NÚRIA MACIÀ ANTOLÍNEZ

DATA COMPLEXITY IN SUPERVISED LEARNING: A FAR-REACHING IMPLICATION

DATA COMPLEXITY IN SUPERVISED LEARNING: A FAR-REACHING IMPLICATION

NÚRIA MACIÀ ANTOLÍNEZ

laSalle
Universitat Ramon Llull

Ph.D. programme: IT and its management
Supervisor: Dr. Ester Bernadó Mansilla
October 2011

Núria Macià Antolínez: *Data complexity in supervised learning: A far-reaching implication* ,
IT and its management, © October 2011

Dedicated to my parents, Anna and Josep.

ABSTRACT

Machine learning techniques have a wide range of practical applications, and algorithms for supervised classification, which infer a decision boundary from a set of training instances, are at the core of fascinating uses. The diversity of domains—medicine, industry, or learning—provides extremely disparate data sets regarding properties such as the type of attributes, volume of instances, and data distribution. All of these characteristics have led to the implementation of different strategies to tackle each problem properly, since learner performance depends partly on the algorithm design. Tremendous progress has been made in refining such algorithms. Actually, the development of techniques has reached an advanced state of maturity offering thousands of methods, all of them very competitive, and providing accurate models from data which are generalised from a sample of the problem at hand. However, despite the progress in data classification, questions such as how the intrinsic characteristics of the data sets affect learners remain unanswered. This, coupled with the little leeway for improvement and the uncertainty of the ability of techniques to fully capture the underlying knowledge of data, induces us to look toward other elements involved in the learning process. At this point, data steal the limelight from learners.

This thesis takes a close view of data complexity and its role shaping the behaviour of machine learning techniques in supervised learning and explores the generation of synthetic data sets through complexity estimates. The work has been built upon four principles which have naturally followed one another. (1) A critique about the current methodologies used by the machine learning community to evaluate the performance of new learners unleashes (2) the interest for alternative estimates based on the analysis of data complexity and its study. However, both the early stage of the complexity measures and the limited availability of real-world problems for testing inspires (3) the generation of synthetic problems, which becomes the backbone of this thesis, and (4) the proposal of artificial benchmarks resembling real-world problems.

Reliable assessment? Indeed, one of the challenges in supervised learning is how to evaluate the quality of the models evolved by different machine learning techniques. Performance, in terms of accuracy, is highly data-dependent. In addition, error estimates are generally obtained by running methods over a small test bed composed of *toy* real-world problems selected without following any criteria. Empirical tests show how the design of experiments can result in different conclusions and stress the relevance of a priori data analysis.

Data complexity. The previous concern feeds the investigation of independent measures focused on the data; measures able to give an estimate of the apparent complexity. Literature presents a very promising set of complexity measures that estimate the difficulty of classification problems by evaluating the geometrical complexity of the class boundary. A review of the state-of-the-art reveals some limitations of the measures, which motivates our implementation and further testing.

Artificial data sets. Moreover, the unknown inherent characteristics of the problems used in experimentation and the bias of learners often lead to inconclusive results. Thus, the need for working under a controlled scenario arises. Our tentative list of relevant features is used to generate synthetic problems by means of an evolutionary multi-objective approach which incorporates the data complexity estimates as guidance.

Benchmarks. Finally, the *raison d'être* of these synthetic problems is to provide an exhaustive and configurable test framework. However, to rely on them, we have to build realistic structures. A study between real-world problems and synthetic problems consolidates the new framework as a benchmark.

The ultimate goal of this research flow is, in the long run, to provide practitioners (1) with some guidelines to choose the most suitable learner given a problem and (2) with a collection of benchmarks to either assess the performance of the learners or test their limitations.

The dissertation comprises a bonus chapter which gathers a personal point of the PhD journey, lessons learnt, and the origin of frequent existential crises.

RESUM

Les tècniques d'aprenentatge artificial tenen un ampli ventall d'aplicació i, els algorismes de classificació—aprenentatge supervisat—, els quals infereixen la frontera de decisió a partir d'un conjunt d'instàncies d'entrenament, són el nucli d'usos fascinants. La diversitat de dominis—medicina, indústria o educació—proporciona però problemes extremadament diversos pel que fa a tipus d'atributs, volum d'instàncies i distribució de dades, entre d'altres. Totes aquestes característiques han portat a la implementació de diferents estratègies per abordar cada problema de la manera més adequada, ja que el rendiment del sistema d'aprenentatge depèn en part del disseny del seu algorisme. S'han aconseguit grans progressos refinant aquests algorismes, tant que el desenvolupament de tècniques ha assolit un nivell de maduresa que ofereix milers de mètodes, tots ells força competitius i capaços d'ajustar models acurats a partir de mostres del problema a resoldre. No obstant això, i malgrat l'avenç en la classificació de dades, encara queden algunes qüestions per resoldre, sense anar més lluny com les característiques intrínseques de les dades afecten els sistemes d'aprenentatge. Això, conjuntament amb el poc marge de millora i la incertesa en l'habilitat de les tècniques per capturar completament el coneixement de les dades, indueix a mirar cap altres elements que formen part del procés d'aprenentatge. És aleshores que les dades concentren tot el protagonisme.

Aquesta tesi estudia la complexitat de les dades i el seu rol en la definició del comportament de les tècniques d'aprenentatge supervisat i explora la generació artificial de conjunts de dades mitjançant estimadors de complexitat. El treball s'ha construït sobre quatre principis que s'han succeït de manera natural. (1) La crítica de la metodologia actual utilitzada per la comunitat científica per avaluar el rendiment de nous sistemes d'aprenentatge ha desencadenat (2) l'interès per estimadors alternatius basats en l'anàlisi de la complexitat de les dades i el seu estudi. Ara bé, l'estat primerenc tant de les mesures de complexitat com de la disponibilitat limitada de problemes del món real per fer el seu test ha inspirat (3) la generació sintètica de problemes, la qual ha esdevingut l'eix central de la tesi, i (4) la proposta de fer servir estàndards artificials amb semblança als problemes reals.

Una avaluació fiable? Efectivament, un dels reptes en aprenentatge supervisat és com avaluar la qualitat dels models evolucionats per diferents tècniques d'aprenentatge ja que el seu rendiment en termes de precisió està altament lligat a les dades. Una dependència que empitjora pel fet que els estimadors d'error s'obtinguin de l'execució de diferents mètodes sobre un reduït conjunt de prova format per problemes del món real seleccionats sense seguir cap mena de criteri. Tests empírics mostren com el disseny dels experiments pot alterar les conclusions i ressalta la rellevància d'una anàlisi de dades *a priori*.

Complexitat de les dades. La preocupació relativa als estimadors d'error nodreix la investigació de mesures independents centrades en les dades, mesures capaces de donar una estimació de la complexitat aparent. En la literatura es troba un prometedor conjunt de mesures de complexitat que estimen la dificultat dels problemes de classificació a partir de l'avaluació de la complexitat geomètrica de la frontera entre classes. La revisió de l'estat de l'art revela algunes limitacions d'aquestes mesures i motiva la seva implementació i test.

Conjunts de dades artificials. A tot això cal afegir que, el desconeixement de les característiques inherents als problemes utilitzats en l'experimentació i el biaix dels sistemes d'aprenentatge deriven sovint en resultats no conclusius que fan que sorgeixi la necessitat de treballar sota un entorn controlat. La nostra proposta d'una llista provisional dels atributs més rellevants ajuda a caracteritzar els conjunts de dades i és utilitzada per generar problemes sintètics a partir d'una aproximació evolutiva multiobjectiu que incorpora estimadors de la complexitat de les dades com a guia.

Estàndards. Finalment, la raó de ser d'aquests problemes sintètics és establir un marc de prova precís i configurable amb menys temps i menys cost. Això comporta que, per realment confiar en ells, haguem de generar problemes que continguin estructures realistes. Un estudi entre problemes del món real i problemes sintètics consolida el nou marc de treball com a estàndard.

L'objectiu que es persegueix a llarg termini amb aquesta recerca és proporcionar als usuaris (1) unes directrius per escollir el sistema d'aprenentatge idoni per resoldre el seu problema i (2) una col·lecció de problemes per, o bé avaluar el rendiment dels sistemes d'aprenentatge, o bé provar les seves limitacions.

Aquesta dissertació inclou un capítol extra que recull un punt de vista molt personal sobre el camí a recórrer en una tesi, les lliçons apreses i l'origen de freqüents crisis existencials.

RESUMEN

Las técnicas de aprendizaje automático tienen gran aplicación y los algoritmos de clasificación—aprendizaje supervisado—, los cuales infieren la frontera de decisión a partir de un conjunto de instancias de entrenamiento, son el núcleo de usos fascinantes. La diversidad de dominios—medicina, industria o educación—proporciona sin duda problemas dispares en lo que atañe a tipo de atributos, volumen de instancias y distribución de datos. Todas estas características han llevado a la implementación de diferentes estrategias para abordar cada problema de la manera más adecuada, ya que el rendimiento del sistema de aprendizaje depende en parte del diseño de su algoritmo. Se han logrado progresos considerables refinando dichos algoritmos, tanto que el desarrollo de técnicas ha alcanzado su nivel de madurez ofreciendo miles de métodos, todos ellos ciertamente competitivos y capaces de ajustar modelos precisos a partir de muestras del problema a resolver. No obstante, y a pesar del avance en la clasificación de datos, quedan aún cuestiones pendientes, sin ir más lejos cómo las características intrínsecas de los datos afectan a los sistemas de aprendizaje. Esto, juntamente con el poco margen de mejora y la incertidumbre en la habilidad de las técnicas para capturar completamente el conocimiento que encierran los datos, induce a mirar otros elementos que forman parte del proceso de aprendizaje. Es entonces cuando los datos acaparan el protagonismo.

Esta tesis se adentra en el estudio de la complejidad de los datos y su papel en la definición del comportamiento de las técnicas de aprendizaje supervisado, y explora la generación artificial de conjuntos de datos mediante estimadores de complejidad. El trabajo se ha construido sobre cuatro pilares que se han sucedido de manera natural. (1) La crítica de la metodología actual utilizada por la comunidad científica para evaluar el rendimiento de nuevos sistemas de aprendizaje ha desatado (2) el interés por estimadores alternativos basados en el análisis de la complejidad de los datos y su estudio. Sin embargo, el estado primerizo tanto de las medidas de complejidad como de la limitada disponibilidad de problemas del mundo real para su testeo ha inspirado (3) la generación sintética de problemas, considerada el eje central de la tesis, y (4) la propuesta del uso de estándares artificiales con parecido a los problemas reales.

¿Una evaluación fiable? Efectivamente, uno de los retos en el aprendizaje supervisado es como evaluar la calidad de los modelos evolucionados por diferentes técnicas de aprendizaje ya que su rendimiento en términos de precisión está estrechamente vinculado a los datos. Una dependencia que empeora con el hecho de que los estimadores de error se obtengan de la ejecución de diferentes métodos sobre un conjunto reducido de prueba formado por problemas del mundo real seleccionados al azar. Pruebas empíricas muestran como el diseño de los experimentos puede alterar las conclusiones y resalta la relevancia de un análisis de datos *a priori*.

