. *Journal of Chemical Information and Modeling*, vol. 46(6):pp. 2369–2380, 2006.

[Can06b] E. O. Cannon and J. B. O. Mitchell. Classifying the world anti-doping agency's 2005 prohibited list using the chemistry development kit fingerprint. In *CompLife*, pp. 173–182. 2006.

[Can08] E. O. Cannon, F. Nigsch, and J. B. O. Mitchell. A novel hybrid ultrafast shape descriptor method for use in virtual screening. *Chemistry Central Journal*, vol. 2(3), 2008.

[CCGI08] C. Chemical Computing Group Inc., Montreal. Molecular operating environment (moe). `http://www.chemcomp.com`, 2008.

[Cha01] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

[cnr85] *Trésor de la Langue Française, Dictionnaire de la langue du XIXe et du XXe siècle*. Centre National de la Recherche Scientifique, 1985.

[Col05] F. Colas. *Propositions de Recommandations en Classification de Documents (Giving Advice for Document Classification Tasks)*. Master's thesis, ESIEA Electrical Engineering Institute (Laval) and Lyon II University (Lyon), France, September 2005.

[Col06a] F. Colas and P. Brazdil. Comparison of svm and some older classification algorithms in text classification tasks. In *Proceedings of the IFIP-AI 2006 World Computer Congress, Santiago de Chile, Chile*, IFIP 217, pp. 169–178. Springer, August 2006.

[Col06b] F. Colas and P. Brazdil. On the behavior of svm and some older algorithms in binary text classification tasks. In *Proceedings of Text Speech and Dialogue (TSD2006), Brno, Czech Republic*, Lecture Notes in Computer Science 4188, pp. 45–52. Springer, September 2006.

[Col07a] F. Colas, I. Meulenbelt, J. J. Houwing-Duistermaat, P. E. Slagboom, and J. N. Kok. A comparison of two methods for finding groups using heat maps and model based clustering. In *Proceedings of the 27th Annual International Conference of the British Computer Society (SGAI), AI-2007, Cambridge, UK*, pp. 119–131. December 2007.

[Col07b] F. Colas, P. Paclík, J. N. Kok, and P. Brazdil. Does svm really scale up to large bag of words feature spaces? In *Proceedings of Intelligent Data Analysis (IDA2007), Ljubljana, Slovenia*, Lecture Notes in Computer Science 4723, pp. 296–307. Springer, September 2007.

[Col08a] F. Colas, I. Meulenbelt, J. J. Houwing-Duistermaat, M. Kloppenburg, I. Watt, S. M. van Rooden, M. Visser, H. Marinus, J. J. van Hilten, P. E. Slagboom, and J. N. Kok. Stability of clusters for different time adjustments in complex disease research. In *Proceedings of the 30th Annual International IEEE EMBS Conference (EMBC'08), Vancouver, British Columbia, Canada*. August 2008.

[Col08b] F. Colas, I. Meulenbelt, J. J. Houwing-Duistermaat, M. Kloppenburg, I. Watt, S. M. van Rooden, M. Visser, J. Marinus, E. O. Cannon, A. Bender, J. J. van Hilten, P. E. Slagboom, and J. N. Kok. A scenario implementation in r for subtypediscovery examplified on chemoinformatics data. In T. Margaria and B. Steffen, editors, *Proceedings of Leveraging Applications of Formal Methods, Verification and Validation*, vol. 17 of *CCIS*, pp. 669–683. Springer, 2008.

[Col08c] F. Colas, S. M. van Rooden, I. Meulenbelt, J. J. Houwing-Duistermaat, A. Bender, E. O. Cannon, M. Visser, H. Marinus, J. J. van Hilten, P. E.

Slagboom, and J. N. Kok. An r package for subtype discovery exemplified on chemoinformatics data. In *Statistical and Relational Learning in Bioinformatics (StReBio ECML-PKDD'08 Workshop), Antwerp, Belgium*. September 2008.