Complejidad de los datos. La preocupación relativa a los estimadores de error alimenta la investigación de medidas independientes centradas en los datos, medidas capaces de dar una estimación de la complejidad aparente. En la literatura se halla un prometedor conjunto de medidas de complejidad que estiman la dificultad de los problemas de clasificación a partir de la evaluación de la complejidad geométrica de la frontera entre clases. La revisión del estado del arte revela algunas limitaciones de dichas medidas que motivan su implementación y testeo.

Conjuntos de datos artificiales. A todo ello hay que sumarle que el desconocimiento de las características inherentes a los problemas utilizados en la experimentación y la desviación de los sistemas de aprendizaje a menudo derivan en resultados inconcluyentes que hacen que surja la necesidad de trabajar bajo un entorno controlado. Nuestra propuesta de una lista provisional de los atributos más relevantes ayuda a caracterizar los conjuntos de datos y es utilizada para generar problemas sintéticos a partir de una aproximación evolutiva multiobjetivo que incorpora estimadores de la complejidad de los datos como guía.

Estándares. Finalmente, la razón de ser de estos problemas sintéticos es establecer un marco de prueba preciso y configurable con menos tiempo y menos coste. Esto conlleva que realmente

para confiar en ellos tengamos que generar problemas que contengan estructuras realistas. Un estudio entre problemas del mundo real y problemas sintéticos consolida el nuevo marco de trabajo como estándar.

El objetivo que se persigue a largo plazo con esta investigación es el de proporcionar a los usuarios (1) unas pautas pare escoger el sistema de aprendizaje más idóneo para resolver su problema y (2) una colección de problemas para evaluar el rendimiento de los sistemas de aprendizaje o probar sus limitaciones.

Esta disertación incluye un capítulo extra que recoge un punto de vista muy personal sobre el camino a recorrer en una tesis, las lecciones aprendidas y el origen de frecuentes crisis existenciales.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

Book chapters

- **N. Macià**, A. Orriols-Puig, and E. Bernadó-Mansilla. *Beyond homemade synthetic data sets*. Berlin, Heidelberg: Springer-Verlag, 2009, vol. 5572, pp. 605-612.

Conference papers

- **N. Macià**, T. K. Ho, A. Orriols-Puig, and E. Bernadó-Mansilla. *The lansdcape contest at ICPR'10*. Contests in ICPR 2010, Berlin, Heidelberg, 2010, pp. 29-5.
- **N. Macià**, A. Orriols-Puig, and E. Bernadó-Mansilla. *In search of targeted-complexity problems*. Genetic and Evolutionary Computation Conference, 2010, pp. 1055-1062.
- **N. Macià**, A. Orriols-Puig, and E. Bernadó-Mansilla. *EMO shines a light on the holes of complexity space*. Genetic and Evolutionary Computation Conference, 2009.
- **N. Macià**, A. Orriols-Puig, and E. Bernadó-Mansilla. *Genetic-based synthetic data sets for the analysis of classifier behavior*. 8th International Conference on Hybrid Intelligent Systems, Barcelona, Spain, 2008, pp. 507-512.
- **N. Macià**, E. Bernadó-Mansilla, and A. Orriols-Puig. *Preliminary approach on synthetic data sets generation based on class separability measure*. 19th International Conference on Pattern Recognition, Tampa, Florida, USA, 2008.

Technical reports

- A. Orriols-Puig, **N. Macià**, and T. K. Ho. *Documentation for the Data Complexity Library in C++*. La Salle – Universitat Ramon Llull, 2010.

ACKNOWLEDGEMENTS

I pondered long and hard about how to, and even if I should, write these acknowledgements, until I realised that it is not always about me and my strange thoughts. I still believe that this section is only words, and that gratitude has to be shown through daily actions. However, I am also aware of how touching this public recognition can be if one mentions the right people. Therefore, my first words of acknowledgement are for Mum and Dad, always there, always giving me their love and unconditional support. I would like to thank Marc Visent, who has walked with me along this journey, witnessed my progress, and contributed to the maturation of my ideas with his valuable comments. Joan Navarro has been a dedicated follower of this work, eager to read and know every single detail about data complexity and my related achievements. Thank you for being the cheerful pale-faced friend. Finally, although wearing black looks mysterious, indeed a touch of colour can have a magical effect, and as such I am deeply thankful to Dr. Sam D. Allen, who sos'd me by adding both soundness and beauty to my prosy dissertation.

FUNDING

Funding for my Ph.D. studies has been supplied by *La Salle — Universitat Ramon Llull*, *Govern d'Andorra*, and *Fundació Crèdit Andorrà*.

In addition, *Generalitat de Catalunya* with its BE grant funded my internship at *Bell Laboratories, Alcatel-Lucent*. My internships at *University of Nottingham* and *Universidad de Granada* as well as my conference trips have been supported by the *Ministerio de Educación y Ciencia* under the projects TIN-2005-08386-C05-04 and TIN2008-06681-C06-05.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ABBREVIATIONS

AA	Amino-Acids
ADS	Artificial Data Sets
BioHEL	Bioinformatics-oriented Hierarchical Evolutionary Learning
CESCA	*Centre de Supercomputació de Catalunya*
DCoL	Data Complexity Library
EMO	Evolutionary Multi-objective Optimisation
FDS	Feature-based Dissimilarity Space
FTP	File Transfer Protocol
FH-GBML	Fuzzy Hybrid Genetic-Based Machine Learning
GA	Genetic Algorithm
GoDS	Generator of Data Sets
HEOM	Heterogeneous Euclidean-Overlap Metric
HVDM	Heterogeneous Value Difference Metric
ICML	International Conference on Machine Learning
k-NN	k-Nearest Neighbour
IBk	Instance-based learning
LDA	Linear Discriminant Analysis
MAP	Maximum A Posteriori
MOP	Multi-objective Optimisation Problem
MDL	Minimal Description Length
MST	Minimum Spanning Tree
NB	Naive Bayes
NFL	No-Free-Lunch
NSGA-II	Nondominated Sorting Genetic Algorithm II
PR	Pattern Recognition
PSDG	Parallel Synthetic Data Generator
PSP	Protein Structure Prediction
PSSM	Position-Specific Scoring Matrices

ROC	Receiver Operating Characteristic
RF	Random Forest
RT	Random Tree
SMO	Sequential Minimal Optimisation
SVD	Singular Value Decomposition
SVM	Support Vector Machine
UCI	University of California at Irvine
UF	Uniform-Frequency
UL	Uniform-Length
VDM	Value Difference Metric
XCS	eXtended Classifier System

DATA COMPLEXITY

1 WHY A THESIS ABOUT DATA COMPLEXITY?

Summary. This chapter contains the motivation and purpose of this thesis, a stimulating endeavour to understand data complexity and its importance in supervised learning. We detect two issues in the evaluation of learners: (1) a partially founded methodology to assess learners' performance and (2) the lack of an exhaustive, representative set of real-world problems to perform the testing. To combat these difficulties, we examine the current methodology and consider data complexity analysis to improve the data selection phase and generate artificial data sets. In addition, this chapter provides the road map of the dissertation.

1.1 INTRODUCTION

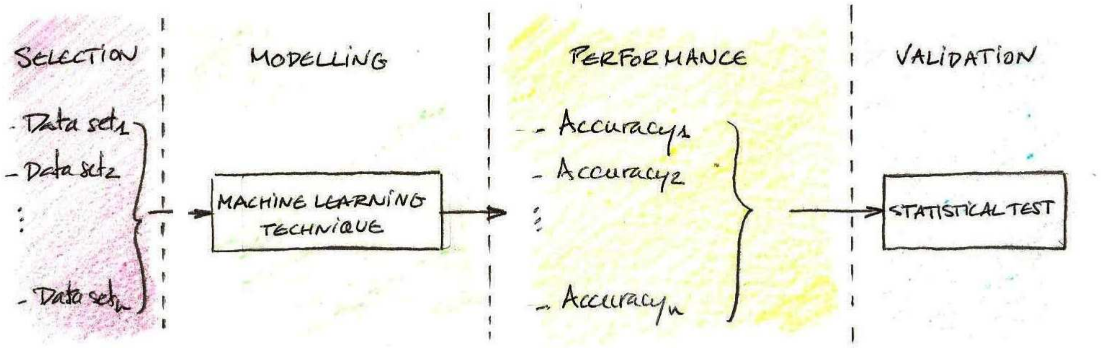
The will of knowing and mastering the details of our lives seems a challenging topic for many people, and a business for many more. With the advances in technology and capacity for storing the data of everything from everywhere, databases accumulate a deluge of terabytes which are waiting to be deciphered and explored [Mitchell, 2009]. In this area, when human beings have difficulties to comprehend large amounts of data, machine learning comes to the rescue and enables us to get some insight into underlying knowledge or hidden patterns. Machine learning consists of designing and developing programs that use experience to solve given problems [Bishop, 2006; Mitchell, 1997]. Over the last few decades, many new supervised leaning techniques based on different learning paradigms have been developed, all of them very competitive—and according to their respective authors, each one outperforms the rest. Most of such assertions are, however, fundamentally flawed by the methodologies used to evaluate and compare the performance of the algorithms. The main reason is that classifiers' accuracy depends both on the constraints of the algorithm and the intrinsic complexities of the data [Ho and Basu, 2002], the latter ignored in the majority of the analyses.

The purpose of this thesis is, therefore, to point out a concern about the current methodology of learner assessment and investigate how to amend it by examining data complexity and its influence on the behaviour of machine learning techniques.

In the following section, we identify two aspects that threaten the conclusions reached by many contributions to the machine learning community and which will set the motivation for this thesis.

1.2 TWO ISSUES IN SUPERVISED LEARNING

The design of learners lives on a plane of error and computational cost. Recently, authors have focused on refining and tuning learners, reaching an advanced state of maturity. But, when automatic classifiers are not perfect—in terms of accuracy—, is it a deficiency of the algorithm by design or a difficulty intrinsic to the given classification task? We do not really know whether the learner has fully captured the knowledge embedded in the training data set. To shed some light on this mystery, some authors have devoted their research to the analysis of the source of difficulty [Basu and Ho, 2006]. These studies are aimed at determining the nature of the class distribution and numerically characterising the class boundary to find a match between learners and data. Nonetheless, such crucial exploration of data is not part of the actual methodology to assess learners' performance. This results in an incomplete procedure that can lead to partial or inexact conclusions. On the other hand, data available for testing are not representative enough, limiting the progress of the methodology and the study of data complexity as well. Hence, we acknowledge the two following issues:



Data selection

∅

Error estimation

Resubstitution validation
Hold-out validation
k-fold cross-validation
Leave-one-out cross-validation
Repeated k-fold cross-validation
Random sub-sampling
Separate subsets
...

Performance measures

Accuracy
Kappa statistic
Mean F-measure
Macro average arithmetic
Macro average geometric
AUC of each class vs. the rest,
– using the uniform class distr.
AUC of each class vs. the rest,
– using the a priori class distr.
AUC of each class vs. each other,
– using the uniform class distr.
AUC of each class vs. each other,
– using the a priori class distr.
Scored AUC
Probabilistic AUC
Macro average mean probability rate
Mean probability rate
Mean absolute error
Mean squared error
LogLoss
Calibration loss
Calibration by bins
...

Statistical tests

Averaging over data sets
Paired t-test
Wilcoxon signed-ranks test
Wins, losses and ties
Repeated-measures ANOVA
Friedman's test
Bonferroni-Dunn test
Nemeny's test
Holm's test
Hommel's test
...

Figure 1.1: Overview of the methodology for assessing and comparing learners' performance.

1. Methodologies for the assessment of learners are not yet well-founded.
2. Available data sets do not suffice for performing rigourous testing.

Methodologies. Learner performance is usually evaluated in terms of accuracy, interpretability, and efficiency. The first of these measures is the most significant to determine the learner quality and is generally estimated by running the learner over different data sets. Unless the development of the learner is problem-oriented, culling data sets for the experiments is often an obscured phase. The lack of detail and guidance in that selection manifests itself in a badly supported procedure to evaluate learners that involves biased conclusions; a cautionary note already made by Salzberg [1997]. Fig. 1.1 shows the steps to evaluate the learners' performance and summarises the flipping pieces along this process to select data sets, performance measures [Ferri et al., 2009], validation methods [Dietterich, 1998; Demšar, 2006], and statistical tests. Huge differences in the achievements for each phase—selection, modelling, performance, and validation—bring the attention to data set selection, which is a far-reaching implication.
Data sets. There are two types of data used to carry out the evaluation of learners' performance: data coming (1) from real-world problems and (2) from synthetic data sets. Public repositories such as the University of California at Irvine (UCI) repository [Frank and Asuncion, 2010] have become popular as repetitive test bed scenarios, from which authors pick a small sample of problems to perform their experimentation. Although making these collections accessible contributes to subsequent works being comparable, they are not big nor representative enough—

in number and diversity—to get conclusive results. Synthetic data sets may solve both issues and allow us to build problems with predefined characteristics at only a minor cost. Indeed, in spite of being accustomed to storing data on a vast scale due to the current information age, there are still many knowledge domains where the high cost or difficulty of performing experiments hinder data collection [Jeske et al., 2005]. On the other hand, the use of artificial data sets offers better understanding of learners' behaviour since the complexity of the problem under study is known. For example, under a controlled framework, Japkowicz and Stephen [2002] investigated algorithm dependence on different degrees of class imbalance. Other issues that are often present in real-world problems but cannot be easily identified nor taken separately (e.g. data sparsity, noise, missing values, or dimensionality) can also be introduced in a controlled way. Actually, the approach of using synthetic data sets has closely been followed in fields such as evolutionary optimisation, where the design of the so-called boundedly-difficult problems has provided great insights into the real behaviour of optimisation techniques [Goldberg, 2002]. However, this approach has rarely been pursued in supervised learning due to (1) the complexity of defining what a difficult supervised learning problem is, (2) the inability of finding real-world problems with truly different complexity—assuming that complexity can be evaluated—, and (3) the lack of investigations about generating artificial problems with a certain complexity. For these reasons, the use of synthetic data sets is still far away from being wide spread and if any work uses them is with very specific homemade data sets, which are not representative enough to become benchmarks.

The objectives of this thesis, articulated in more detail in the next section, cope with the two aforementioned drawbacks, which are intimately related to the selection and generation of data sets.

1.3 thesis objectives

Although there is no standardised procedure on the evaluation of learners—i.e. different data sets, different reference techniques, different performance measures, and different statistical procedures are used—, recent works have tried to systematise or provide guidelines for each phase. Efforts have been made to identify the most influential learning techniques [Wu et al., 2007], to design procedures to estimate learners' error [Dietterich, 1998], and to define a statistical framework to extract significant and reliable conclusions from results [Demšar, 2006; García and Herrera, 2008]. These advances have quickly been accepted by practitioners; however, in the majority of these studies data set selection is still unexamined.

After sifting through the current methodology used to analyse the performance of learners and taking into account that learner performance is strongly dependent on data characteristics, we start by empirically building evidence of how data can influence the conclusions and we then stress the need of further probing their complexity.

Thus, the objectives of this thesis consist in transferring thoughts, sharing observations, and making some contributions to the field about the:

1. Study of a set of complexity measures as an alternative/complementary method to analyse the behaviour and performance of machine learning techniques.
2. Development of a framework for experimental analyses based on artificial problems.
3. Consolidation of artificial problems as benchmarks.

In the following, each objective is elaborated.

Study of a set of complexity measures as an alternative/complementary method to analyse the behaviour and performance of machine learning techniques. A closer reading of data complexity can help practitioners to understand the performance of supervised learning techniques and their behaviour. Previous attempts have been made to link the resultant characterisation of data complexity to accuracies achievable by different classifiers [Bernadó-Mansilla, 2004]. Early evidence showed that the domains of competence of certain classifier families are separable in such a space of data complexity measures [Duin and Pekalska, 2006; Basu and Ho, 2006;

Luengo and Herrera, 2010; Macià et al., 2010a]. We revive this train of thoughts and add some experiments to bolster the inclusion of data complexity analysis in the evaluation of learners' performance as an essencial procedure to guide the selection of data sets according to the purpose of the experimentation.

Development of a framework for experimental analyses based on artificial problems. The idyllic landscape refers to a landscape providing a complete coverage of the characteristic space, as small as possible but able to map all problems and have enough resolution, granularity in the complexity values. This framework can help to (1) better understand the domain of competence of different learners and (2) compare different learners on the complexity space. The creation of this space is not trivial since there are many ways to characterise a problem and its linkage to the properties of the learners is not a direct function. We rely on evolutionary computation to evolve different complexity dimensions defined by the complexity measures proposed by Ho and Basu [2002] and produce artificial data sets.

Consolidation of artificial problems as benchmarks. Boundedly-difficult problems are very helpful to test learners under specific conditions. However, synthetic data sets with real-world structures are preferable. We study real-world problems and try to achieve similar structures with artificial data sets in order to define some benchmarks and provide practitioners with a common baseline.

1.4 ROAD MAP

The dissertation is organised as follows.

Chapter 2. **Assessment of learners in machine learning. Reliable enough?** reviews the work of the research community regarding the development and improvement of machine learning techniques. Lackadaisical decisions in the evaluation procedure, such as which data sets have to be used in the experimentation and how many[1], lead us to call the interest of this kind of research into question and propel the use of data complexity analysis.

Chapter 3. **Data hold the key of learnable patterns** introduces data complexity and some complexity measures proposed by Ho and Basu [2002]. These measures, which estimate the geometry of the class boundary, have been revised and implemented in the Data Complexity Library (DCoL) [Orriols-Puig et al., 2010].

Chapter 4. **Complexity measures may explain supervised learning behaviour** envisions the complexity characterisation as a basis to study data set selection and design a meta-learner that, given a new classification problem, automatically suggests which algorithm should be utilised to maximise the learner accuracy. Besides, it describes other usage of the complexity measures in the literature thus far. Nevertheless, despite their popularity, the publicly available data sets appear to be a scarce sample which should be perused.

Chapter 5. **Real-world problems vs. Artificial data sets** analyses the coverage of the available testing problems on the complexity space. It shows the difference—in terms of intrinsic complexity—among real-world problems and some popular artificial data sets, and highlights a few gaps in certain dimensions of complexity. Standing in the way of enlarging the diversity of the problems is the concern of providing artificial problems with real-world structure and high dimensionality.

Chapter 6. **Generation of artificial data sets** presents our approach which combines complexity measures and evolutionary computation in a quest to create synthetic problems with realistic structures. Thanks to this synthesising technique, we are able to build boundedly-difficult problems and fill some holes of the complexity space.

Chapter 7. **The landscape** details the generation of a large collection of synthetic problems which provide sufficient diversity to cover several gaps of the complexity measurement space and may help preliminary benchmarks for learner assessment to blossom. This may encourage

[1] How many learners should be involved in the comparison, which ones, and what statistical tests should be applied are also other considerations to take into account in the experimental design. However, this thesis focuses on the data analysis.

the search of golden standards—term used in medicine—and contribute to the improvement of the UCI repository.
Chapter 8. **Summary, conclusions, and future work** ends the dissertation with a summary of the scientific content of this thesis and several future directions for investigation.

Each chapter ends with the scientific contributions of this thesis to the community, which have pushed the boundaries of the field.

Finally, the epilogue **Philosophical doubts...** contains some comments born from existential crises about randomness, caprice, and surprise in research, some comments about what research means and what research really is from my point of view.

2 ASSESSMENT OF LEARNERS IN MACHINE LEARNING. RELIABLE ENOUGH?

Summary. This chapter reviews the literature of the different methodologies employed in learner comparison and performance evaluation, and highlights some bad practices, which are empirically supported by a case study where well-known supervised learning techniques are compared over different collections of problems. Contradictory conclusions about the excellence of the learners lead to stressing the key role of data set selection, as well as the importance of data complexity analysis.

2.1 INTRODUCTION

Increasing interest in *pattern recognition* and *machine learning* has fed the growth of a rich set of brand-new learning techniques which have been designed with the aim of improving existing algorithms. This kind of research has resulted in several benchmarking studies which claim the superiority of new learners by comparing their performances with those of a set of reference learners. These comparisons have usually been built through a three-phase procedure: (1) select a collection of data sets, typically from public repositories, (2) choose the methods to compare the new approach with, and (3) extract statistical conclusions, often from tables of performance measures of each learner on each problem. Although there is no standardised agreement on the comparison methodology, recent works have tried to systematise or provide guidelines for each phase. For instance, Wu et al. [2007] determined the most influential learning techniques, Dietterich [1998] proposed procedures to estimate learners' error, and Demšar [2006], Martorell [2007], and García and Herrera [2008] defined statistical frameworks to extract significant and reliable conclusions from results. Despite these important advances, which have been accepted and adopted by practitioners, data set selection is still overlooked in the majority of the studies; data sets are haphazardly selected without a detailed analysis of their characteristics. A proper selection is, however, a critical aspect in any comparison since results are built upon the selected data sets, and therefore, a change in the data collection may cause variations in the final conclusions. A likely explanation of why data analysis is not amply performed in this field is because it does not fall neatly into the class of approaches with which most practitioners are familiar.

The purpose of this chapter is to highlight the importance of data set selection by studying whether it can belie the conclusions resulting from evaluations and comparisons of supervised learning techniques. To answer this question, we conduct a systematical comparison of three well-known learners over different collections of real-world problems with no apparent difference between them.

In the following, we first briefly review techniques and methodologies for the assessment of supervised learning and we then replicate the most popular approach for learner comparison. We present a case study which empirically proves that conclusions can be very different according to the collections of data sets used and motivates us to proceed in the face of criticism.

2.2 SUPERVISED LEARNING: TECHNIQUES AND METHODOLOGIES

In the past few years, the literature has been extended with a vast amount of new machine learning techniques. Among these proposals, we can distinguish enhanced techniques from the referenced learners of each learning paradigm—logic-based algorithms, perceptron-based techniques, statistical learning algorithms, instance-based learning, and support vector machines [Kotsiantis et al., 2006]. Certainly, learning approaches and knowledge representation will be

Table 2.1: Overview of the current state of data set selection in the experiments.