[Cri00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel based learning methods*. Cambridge University Press, 2000.

[Dae03] W. Daelemans, V. Hoste, F. De Meulder, and B. Naudts. Combined optimization of feature selection and algorithm parameter interaction in machine learning of language. In *Proceedings of the 14th European Conference on Machine Learning (ECML-2003)*, Lecture Notes in Computer Science 2837, pp. 84–95. Springer-Verlag, 2003.

[Dav04] D. Davidov, E. Gabrilovich, and S. Markovitch. Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In *Proceedings of the 27th annual international conference on research and development in information retrieval*, pp. 250–257. 2004.

[Dem77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*, vol. 39(1):pp. 1–38, 1977.

[Die05] P. A. Dieppe and L. S. Lohmander. Pathogenesis and management of pain in osteoarthritis. *Lancet*, vol. 365:pp. 965–73, 2005.

[Dum98] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th international Conference on Information and Knowledge Management (CIKM'98)*, pp. 148–155. ACM Press, New York, US, 1998.

[Eis98] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of National Academy of Science USA*, vol. 95, pp. 11863–14868. 1998.

[Eve74] B. Everitt. *Cluster Analysis*. Social Science Research Council. Heinemann Educational Books Ltd, 1974.

[Ewe05] W. J. Ewens and G. R. Grant. *Statistical Methods in Bioinformatics, An Introduction*. Springer, 2005.

[For03] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, vol. 3:pp. 1289–1305, 2003.

[Fra98a] C. Fraley. Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, vol. 20:pp. 270–281, 1998.

[Fra98b] C. Fraley and A. E. Raftery. How many clusters? Which clustering method? - Answers via model-based cluster analysis. *Computer Journal*, vol. 41:pp. 578–588, 1998.

[Fra99] C. Fraley and A. E. Raftery. MCLUST: Software for model-based cluster analysis. *Journal of Classification*, vol. 16:pp. 297–306, 1999.

[Fra02a] C. Fraley and A. E. Raftery. MCLUST: Software for model-based clustering, density estimation, and discriminant analysis. Technical Report 415, University of Washington, Department of Statistics, October 2002.

[Fra02b] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, vol. 97:pp. 611–631, 2002.

[Fra03] C. Fraley and A. E. Raftery. Enhanced software for model-based clustering, density estimation, and discriminant analysis: MCLUST. *Journal of Classification*, vol. 20:pp. 263–286, 2003.

[Fra06] C. Fraley and A. E. Raftery. MCLUST version 3 for r: Normal mixture modeling and model-based clustering. Technical Report 504, University of Washington, Department of Statistics, September 2006.

[Für02] J. Fürnkranz. Pairwise classification as an ensemble technique. In *Proceedings of the 13th European Conference on Machine Learning (ECML-2002)*, pp. 97–110. 2002.

[Gen05] R. Gentleman, V. J. Carey, W. Huber, R. A. Irizarry, and S. Dudoit. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer Series in Statistics for Biology and Health. Springer, 2005.

[Gib88] W. R. Gibb and A. J. Lees. The relevance of the lewy body to the pathogenesis of idiopathic parkinson's disease. *Journal of Neurology, Neurosurgery, and Psychiatry*, vol. 51:pp. 745–752, 1988.

[Has01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2001.

[Het99] S. Hettich and S. D. Bay. The uci kdd archive [http://kdd.ics.uci.edu]. *Irvine, CA: University of California, Department of Information and Computer Science*, 1999.

[Ins85] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, vol. 1(2):pp. 69–91, 1985.

[Jan08] J. Jankovic. Parkinson's disease: clinical features and diagnosis. *Journal of Neurology, Neurosurgery, and Psychiatry with Practical Neurology*, vol. 79:pp. 368–376, 2008.

[Joa98] T. Joachims. Making large-scale support vector machine learning practical. In A. S. B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*, pp. 169–184. MIT Press, Cambridge, MA, 1998.

[Kas95] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, vol. 90(430):pp. 773–795, 1995.

[Kel57] J. H. Kellgren and J. S. Lawrence. Radiological assessment of osteo-arthrosis. *Annals of the Rheumatic Diseases*, vol. 16(4):pp. 494–502, December 1957.

[Kos91] V. Kostic, S. Przedborski, E. Flaster, and N. Sternic. Early development of levodopa-induced dyskinesias and response fluctuations in young-onset parkinson's disease. *Neurology*, vol. 41(202), 1991.

[Lew04] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, vol. 5:pp. 361–397, 2004.

[Liu05] T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Exploration Newsletter*, vol. 7(1):pp. 36–43, 2005.

[lon84] *Longman Dictionary of the English Language*. Longman Group Limited, 1984.

[LUM08a] LUMC. Garp: Genetics, arthrosis and progression. `http://www.lumc.nl/4060/research/garp.html`, 2008.

[LUM08b] LUMC. Scopa-Propark: Scales for outcomes in parkinson's disease. `http://www.scopa-propark.eu`, 2008.

[Mar03a] J. Marinus. *Clinimetrics in Parkinson's disease*. Ph.D. thesis, Leiden University, june 2003.

[Mar03b] J. Marinus, M. Visser, J. J. van Hilten, G. J. Lammers, and A. M. Stiggelbout. Assessment of sleep and sleepiness in parkinson disease. *Sleep*, vol. 26(8):pp. 1049–1054, 2003.

[Mar03c] J. Marinus, M. Visser, N. A. Verwey, H. A. M. Middelkoop, A. M. Stiggelbout, and J. J. van Hilten. Assessment of cognition in parkinson's disease. *Neurology*, vol. 61:pp. 1222–1228, 2003.

[Mar04] J. Marinus, M. Visser, A. M. Stiggelbout, J. Rabey, P. Martinez-Martin, U. Bonuccelli, P. Kraus, and J. J. van Hilten. A short scale for the assessment of motor impairments and disabilities in parkinson's disease: the spes/scopa. *Journal of Neurology, Neurosurgery, and Psychiatry*, vol. 75(3):pp. 288–396, 2004.

[McC96] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/~mccallum/bow`, 1996.

[McC98] A. K. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[Meu97] I. Meulenbelt. *Genetic predisposing factors of osteoarthritis*. Ph.D. thesis, Leiden University, 1997.

[Min07] J. Min. *Generalized Osteoarthritis from Mendelian Disorder to Complex Disease*. Ph.D. thesis, Leiden University, January 2007.

[Mit97] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[MJF08] MJFF. Michael j. fox foundation for parkinson's research / research programs / parkinson's disease subtypes. `http://www.michaeljfox.org`, 2008.

[Mla04] D. Mladenic, J. Brank, M. Grobelnik, and N. Milic-Frayling. Feature selection using linear classifier weights: interaction with classification models. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 234–241. ACM Press, New York, US, 2004.

[Rif99] R. Rifkin, M. Pontil, and A. Verri. A note on support vector machine degeneracy. In *Proceedings of the 10th international conference on Algorithmic Learning Theory (ALT '99), Tokyo, Japan*, vol. 1720, pp. 252–263. Springer, December 1999.

[Riy06] N. Riyazi. *Familial osteoarthritis, risk factors and determinants of outcome*. Ph.D. thesis, Leiden University, 2006.

[rla08] R*: A Language and Environment for Statistical Computing*. R Development Core Team, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[Rog02] M. Rogati and Y. Yang. High-performing feature selection for text classification. In *Proceedings of the 11th international Conference on Information and Knowledge Management (CIKM'02)*, pp. 659–661. 2002.