		PR			ICML		
	2008	2009	2010	2008	2009	2010	Total
Total published papers	322	315	353	158	160	152	1460
Papers relevant to our study	31	50	48	35	27	24	215
Source of data sets[2]							
Repositories	25	32	36	31	22	20	166 (77.2%)
Homemade	8	17	14	3	3	4	49 (22.8%)
Specific	2	11	5	3	2	1	24 (11.2%)
Number of data sets[3]							
1	7	19	12	6	6	5	55 (25.5%)
(1,10]	16	22	28	23	16	14	119 (55.3%)
(10,30]	7	7	8	4	5	5	36 (16.7%)
>30	1	2	0	2	0	0	5 (2.3%)
Number of classes							
2	8	11	9	3	2	2	35
>2	14	13	10	5	5	4	51
Number of instances							
(0,1000]	14	11	14	2	3	3	47
(1000,10000]	9	7	10	8	6	3	43
(10000,100000]	3	1	6	4	3	2	19
>100000	2	1	1	0	1	1	6
Number of attributes							
(0,10]	10	13	12	3	4	3	45
(10,25]	11	12	13	3	5	3	57
(25,100]	11	10	12	3	4	3	43
>100	4	2	3	6	3	2	20

essential aspects to take into consideration to bridge complexity and learners' performance. But, how has the community demonstrated the relevance of new learning techniques so far? This section presents a literature review that shows the different ways in which the research community has been assessing the performance of classifiers.

First of all, we analyse papers from journals and proceedings of machine learning and pattern recognition, particularly Pattern Recognition (PR) (2008-2010) and International Conference on Machine Learning (ICML) (2008-2010)[1]. We focus on those papers where a particular classifier has been analysed through the comparison of its performance with at least one other classifier on a set of problems. Actually, we select those which are categorised as *classification* or contained this word and derivatives in the title or abstract. Thus, among 1,460 papers, 215 constituted our panel study. Table 2.1 summarises the test beds used most frequently in the literature.

[1] 2008: http://icml2008.cs.helsinki.fi/abstracts.shtml
2009: http://www.machinelearning.org/archive/icml2009/abstracts.html
2010: http://www.icml2010.org/abstracts.html#main

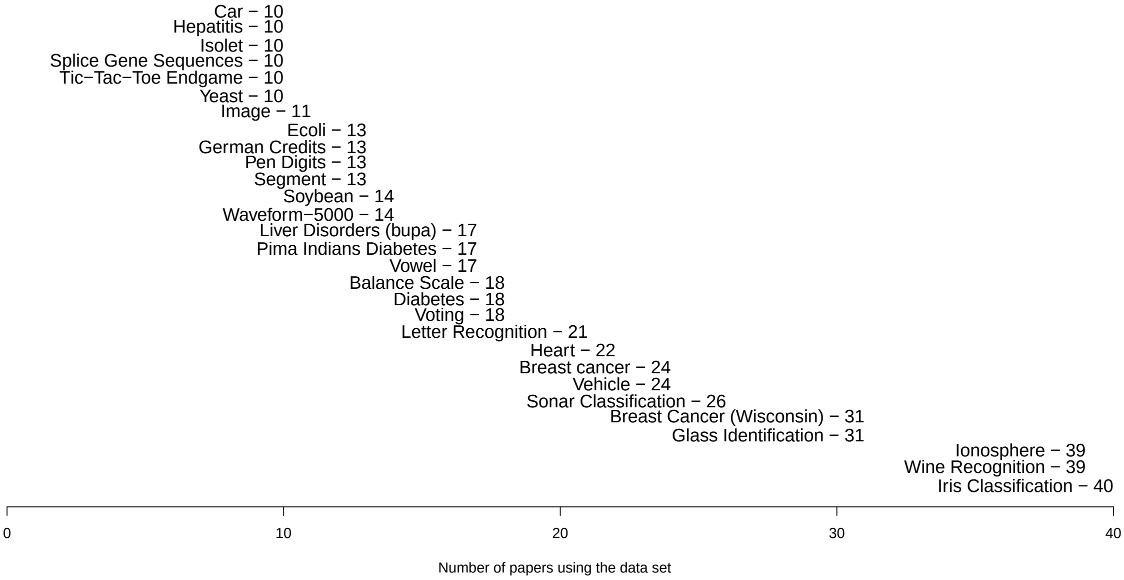


Figure 2.1: The thirty most commonly used data sets according to our overview. The x-axis refers to the number of papers from the study that use the data set in the experimentation.

The majority of the papers resort to data sets from public repositories (77.2%), the UCI repository being the most popular (63.9%). We observe that certain data sets from the UCI repository are frequently used with no apparent reason, e.g. *Iris classification*, *Wine recognition*, *Ionosphere*, *Glass identification*, and *Breast cancer (Wisconsin)* (see histogram in Fig. 2.1). This list partly matches the popularity ranking maintained by the UCI repository (see Table 2.2). Fig. 2.2 shows the distribution for PR and ICML separately; both communities seem to apply the same data sets with the same frequency. We indicate though that this is an approximate picture since sometimes data sets are altered for the experimentation or their version is not specified—very common for *Breast cancer*, *Glass identification* problems. Popularity promotes a higher use of these data sets since authors often need to place the learner performance with respect to well known expected performances. These problems happen to be popular, easy to use due to the size of the data set, the type of attributes, or the presence of missing data—considered as extrinsic characteristic in our taxonomy, see Chapter 6. Other papers (22.8%) use synthetic data sets specifically designed for the particular purpose of the paper, and a smaller percentage (11.2%) uses a given problem which must be addressed.

Regarding the size of the test bed, the majority of the works select a small set of problems. Particularly, more than 50% of the papers use around 2–10 data sets. In general, the selection is composed of at most thirty data sets (16.7%) and in a very few cases (2.3%) works exceed this threshold. The mean value is eight data sets when data sets are picked from the UCI repository, but there is not any defined reason for such a value; it is mostly due to the compromise between what is normally understood as a representative amount of problems and the computational cost.

The dimensionality of the problems tends to be small. Data sets with fewer than 1,000 instances and data sets with up to 100 attributes are the most common. Nonetheless, it is very difficult to parse such information because it was often not included in the papers. As seen in the low values reported in Table 2.1, few papers mention these characteristics—although this trend is reversing. Since October 2008, most of the papers from *Pattern Recognition* show a table with the *observable* (also referred to as extrinsic) characteristics of the data sets—i.e. the number of attributes, classes, and instances—and justify their selection under the basis of diversity. However, there is no

[2] The sum of the three sources can be greater than the number of studied papers since some works are used in the experimentation data sets from different sources at the same time.

[3] In sub-sections *number of classes*, *number of instances*, and *number of attributes*, the number of papers can be lower than the number of studied papers since.

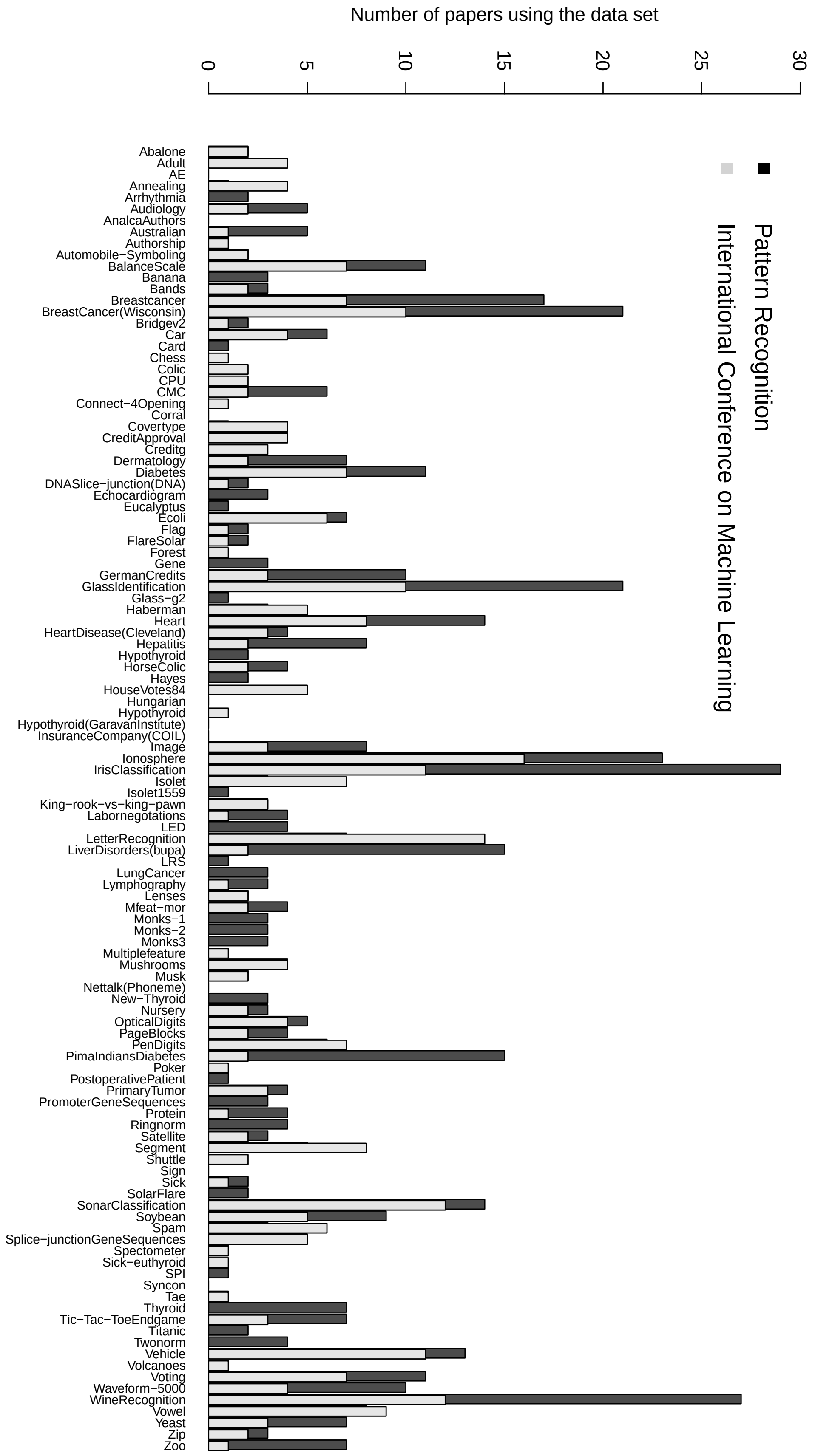


Figure 2.2: Data sets from the UCI repository used in PR and ICML.

Table 2.2: Most popular classification data sets from the UCI repository according to `http://archive.ics.uci.edu/ml/` on Jan, 19 2010 (hits counted since 2007) and their characteristic description. *#Cl* is the number of classes, *#Inst* is the number of instances, and *#Att* is the number of attributes. *#Real*, *#Int* and *#Nom* indicate the number of real-, integer- and nominal-valued attributes respectively. *%missInst*, *%missAtt*, and *%missVal* correspond to the percentage of instances with missing values, attributes with missing values, and the total percentage of missing values respectively. Finally, *%Maj* is the percentage of instances of the majority class and *%Min* is the percentage of instances of the minority class.