[Roo08a] S. M. van Rooden, F. Colas, M. Visser, D. Verbaan, J. Marinus, J. N. Kok, and J. J. van Hilten. Discovery and validation of subtypes in parkinson's disease. *Submitted for publication*, 2008.

[Roo08b] S. M. van Rooden, M. Visser, F. Colas, D. Verbaan, J. Marinus, J. N. Kok, and J. J. van Hilten. Factors and subtypes in motor impairment in parkinson's disease: a data driven approach. *Submitted for publication*, 2008.

[Rou06] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Learning hierarchical multi-category text classification models. *Journal of Machine Learning Research*, vol. 7:pp. 1601–1626, 2006.

[Sch06] P. Schönhofen and A. A. Benczúr. Exploiting extremely rare features in text categorization. In *Proceedings of the 16th European Conference on Machine Learning (ECML-2006)*, pp. 759–766. 2006.

[Sne73] P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy, The Principles and Practice of Numerical Classification*. Books in Biology. W. H. Freeman and Company, 1973.

[SPS05] SPSS Inc. *SPSS Base 14.0 for Windows User's Guide*. SPSS Inc., Chicago IL, October 2005.

[Str86] G. Strang. *Introduction to applied mathematics*. Wellesley-Cambridge Press, 1986.

[Tuf83] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connectictut Connectictut, 1983.

[Tuf90] E. R. Tufte. *Envisioning Information*. Graphics Press, Cheshire, Connectictut, 1990.

[Tuk77] J. W. Tukey. *Exploratory Data Analysis*. MA: Addison-Wesley, 1977.

[Vap95] V. N. Vapnik. *The Nature of Statistical Theory*. Information Science and Statistics. Springer-Verlag, 1995.

[Vis04] M. Visser, J. Marinus, A. M. Stiggelbout, and J. J. van Hilten. Assessment of autonomic dysfunction in parkinson's disease: The scopa-aut. *Movement Disorders*, vol. 19(11):pp. 1306–1312, 2004.

[Vis05] M. Visser. *The Assessment of the Disablement Process in Parkinson's Disease*. Ph.D. thesis, Leiden University, November 2005.

[Vis06] M. Visser, A. F. G. Leentjens, J. Marinus, A. M. Stiggelbout, and J. J. van Hilten. Reliability and validity of the beck depression inventory in patients with parkinson's disease. *Movement Disorders*, vol. 21(5):pp. 0885–3185, 2006.

[Vis07] M. Visser, D. Verbaan, S. M. van Rooden, A. M. Stiggelbout, J. Marinus, and J. J. van Hilten. Assessment of psychiatric complications in parkinson's disease: The scopa-pc. *Movement Disorders*, vol. 22(15):pp. 2221–2228, August 2007.

[Wik08] Wikipedia. Parkinson's disease. `http://en.wikipedia.org/Parkinson's_disease`, 2008.

[Yan97] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*, pp. 412–420. 1997.

[Yan99a] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, pp. 69–90, 1999.

[Yan99b] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval*, pp. 42–49. ACM Press, New York, NY, USA, 1999.

[Yan03] Y. Yang. A scalability analysis of classifiers in text categorization. In *Proceedings of the 26th ACM SIGIR conference on research and development in information retrieval*, pp. 96–103. ACM Press, New York, US: Toronto, CA, 2003.

[Zha01] T. Zhang and F. J. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, vol. 4(1):pp. 5–31, 2001.

Samenvatting

Een *data mining scenario* is een logische opeenvolging van stappen om uit gegevens patronen af te leiden. In dit proefschrift presenteren wij twee data mining scenarios: één voor *subtype ontdekking door cluster analyse* en één voor de vergelijking van algoritmen voor *automatische text classificatie*.

Het eerste hoofdstuk (*Introduction*) bevat een algemene inleiding over data mining scenarios alsmede de onderzoekdoelstellingen binnen elk toepassingsgebied.