Data set	**Hits**	**#Cl**	**#Inst**	**#Att**	**#Real**	**#Int**	**#Nom**	**%missInst**	**%missAtt**	**%missVal**	**%Maj**	**%Min**
Iris	177422	3	150	4	4	0	0	0.00	0.00	0.00	33.33	33.33
Adult	130239	2	48842	14	0	6	8	7.41	21.43	0.95	76.07	23.93
Wine	115366	3	178	13	13	0	0	0.00	0.00	0.00	39.89	26.97
Breast Cancer Wis. (Diagnostic)	94377	2	699	9	0	9	0	2.29	11.11	0.25	65.52	34.48
Abalone	74795	29	4177	8	7	0	1	0.00	0.00	0.00	16.50	0.02
Car Evaluation	71042	4	1728	6	0	0	6	0.00	0.00	0.00	70.02	3.76
Poker Hand	70613	10	1025010	11	0	5	6	0.00	0.00	0.00	50.12	7.80e-4
Yeast	51539	10	1484	8	8	0	0	0.00	0.00	0.00	31.20	0.34
Internet Advertisements	49457	2	3279	1558	3	0	1555	28.06	0.19	0.05	86.03	13.97
SPECT Heart	46258	2	267	22	0	0	22	0.00	0.00	0.00	79.40	20.60

guarantee that diversity of the dimensionality also implies diversity of intrinsic complexity. Even though this has happened, it is also debatable that one can conclude any quality of the method across several types of problems [Wolpert, 1992, 1996].

On a side note, we are not suggesting to neglect any of these characteristics since relating difficulty factors to performance will be essential to assess learners—their robustness, scalability, and predictive accuracy. For example, learner scalability can be tested by varying the number of features and the number of instances. Noise, missing values, or ambiguity as well as irrelevant and redundant attributes are suitable characteristics to test the learner robustness. Determining the number of classes of the problem adds another layer of difficulty, since the interaction between classes can be interpreted differently depending on how they are grouped.

The estimation of classifiers' performance and the use of statistical tests are not directly relevant to this work. Nevertheless, it is worth mentioning that performance assessment includes some steps to support the experiments. For instance, usually cross validation or leave-one-out methods are applied to estimate the error, which is especially needed when only small data set samples are available. Across the literature, we observe that there is no agreement in the parametrisation of the error estimation and the well-known k-fold cross validation can be set as $k = \{4, 5, 10, 20\}$. Classifiers' performance is measured almost exclusively by classification accuracy or error, although Receiver Operating Characteristic (ROC) curves provide more information about the error of the classifiers thanks to measures like the specificity and sensitivity [Swets, 1996; Fawcett, 2006]. Statistical tests are used to confer reliability on the observed differences between the methods. However, most of them are based on pairwise comparisons even though several classifiers are involved. Multiple comparison tests have still not been applied despite the recommendation by Demšar [2006].

After examining the papers published by leading journals on neural networks in 1993 and 1994, Prechelt [1996] sadly pointed out that 78% of the contributions did not even meet the following soft requirement: an evaluation that *"uses a minimum of two real or realistic problems and compares the results to those of at least one alternative algorithm"*. Substantial progress in experimental testing has been made since this remark; regrettably, his concerns are still valid and the evaluation of the learning techniques is very often not performed thoroughly enough. The suggestion of striving for better assessment practices, and increasing the number of publications using easily accessibly benchmark problems remain a milestone.

Our impression is that the community is aware of the need for a proper methodology for estimating performance and statistical significance, but has not paid much attention to the data set selection. When it seems that there is some interest, indications like *"we have chosen these datasets because they vary greatly in size, number of classes, and other characteristics such as class distribution"* or *"selection criteria for this subset of the repository were minimum requirements on sample size and number of attributes"* are not accompanied by any analysis. In addition, *"we chose the first four data sets that represented standard regression problems and had a few hundred instances; no other data sets were tried"* in parenthesis or *"these datasets cover a full spectrum of values for each of the characteristics"* when the test bed is just composed of fifteen problems appear to be quick justifications added to satisfy some reviewers' comments and mask the fact that data selection is still shallow. We conjecture that data set selection is a critical aspect that may cause a strong bias in the results. A good choice of the data collection is the basis for a good experimental framework over which the rest of the statistical methodology should be sustained.

Choosing data set carelessly from large repositories may thwart the statistical analysis, especially when data sets with similar characteristics are selected; some significant differences may not be detected if certain types of problems are not represented in the collection of data sets. Conversely, having a large collection of data sets with different complexities may lead to the conclusion that the learners perform the same on average, as shown in the case study that follows.

2.3 LEARNER 1, LEARNER 2, AND LEARNER 3: READY TO RUMBLE

The previous section showed that the literature often analyses the performance of learners by means of comparisons between several techniques on a moderate number of data sets. This section replicates this approach and evidences the risks behind multiple learner comparisons. In the following, we present a case study—methodological description and analysis of results—where three learners are compared upon three different collections to empirically prove that, although a correct statistical analysis is performed, contradictory conclusions can be reached depending on the collection of data sets used.

2.3.1 *Methodology for learner comparison*

For the analysis that we conduct, we depart from the classical methodology: (1) selection of data sets, (2) selection of learners, and (3) performance analysis which involves the selection of statistical tests to better support the conclusions.

Firstly, we collect 88 data sets from the following repositories: the UCI machine learning[4], Delve[5], LibSVM[6], and Kent Ridge Bio-Medical[7]. For problems that contain more than two classes, each discrimination of a class with respect to all the other classes are considered as an individual data set. Therefore, for an m-class problem ($m > 2$), the data set are transformed into m two-class problems. This data preprocessing results in 308 binary classification problems. Among them, we deliberately build three collections of twenty data sets each—twice the average number used in the literature as seen in Sect. 2.2. Each collection enables each particular learner to statistically outperform the others. To avoid biasing the results, we do not insert two data sets generated from the same m-class problem into the same collection. Also, following the community's belief about diversity in data set selection, extrinsic characteristics of the data sets, in terms of number of attributes and number of instances, are summarised in Fig. 2.3. We can see that the picked samples are spread and reach a good coverage in the space defined by these two dimensions.

[4] http://mlearn.ics.uci.edu/MLRepository.html
[5] http://www.cs.toronto.edu/~delve/
[6] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
[7] http://sdmc.lit.org.sg/GEDatasets/Datasets.html

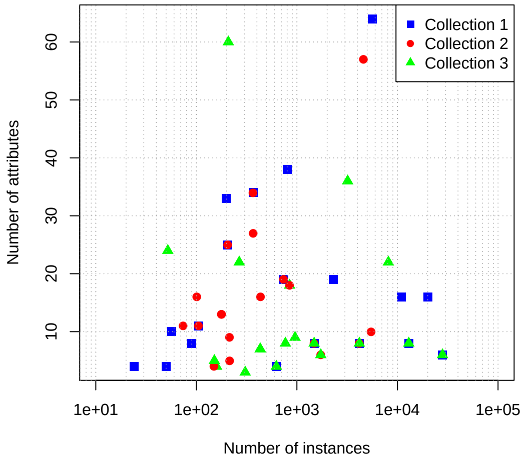


Figure 2.3: Extrinsic characteristics—number of instances and number of attributes—of the data sets included in the three collections. Blue squares represent data sets from collection 1, red points, data sets from collection 2, and green triangles, data sets from collection 3.

Secondly, we select three widely-used learning techniques from different learning paradigms: Instance-based learning (IBk) [Aha et al., 1991], Random Forest (RF) [Breiman, 2001], and Sequential Minimal Optimisation (SMO) [Platt, 1999]. IBk is an implementation of the nearest neighbour algorithm; to classify a previously unseen instance, it searches the k nearest neighbours and returns the majority class among them. RF builds an ensemble of decision trees which are generated with a group of randomly selected attributes from the sampled training set; the output of new test instances is inferred by considering the most popular class among the trees. This is Breiman's modification of the random decision forest method proposed by Ho [1995, 1998]. SMO is an efficient implementation of support vector machines [Vapnik, 1995]. All these methods are run using the WEKA package [Witten and Frank, 2005] with the following configurations: (1) $k = 7$ for IBk, (2) a polynomial kernel of order 5 for SMO, and (3) the rest of the parameters set to their default value. The performance of each technique is estimated with stratified ten-fold cross validation [Dietterich, 1998].

Finally, we compare the results of the three learners with multiple comparison procedures based on non-parametrical tests as suggested by Demšar [2006]. The statistical analysis first applies the Friedman's test [Friedman, 1937, 1940], equivalent to the repeated-measures ANOVA [Fisher, 1959], to check the null hypothesis that all the learning algorithms perform equivalently on average. If the Friedman's test rejects the null hypothesis, post hoc tests are applied to identify which learners behave differently. Then, the aim turns to analyse whether all the methods perform equivalently to the best ranked learner. To this end, we use the Bonferroni-Dunn test [Dunn, 1961], which identifies significant differences between a control learner (in our case, the best ranked method) and the others. The Bonferroni-Dunn test indicates that a method is significantly different from a control learner if the corresponding average rank differs by at least a critical distance (CD), which is computed as

$$CD = \left| q_\alpha \sqrt{\frac{n_\ell(n_\ell + 1)}{6 n_{ds}}} \right|, \tag{2.1}$$

where n_ℓ and n_{ds} are the number of learners and the number of data sets respectively, and q_α is the critical value based on the studentized range [Sheskin, 2000].

2.3.2 *Analysis of results: And the winner is...*

Tables 2.3 and 2.4, 2.5 report the average test classification accuracies obtained by each learner on the problems of collection 1, collection 2, and collection 3, respectively. The row *Frd* is the

Table 2.3: Comparison of IBk, RF, and SMO on the first collection of data sets (DS1). The main entries correspond to the average test classification accuracies obtained by each learner over a ten-fold cross-validation. The row *Frd* reports the p-value resulting of applying the Friedman's test and the row *Rank* provides the average rank of each learner.

DS1	**IBk**	**RF**	**SMO**
aba-17	98.99	98.97	98.97
ann-3	100	100	100
authors-2	100	99.76	100
bal-2	94.24	87.36	92.80
bondrate-2	78.95	78.95	77.19
briv1-2	85.71	84.76	79.05
cmc-1	76.44	74.54	69.86
drm-5	100	99.45	99.18
euc-1	86.14	84.10	82.34
krk-1	99.90	99.93	99.95
let-21	99.73	99.61	99.69
lng-2	84.38	81.25	75.00
nrs-0	100	100	100
opt-0	99.96	99.73	99.89
pen-2	99.84	99.57	99.70
pos-1	73.33	62.22	61.11
statlog-sgm-1	100	100	100
wav21-3	100	100	100
wbcd	96.57	96.14	94.99
yea-8	99.12	98.65	99.07
Frd		**$2.42 \cdot 10^{-4}$**	
Rank	**1.35**	**2.30**	**2.35**

p-value obtained from applying the Friedman's test, and the row *Rank* provides the average rank of each learner, which is calculated as follows. For each data set, we rank the learning algorithms according to their test classification accuracy; the learner with highest accuracy holds the first position of the ranking. If a group of learners has the same performance, we assign the average rank of the group to each of the learners[8]. The detailed information about accuracies is visualised in Fig. 2.4.

Fig. 2.4a plots the classification accuracies of IBk, RF, and SMO obtained over the first collection of data sets. At first glance, we have no chance of finding any difference among the learners since their accuracies lie within similar intervals. In the same vein, we do not detect differences in Figs. 2.4b and 2.4c that refer to the accuracies reached over the second and third collection of data sets. However, note that for the three collections the Friedman's test rejected the null

[8] The table below is an illustrative example of how to calculate learners' ranking, especially how to proceed in case of multiple ties.

Data set	**Accuracy**			**Rank**		
	IBk	**RF**	**SMO**	**IBk**	**RF**	**SMO**
dataset1	94.09	86.41	92.98	1	3	2
dataset2	97.98	97.97	97.98	1.5	3	1.5
dataset3	94.00	94.00	97.98	1	2.5	2.5
dataset4	100	100	100	2	2	2

Table 2.4: Comparison of IBk, RF, and SMO on the second collection of data sets (DS2). The main entries correspond to the average test classification accuracies obtained by each learner over a ten-fold cross-validation. The row *Frd* reports the p-value resulting of applying the Friedman's test and the row *Rank* provides the average rank of each learner.

DS2	**IBk**	**RF**	**SMO**
aba-2	99.57	99.59	99.64
aud-6	93.36	95.13	96.90
aut-5	91.22	97.56	93.66
briv1-3	87.62	89.52	88.57
car-3	97.28	98.55	100
drm-3	96.45	97.81	95.36
ech	91.89	98.65	90.54
euc-0	89.95	92.12	91.85
gls-0	80.37	84.11	78.50
ion	85.19	93.45	86.04
irs-1	96.00	94.67	66.00
pbc-0	95.78	97.61	94.61
spa	90.18	94.83	61.60
thy-0	92.56	93.49	87.44
veh-3	93.50	95.74	97.40
wav40-2	87.84	88.22	87.16
win-1	96.63	98.31	96.07
wne-1	96.63	98.31	96.07
yea-1	75.61	75.34	75.00
zoo-2	95.05	97.03	95.05
Frd	**$5.78 \cdot 10^{-4}$**		
Rank	**2.33**	**1.30**	**2.38**

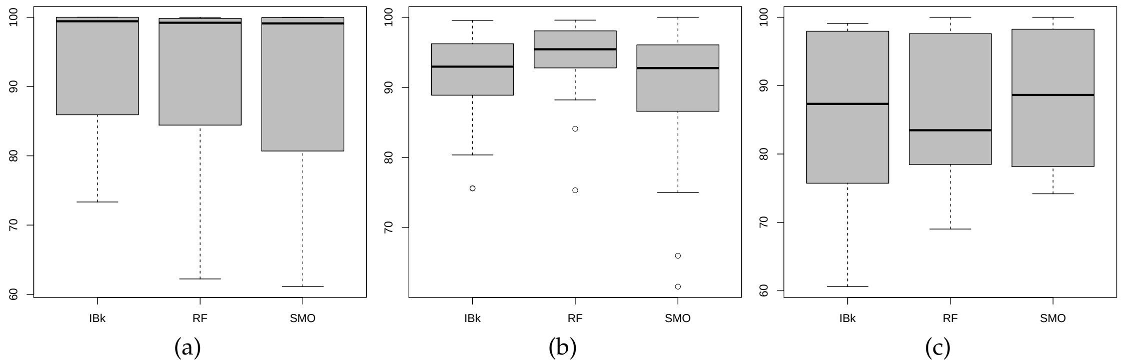


Figure 2.4: Test classification accuracies of IBk, RF, and SMO over (a) the first collection of data sets, (b) the second collection of data sets, and (c) the third collection of data sets.

hypothesis. The three p-values $\{\mathtt{Frd}_{DS1} = 2.42 \cdot 10^{-4}, \mathtt{Frd}_{DS2} = 5.78 \cdot 10^{-4}, \mathtt{Frd}_{DS3} = 4.51 \cdot 10^{-4}\}$ are less than 0.05, the critical value, and it can be concluded that at least two of the learners are significantly different from each other. As a consequence, the Bonferroni-Dunn test is applied

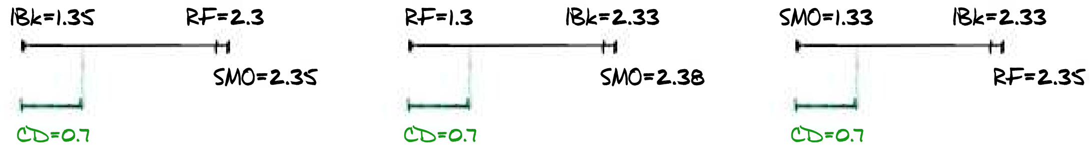
Table 2.5: Comparison of IBk, RF, and SMO on the third collection of data sets (DS3). The main entries correspond to the average test classification accuracies obtained by each learner over a ten-fold cross-validation. The row *Frd* reports the p-value resulting of applying the Friedman's test and the row *Rank* provides the average rank of each learner.

DS3	**IBk**	**RF**	**SMO**
aba-8	81.25	78.96	83.50
aud-4	99.12	100	100
bal-1	92.16	86.40	92.16
car-1	93.87	91.20	98.09
flg-4	97.94	97.42	97.94
hay-1	79.25	77.99	80.50
hrs	60.60	69.02	74.18
hrt-0	82.51	79.21	76.90
krk-17	98.90	98.91	99.42
krkp	95.40	98.87	99.22
lns-2	62.50	70.83	75.00
pas-1	72.22	80.56	80.56
pim	74.74	73.83	75.13
shu-2	92.31	90.38	96.15
tae-2	66.23	80.13	75.50
tic	98.75	93.01	99.69
veh-1	76.71	77.07	85.11
vot	77.64	79.15	79.46
whi-3	98.41	98.41	98.41
yea-7	97.98	97.78	97.98
Frd	$4.51 \cdot 10^{-4}$		
Rank	**2.33**	**2.35**	**1.33**

Table 2.6: Results obtained by IBk, RF, and SMO over the three collections of data sets. Each cell reports the average rank of each learner and, the significantly best learner of each comparison is marked in bold. The last column reports the p-value according to the Friedman's test.

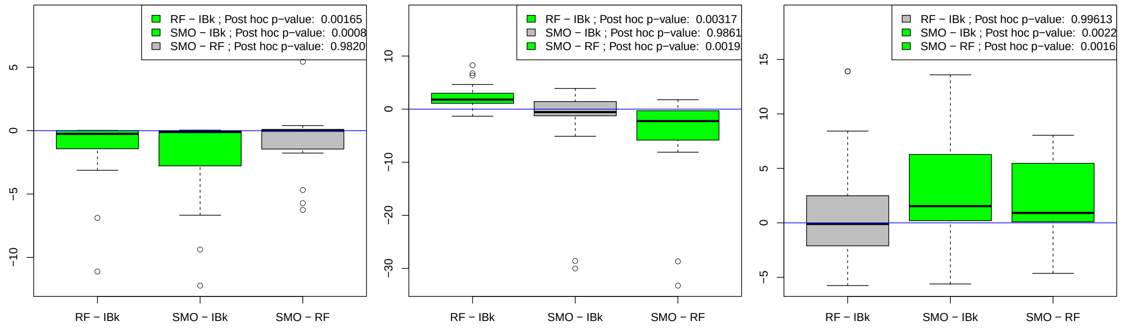
	IBk	**RF**	**SMO**	**Friedman**
Collection 1	**1.33**	2.30	2.35	$2.42 \cdot 10^{-4}$
Collection 2	2.33	**1.30**	2.38	$5.78 \cdot 10^{-4}$
Collection 3	2.33	2.35	**1.33**	$4.51 \cdot 10^{-4}$
All data sets	**1.99**	**1.98**	**2.02**	0.91430

to find out which methods significantly degraded the performance with respect to the best method at a significance level of 0.05, which corresponds to a CD = 0.70 (see Eq. 2.1). Table 2.6



(a) (b) (c)

Figure 2.5: Critical difference diagram for (a) collection 1, (b) collection 2, and (c) collection 3. CD is the critical distance for the Bonferroni post hoc correction.



(a) (b) (c)

Figure 2.6: Post hoc Friedman's test applied to discover significant differences among pairs of learners over (a) the first collection of data sets, (b) the second collection of data sets, and (c) the third collection of data sets. Green boxplots indicate the pair of learners that present significant differences.

summarises the rank results for each learner over each collection and shows the average rank of each learner. The last column reminds us of the p-value of the Friedman's test.

After crunching the numbers, for the first collection of data sets, IBk presents significantly better results than RF and SMO (see Fig. 2.5a). For the second collection, RF presents significantly better results than IBk and SMO (see Fig. 2.5b). And, for the third collection, SMO presents significantly better results than IBk and RF (see Fig. 2.5c).

The post hoc Friedman's test supports the results of the previous statistical test and shows that the difference between the three learners in the first collection is due to differences between IBk and RF (p-value $1.6 \cdot 10^{-3}$) and between IBk and SMO as well (p-value $8.4 \cdot 10^{-4}$)(see Fig. 2.6a). We also confirm the conclusions for the other two collections where there are significant differences between pairs of classifiers. Fig. 2.6b shows that RF outranks IBk (p-value $3.17 \cdot 10^{-3}$) and SMO (p-value $1.93 \cdot 10^{-3}$) and Fig. 2.6c shows that SMO outperforms IBk (p-value $2.26 \cdot 10^{-3}$) and RF (p-value $1.65 \cdot 10^{-3}$).

So, apparently each collection has its own best classifier. However, no significant differences can be found when all the data sets are considered in a single comparison (see the last row of Table 2.6)—the distance between ranks is hardly of some decimals. Similar conclusions can be obtained with other statistical procedures such as the Holm's test [Holm, 1979] and the Hommel's test [Hommel, 1988].

Therefore, the experimental results determine that there is no overall winner since contradictory conclusions are extracted from different collections of data sets despite the statistical analysis—only local winners for each collection. Reliability of this methodology is challenged by an arbitrary selection of data sets.

2.4 CRITIQUE OF THE CURRENT METHODOLOGIES FOR THE PERFORMANCE EVALUATION

The case study evidences the principal limitation of the typical comparisons: conclusions reached over a collection of data sets are strictly valid only for that collection of data sets. Trying to extrapolate the conclusions to other domains may lead to incorrect claims. This section just emphasises the need to revise the current methodology. We suggest to examine the data—complexity and distribution—before any experimentation since these things are at the heart of what supervised learning methodologies need to master to be effective.

The case study exemplifies two of the most common situations that can be found in comparative analyses. On the one hand, a comparison that considers just a single collection of data sets may result in conclusions too specialised to the chosen domains. In general, specific learning techniques may be overrated if the comparison is made over collections that contain data sets with certain characteristics which happen to be well-suited to the given learner. On the other hand, a comparison that considers all the data sets from the three collections yields the conclusion that all techniques are equivalent on average, providing no valuable information to the practitioner. Actually, this conclusion is announced by the No-Free-Lunch (NFL) theorem[9] [Wolpert, 1992, 1996], which formally demonstrates that no learner can systematically outperform any other if all possible classification problems are contemplated.