In het eerste deel van deze proefschrift presenteren wij een scenario ontwikkeld in samenwerking met onderzoekers van het Leidsch Universitair Medisch Centrum om subtypes van ziekten te identificeren die door klinische ongelijksoortigheid worden gekenmerkt. In het bijzonder kijken wij naar subtypes in patiëntencohorten die door Osteoarthritis (OA) en ziekte van Parkinson (PD, Parkinson's disease) worden getroffen. Een gevoeligere classificatie kan aan het onderzoek naar het onderliggende ziektemechanisme bijdragen.

Naast klinish onderzoek naar OA en PD, probeerden wij ons subtyping scenario ook uit in een chemische gegevensbestand om het verband tussen verschillende bioactiviteitklassen van molecules te begrijpen. Subtyping kan het begrip van gelijkheid (en afstand) tussen verschillende entiteiten verbeteren.

In hoofdstuk 1 (*Application Domains*), presenteren wij de drie verschillende toepassingsgebieden van ons subtyping scenario: Osteoarthritis, de ziekte van Parkinson en drug discovery.

In hoofdstuk 2 (*A Scenario for Subtype Discovery by Cluster Analysis*), presenteren wij ons scenario dat verschillende technieken omvat om de gegevens te verwerken, onze benadering die het aantal van subtypes en het soort van model selecteert, en extra methodes om de waarschijnlijkste modellen te karakterisen, te vergelijken en te evalueren. Het scenario groepeert niet alleen gegevens maar het

geeft ook een reeks resultaten om een analyse van de gevonden subtypes mogelijk te maken.

In hoofdstuk 3 (*Reliability of Cluster Results for Different Types of Time Adjustments*), bekijken wij twee belangrijke punten van de clusteranalyse. Hoe de tijd te behandelen? En hoe betrouwbaar zijn de clusterresultaten? Zowel leeftijd voor OA en de ziekteduur voor PD, kunnen aan de veranderlijkheid van de gegevens bijdragen. Wij zoeken naar de meest aangewezen manier om hiervoor te corrigeren. Dan, strevend naar betrouwbare clusteranalyses, zouden de clusterresultaten *robuustheid* met betrekking tot kleine veranderingen in de gegevens moeten laten zien.

In hoofdstuk 4 (*Subtyping in Osteoarthritis, Parkinson's disease and Drug Discovery*), geven wij de experimentele resultaten voor OA, PD en drug discovery.

In hoofdstuk 5 (*Scenario Implementation as the R SubtypeDiscovery Package*), presenteren wij de structuur van ons softwarepakket.

In de tweede gedeelde van ons proefschrift presenteren wij een tweede data mining scenario. Het doel was een grote studie uit te voeren om ons begrip van automatische tekstclassificatie te verbeteren.

In hoofdstuk 6 (*A Scenario for the Comparison of Algorithms in Text Classification*), geven wij de experimentele methodologie van ons data mining scenario.

Dan, hoofdstuk 7 (*Comparison of Classifiers*), rapporteert dat alle classificatiealgoritmen vergelijkbare prestaties hebben op de meeste problemen. SVM is geen duidelijke winnaar ondanks zijn goede algemene prestaties, k nearest neighbors bereikt zeer goede resultaten en naive Bayes presteert goed. Met de juiste voorbewerking, schaalt k nearest neighbors met het aantal documenten. Dit is niet het geval voor SVM. In de experimenten merkten wij sommige patronen op waarvoor wij geen verklaring konden verstrekken. In het bijzonder, wij zagen een prestatiedaling van SVM op verscheidene classificatietaken.