These observations not only exhibit how tricky multiple learner comparisons are, but also how relevant the study of data set characteristics are to boost the comparative analysis. If data difficulty was characterised, we could try to identify the sweet spot in the problem complexity space where each learner actually excels the others. With this idea in mind, the next chapter introduces data complexity analysis as a basis for identifying problem complexity.

Contribution.

1. Empirical demonstration of the NFL theorem.
2. Evidence of inconclusive results about the claim of the excellence of supervised learning techniques in the literature due to a badly supported performance/comparison methodology.
3. A motivation to advocate the data complexity analysis and scrutinise its influence on supervised learning.

[9] The NFL theorem is not valid for classifiers systems such as boosting, bagging, and ensemble; just only for simple classifiers.

3 DATA HOLD THE KEY OF LEARNABLE PATTERNS

Summary. This chapter introduces data complexity in supervised leaning as an essential component to design experiments and understand learners' limitations. It shifts from theoretical estimates to practical complexity measures, where it becomes more natural to talk about approximations and their implication as well as implementations and computational efficiency. It also introduces our open source code that provides researchers with a common implementation; a software to consider for the analysis of data and the analysis of learners' behaviour.

3.1 INTRODUCTION

Machine learning, as a mature discipline, has great advantage over data complexity. However, in an era where highly-competitive learners abound in the field, the interest of some researchers has turned to go over particular machine learning techniques with a fine-tooth comb to better understand their strengths and weaknesses; and to this end, they have moved back to complexity estimates. Since the performance of these techniques depends on the data distribution and the knowledge representation used, several works have paid special attention to data and quantitatively estimate different sources of problem difficulty to investigate their influence on performance [Basu and Ho, 2006]. The construction of a complexity space is expected to reveal the bonds between data and learners—why and how data affects learners' performance. This may have a strong influence on the methodology for learner comparison by endowing it with new procedures based on previous analysis of the characteristics of the test bed.

The purpose of this chapter is to examine and discuss the state-of-the-art of data complexity and primarily focus on its practical side which can allow us to find commonality in the intricacy and interwoven nature of data.

In the following, we walk the road of data complexity from theory to practice and introduce in detail a set of complexity measures designed to evaluate the difficulty of the class boundary for classification problems [Ho and Basu, 2002]. We then present their implementation under the open source DCoL [Orriols-Puig et al., 2010] which should be empowered by multi-threaded optimisation and distributed computing to face large-scale problems and improve its efficiency.

3.2 DATA COMPLEXITY: FROM THEORY TO PRACTICE

Universal complexity of a data set is defined by the Kolmogorov complexity [Kolmogorov, 1965], which is related to several extant principles from machine learning such as Occam's razor and the Bayesian paradigm. This section describes some of these theory-based complexities, mentions some practical approaches, and announces the need for both independent and computable measures. We first formally define classification problems in supervised learning and the error of learning models. We then review theoretical complexity measures—Kolmogorov, universal data, and learning model-based complexity. Last, we present some related work about practical measures to later provide the detailed formulation of those proposed by Ho and Basu [2002], which are the backbone of the thesis.

3.2.1 *Catch up on notation and definitions*

Before addressing data complexity measures, we need to have a look at some definitions in machine learning and get used to the corresponding notation.

Let us start with the definition of machine learning phrased by Mitchell [1997, p. 2]: *"a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"*. This experience E can be seen as a *data set* or *training set* composed of instances (also referred to as examples, observations, or cases) which are described by a set of attributes (also referred to as features, variables, or dimensions). In particular, classification problems have a specific attribute, the class (also referred to as label), whose value identifies the concept or category of each instance. For many of these data sets, there is an unknown function f which is a deterministic mapping from the input space $\mathcal{X}$ (instances) to the output space $\mathcal{Y}$ (classes), the latter defined as $\mathcal{Y} \in \{0,1\}$ for binary classification problems. This mapping, called *concept*, is denoted as $f : \mathcal{X} \mapsto \{0,1\}$.

To sum up, an instance $\mathbf{z}$ is in the form of an input-output pair $(\mathbf{x}, y)$, where $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ is independently generated from an unknown probability distribution $P_{\mathcal{X}}$ and, y is computed as $y = f(x)$. Hence, the data set is formally defined as

$$\mathcal{D} : \{\mathbf{z}_i | \mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^{n}, \tag{3.1}$$

where n is the number of instances contained in the data set, i.e. the size of $\mathcal{D}$, which is denoted as $n = |\mathcal{D}|$.

In supervised learning, the *learning model* produces a hypothesis h (also referred to as prediction), from the possible set of candidates $\mathcal{H}$, whose out-of-sample error—i.e. the expected error—is defined as

$$\pi(h) = \begin{cases} 0 & \text{if } h(\mathbf{x}) = f(\mathbf{x}), \\ 1 & \text{otherwise.} \end{cases} \tag{3.2}$$

However, taking into account that the probability distribution $P_{\mathcal{X}}$ and the concept f are unknown, the learning technique has to extract the information from the training set $\mathcal{D}$ and look for a hypothesis that minimises the number of errors, which is defined as

$$e_{\mathcal{D}}(h) = \sum_{i=1}^{n} [h(\mathbf{x}_i) \neq y_i], \tag{3.3}$$

where the Boolean test $[\cdot]$ returns 1 if the condition is true and 0 otherwise.

Definition 1. *A class is apparently learnable if there is an algorithm, when given any set of labelled instances* $\mathbf{z} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$*, where* $x_i \in \mathcal{X}$ *and* $y_i \in \mathcal{Y}$*, finds a concept* $c \in \mathcal{C}$*, where* $\mathcal{C}$ *is the space of learnable concepts, that is consistent with the whole set (i.e.* $f(x_i) = y_i$ *for all* i*) or correctly indicates that there is no such concept otherwise.*

We meticulously specify that a concept is *apparently* learnable since no error does not entail a low out-of-sample error. Memorising all the instances is an example of the lack of generalisation and, as a consequence, may incur high error rates over unseen instances. Such overfitting, usually due to a learning model excessively complex or flexible, is regulated by controlling the hypotheses' complexity.

The Bayes' rule indicates that, given a training set $\mathcal{D}$, the most probable hypothesis h—Maximum A Posteriori (MAP) hypothesis—is the one with highest likelihood $\Pr\{\mathcal{D}|h\}$ and prior probability $\Pr\{h\}$ (see Eq. 3.4 and Eq. 3.5).

$$\Pr\{h|\mathcal{D}\} = \arg\max \left\{ \frac{\Pr\{\mathcal{D}|h\} \cdot \Pr\{h\}}{\Pr\{\mathcal{D}\}} \right\} \tag{3.4}$$

$$h_{\text{MAP}} = \arg\max_{h \in \mathcal{H}} \Pr\{h|\mathcal{D}\} \tag{3.5}$$

Having fewer errors results in greater likelihood, but again it is not a guarantee of high prior probability. This modification follows Occam's razor in the sense that it values simple hypotheses, which should have higher prior probabilities. The statement *entities should not be multiplied beyond* *necessity* [Li and Vitanyi, 1993] leads to define measures to estimate the simplicity or complexity of hypothesis such as Kolmogorov complexity and universal distribution.

Moreover, real-world problems hinder the hypothesis generalisation because of imperfections in the training sets. Instances can have contaminated inputs and/or outputs, which leads us to consider other layers of difficulty coined such as noise, inconsistency, missing values, and uncertainty.

3.2.2 *Universal complexities*

The Kolmogorov complexity is a universal measure that deals with complexity from a descriptive point of view. Such complexity is also related to the universal probability distribution [Solomonoff, 2003] which approximates any computable distributions. In the following, we review these complexity measures.

3.2.2.1 *Kolmogorov complexity and universal distribution*

Given a universal Turing machine $\mathcal{U}$ [Turing, 1936], the Kolmogorov complexity determines the complexity of a string s as the length of the shortest program p which is able to output s on $\mathcal{U}$, i.e.

$$K_{\mathcal{U}}(s) = \min\{|p| : \mathcal{U}(p) = s\}, \tag{3.6}$$

where $|p|$ is the length of the program p in bits.

Theorem 1. *For two universal Turing machines $\mathcal{U}_1$ and $\mathcal{U}_2$, there exists a constant c that only depends on $\mathcal{U}_1$ and $\mathcal{U}_2$ such that for any string s,*

$$K_{\mathcal{U}_1}(s) \leqslant K_{\mathcal{U}_2}(s) + c. \tag{3.7}$$

As the choice of the Turing machine just affects the complexity by a constant that depends only on $\mathcal{U}$ (see Eq. 3.7), Eq. 3.6 can be rewritten as $K(s)$, dropping the $\mathcal{U}$. Then, the Kolmogorov complexity $K(s)$ becomes a special case—empty string x—from the conditional Kolmogorov complexity, which is defined as

$$K(s|x) = \min\{|p| : \mathcal{U}(p,x) = s\}. \tag{3.8}$$

This is the length of the shortest program that outputs the string s using the auxiliary string x and it can be interpreted as the additional information in bits that are required to compute s given a known x.

Many programs can output an arbitrary string s for a universal Turing machine $\mathcal{U}$. Then, the probability that a random program prints out s is

$$P_{\mathcal{U}}(s) = \sum_{p:\mathcal{U}(p)=s} 2^{-|p|}, |p| > 0. \tag{3.9}$$

Again according to Eq. 3.7, the choice of the Turing machine is independent and only affects by a constant, so Eq. 3.9 can be rewritten as $P(s)$, which is known as the *universal distribution* or *universal prior*.

The Kolmogorov complexity can be approximated by taking the logarithm of the universal distribution: $K(s) \approx -\log P(s)$. If we then use the prior probability to rewrite the Bayes' rule from Eq. 3.4, we obtain

$$-\log \Pr\{h|\mathcal{D}\} = -\log \Pr\{\mathcal{D}|h\} - \log \Pr\{h\} + \log \Pr\{\mathcal{D}\}. \tag{3.10}$$

This means that the most probable hypothesis h given a training set $\mathcal{D}$ will minimise $-\log \Pr\{h|\mathcal{D}\}$. By representing h with a string s, $-\log \Pr\{h\}$ ($-\log \Pr\{s\}$) will be approximately the length of the program that outputs s, and $-\log \Pr\{\mathcal{D}|h\}$ will be the shortest length of the $\mathcal{D}$ given h. This corresponds to the Minimal Description Length (MDL) principle proposed by Rissanen [1997, nd].

3.2.2.2 *Universal data complexity*

As universal measures, the Kolmogorov complexity and the universal prior probability are applied to estimate the universal data complexity, which essentially consists in replicating the underlying input-output relationship of a training set.

First, the conditional Kolmogorov complexity was used to find the program with the shortest length which was able to generate the correct outputs by taking the whole set of input instances.

$$K(y_1, y_2, ..., y_n | \mathbf{z}). \tag{3.11}$$

However, this measure can perform some permutations of the instances that may mislead the amount of information needed to encode the mapping function. Then, considering that the order of the instances is not striking, the Kolmogorov complexity is defined as follows. Given a universal Turing machine $\mathcal{U}$, the data complexity of the training set $\mathcal{D}$ is

$$C_{\mathcal{U}}(\mathcal{D}) = \min\{|p| : \forall(\mathbf{x}, y) \in \mathcal{D}, \mathcal{U}(p, \mathbf{x}) = y\}. \tag{3.12}$$

$C_{\mathcal{U}}(\mathcal{D})$ provides the length of the shortest program that correctly maps the input space $\mathbf{x}$ to the corresponding output.