In hoofdstuk 8 (*Does SVM Really Scale up to Large Bag of Words Feature Spaces?*), onderzoeken wij dit verder. Wij merken in onze empirische studie op dat de beperkte oplossingen van SVM hoge uitvoerders (high performers) zijn. Maar de meeste documenten zijn dan begrensde support vector en dit maakt dat SVM zich gedraagt als een nearest mean classifier. Dit resultaat stelt vraagtekens bij de verdiensten SVM in sparse bag of words feature spaces. Bovendien laten wij zien dat SVM lijdt aan prestatieverslechtering voor bijzonder combinaties van training set size/aantal van features. Wij verklaren deze prestatieverslechtering door documenten van verschillende klassen die in de feature space overlappen.

Curriculum Vitae

Fabrice Colas was born on September 14, 1981, in Laval, France.

Fabrice conducted electrical engineering studies at the ESIEA institute[1]. As part of ERASMUS, he spent a semester in Budapest[2] and then, he went for a one year specialization in data mining research in Lyon[3]. For his final graduation project, he worked in text mining with Prof. Brazdil[4] provided the portuguese pluri-anual funding of the Fundação para a Ciencia e Tecnologia under the SUMO project. In '05, he graduated with the jury honors of the ESIEA institute and the Lyon II University, respectively with a degree of engineering and a MSc research oriented in data mining. After what he joined Prof. Kok to prepare a doctoral dissertation at the Leiden University[5] made possible by the support of the Netherlands BioInformatics Center (NBIC).

During his three years-long PhD research, he authored seven articles in peer-reviewed conferences that totalize fourteen co-authors from ten research groups. He was also a program committee member of IFIP AI '08, Milan (Italy). Additionally, Fabrice prepared and made available publicly the R SubtypeDiscovery package, a data mining scenario for subtyping that was developed in the course of his PhD. This package was presented at the R BioConductor conference '08, Seattle (USA).

This thesis summarizes his research on *Data Mining Scenarios for the Discovery of Subtypes and the Comparison of Algorithms*.

[1] '00-'05: École Supérieur d'Informatique, Électronique, Automatique, Laval, France.
[2] '04: Budapesti Műszaki és Gazdaságtudományi Egyetem, Budapest, Hungary.
[3] '04-'05: DEA ECD, Lyon II University, France.
[4] '05: Laboratorio de Inteligência Artificial e Apoio a Decisão, University of Porto, Portugal.
[5] '06-'09: Leiden Institute of Advanced Computer Science, the Netherlands.

Acknowledgements

Between my colleagues, because of their constant support over the first two years of my PhD, I am particularly grateful to my two Czech friends, Pavel Paclík and Professor Pavel Brazdil.

Since they contributed so much to my PhD, I am also very thankful to Ingrid Meulenbelt, Jeanine Houwing-Duistermaat and Eline Slagboom, Stephanie van Rooden, Martine Visser, Han Marinus and Bob van Hilten, and more recently Andreas Bender. What a challenge to work with you all!

My former colleagues of the LIAAD in Porto also played a very important role as, during my MSc thesis, I adapted to their way of conducting research. In the last years, each time I visited them, they left me with a unique feeling of availability and friendliness.

Other than my project colleagues, I wish to thank Wim Aspers and Thijs Dijk for counselling, chatting and arranging daily things at the LIACS. I also shared office with Yanju Zhang over the last two years and however distant our respective countries, I did much appreciate the understanding which underlied our regular talks, especially those not related to our jobs. I further enjoyed the cultural diversity at the LIACS having neighbour colleagues from the Netherlands, Morocco, China, Russia, Israel, Italy, etc.

Job accounting for 50% to my coming to the Netherlands, I want to thank those who, by their teaching, contributed to a today being which I feel more confortable with than before. Thanks to René and Jelena, Aharona and Pauline, Peter, Natalia.

Already unrooted for a couple of years before starting my PhD and living all that time with a single backpack, I went with some relief for a few years to the Netherlands. Being relatively close to my motherland, I enjoyed returning to my family who, from my parents to my grandma's, from my sister to my cousins and

from my uncles to my nephews, has been so welcoming and caring each time that I visited them. What a family, *merci*.

Fabrice,
July-November 2008.