Unfortunately, neither of them—Kolmogorov complexity or universal distribution—are computable, as proven by Li [2006].

3.2.2.3 *Models of learning*

By an additional association, the previous reasoning was refined on machine learning. Assuming that the learning model represents all possible programs, the data complexity is considered as the length of the shortest hypothesis that captures the description of the data set. However, such an assumption is not plausible; the learning model only includes a limited set of hypotheses. Therefore, the data complexity of a training set $\mathcal{D}$ given a learning model $\mathcal{H}$ is defined as

$$C_{\mathcal{H}}(\mathcal{D}) = \min\{|h| : h \in \mathcal{H} \text{ and } \forall(\mathbf{x}, y) \in \mathcal{D}, h(\mathbf{x}) = y\}. \tag{3.13}$$

This definition of data complexity is not universal since it depends on the learning model and its encoding scheme. Certainly, for the same learning model more than one scheme can coexist. For instance, the encoding scheme of a Genetic Algorithm (GA) [Goldberg, 2002] would be the concatenation of the population size, the individual size, the crossover and mutation rates, etc. Moreover, we have to keep in mind that complexity is subject not only to the learning model constraints, but also to data characteristics—because the same parameterisation does not work for every problem.

Then, if the data set is not consistent—i.e. contains noisy instances—, some exception control has to be implemented. This means adding a program p that maintains a lookup table with the problematic instances such as

$$p = \textbf{if } \text{input is } \mathbf{x}_1, \textbf{ then } \text{output } y_1, \textbf{ else } \text{run the interpreter } p_{\mathcal{H}} \text{ on } h. \tag{3.14}$$

By including lookup tables, the definition of Eq. 3.13 can be slightly modified as follows.

$$C_{\mathcal{H},\lambda}(\mathcal{D}) = \min\{|h| + \lambda e_{\mathcal{D}}(h) : h \in \mathcal{H}\}, \tag{3.15}$$

where the constant λ corresponds to the complexity cost of one error.

However, again, these measures are either non-computable or infeasible for practical applications.

1. $C_{\mathcal{U}}(\cdot)$ is not computable because there is no effective way to decide whether a program halts and therefore find the shortest program.

2. $C_{\mathcal{H}}(\mathcal{D})$ may be computable for some $\mathcal{H}$ but finding consistent hypotheses is a NP-complete problem.

3. $C_{\mathcal{H},\lambda}(\mathcal{D})$ suffers from a high computational cost searching for the shortest hypothesis.

Thus, approximations have to be designed.

3.2.3 *Some practical complexity measures: Related work*

Without despising the relevance of theoretical estimates, we firmly believe that to tame data complexity practical measures have to be implemented.

The first attempts were made by Gamberger and Lavrac [1997] using the number of different literals required to build an *inductive learning by logic minimisation* system as a complexity measure. At the same time, Wolpert and Macready [1997] suggested a measure based on self-dissimilarities which revealed different structural patterns in data.

Later, Ho and Basu [2002] designed quantitative estimates of the boundary complexity, and more specifically, of different notions of difficulty associated with (1) overlaps in the feature values from different classes, (2) class separability, and (3) geometry, topology, and density of manifolds. Shortly after, Singh [2003] reviewed the data complexity field—from Bayes' error-based approaches to non-parametric measures [Ho and Basu, 2002]—and proposed two new indicators based on feature space partitioning: (1) purity and (2) neighbourhood separability. In turn, Mollineda et al. [2005] extended some of Ho and Basu's measures to m-class problems and defined three more measures which estimated data density: (1) the average number of instances per unit of volume (D1), (2) the average volume taken by the k-nearest neighbours of each training instance (D2), and (3) the class density in the overlap regions (D3).

Recently, Li and Abu-Mostafa [2006] studied data complexity in two learning problems: (1) data decomposition, where they observed that data are better approximated by their principal subsets and (2) data pruning, where they showed that outliers have high complexity contributions. On the other hand, Baumgartner and Somorjai [2006] defined specific measures based on singular value decomposition analysis to assess the data complexity for under-sampled classification problems. Duin and Pekalska [2006] presented margin based methods as a complementary measure to evaluate the complexity of the decision boundary and provide a description of the degree of separation between two classes. This measure attempts to change the knowledge representation so that the problem can be solved by other machine learning techniques. The idea is to simplify the problem to adapt it to simple learning schemes or detect complex structures that may demand more complex tools.

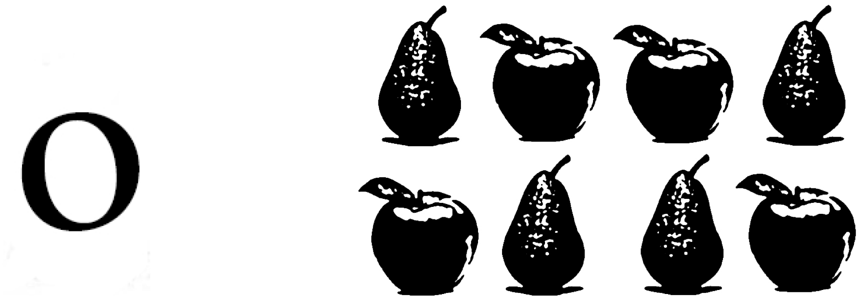
Finally, it is important to mention that there are two notions of complexity that should not be confused. For instance, Vapnik-Chervonenkis (VC) dimensions [Vapnik and Chervonenkis, 1971] correspond to the maximum number of training points that can be quartered by a set of given functions. This is a measure of the complexity/capacity of the classifier functions, not a measure of the complexity of a classification problem. Our goal is to characterise the problems to the extent that the chosen measures are suitable for, without specific reference to a fixed family of functions. For this reason, we focus on the measures proposed by [Ho and Basu, 2002] which are intended to serve as an alternative to non-computable measures such as the Kolmogorov complexity.

3.3 geometrical complexity measures in supervised learning

After whizzing around pure complexities and some approximations, we focus on the work done by Ho et al. [2006] which deals with the characterisation of the class boundary under a geometrical point view. This section introduces and extends this work. We first acquaint the reader with the different sources of problem difficulty in supervised learning and then provide the formal description of the complexity measures designed to specifically estimate the boundary complexity.

3.3.1 *Sources of problem difficulty*

The complexity of classification problems [Basu and Ho, 2006] has been attributed to three main sources: (1) class ambiguity, (2) boundary complexity, and (3) sample sparsity and feature space dimensionality. In the following, these three sources of data complexity are discussed in some detail.



(a) The shape of the letter "o" and the number "0" is the same for some fonts and cannot be distinguished without the context.

(b) The shape gives enough information to clearly differentiate between the two types of fruit. However, this task would not be possible if the features to characterise the problems would have been just the person who collected them or what time they were collected for instance.

Figure 3.1: Ambiguous classes due to (a) the class definition and (b) the lack of relevant attributes.

Class ambiguity refers to the situation where examples of different classes cannot be distinguished. This could be due to a poor capability of the selected attributes to describe the concepts that (1) are not well-defined (i.e. the attributes of the problem are not sufficient to describe the concepts which are intrinsically inseparable, see Fig. 3.1a) or (2) are well-defined, but more discriminative attributes are required (e.g. having some instances that belong to two classes, see Fig. 3.1b). This type of complexity cannot be solved at the classifier level, and data preprocessing may be needed to disambiguate the classes or the concepts. Data sets that contain classes that are ambiguous for some cases are said to have non-zero Bayes error, which sets a lower-bound on the achievable error rate.

Boundary complexity is related to the length of the description needed to represent the class. Given a complete sample, the Kolmogorov complexity is defined as the length of the shortest program that describes the class boundary. Nonetheless, the Kolmogorov complexity is known to be incomputable [Maciejowski, 1979], as seen in the previous section. Therefore, other estimates have been designed to analyse the class boundary complexity, which mainly extract different *geometrical indicators* from the data set. Fig. 3.2 illustrates several examples of boundaries. Note, moreover, that the boundary complexity is closely related to the knowledge representation used by the supervised learning techniques. Thence, the type of representation used may impose a minimum bound of the classification error. For example, linear classifiers can not easily fit curved boundaries, and so, they accumulate large errors on the class boundary; conversely, kernel-based methods can easily reproduce curved boundaries if the kernel has enough freedom to fit the boundary shape.

Finally, *sample sparsity and feature space dimensionality* aims at characterising complexities generated by regions with sparse samples in the feature space. Generalisation over empty spaces of the training data set is largely arbitrary and depends mainly on how the classifier constructs the data model. The difficulty of dealing with sparse samples in high dimensional spaces has been addressed in many works [Devroye, 1988; Raudys and Jain, 1991; Vapnik, 1998] and, some approaches expressly avoid evolving knowledge in empty regions in the feature space [Casillas et al., 2008].

Among the different sources of problem difficulties, boundary complexity has received special attention since it is the type of complexity that is more likely to be assessed. In particular, Ho and Basu [2002] designed a set of twelve measures able to extract different indicators that characterise the apparent geometrical complexity of the class boundary. These measures, which have been

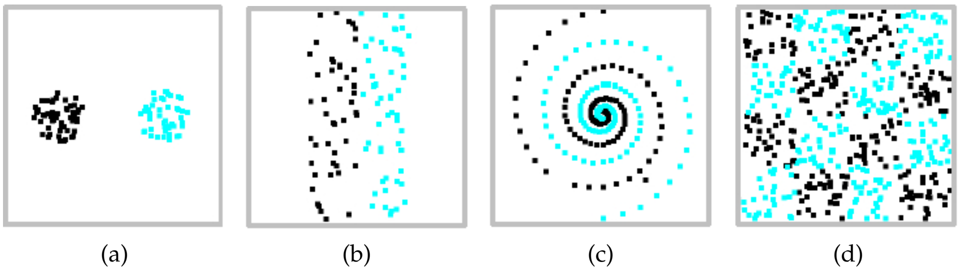


Figure 3.2: Binary classification problems with different geometrical complexity: (a) linearly separable problem with wide margins on the boundary and compact classes, (b) linearly separable problem with narrow margins on the boundary and spread classes, (c) non-linear problem, and (d) problem with highly interleaving classes following a checkerboard layout.

revised and updated from their initial definitions [Orriols-Puig et al., 2010], are explained in more detail in the next subsection.

3.3.2 *Complexity measures*

The set of complexity measures, extended to fourteen measures, can be divided into three categories: (1) measures of overlap in the feature values from different classes, (2) measures of class separability, and (3) measures of geometry, topology, and density of manifolds.

In the following, we quickly run over the description of each group and list the name of the measures that are included.

Measures of overlap in the feature values from different classes focus on the capacity of the features to separate examples of different classes. For each individual attribute, they examine the range and spread of the values of instances of different classes and check the discriminant power of a single attribute or a combination of them. This category comprises the following measures: (1) the maximum Fisher's discriminant ratio (F1), (2) the overlap of the per-class bounding boxes (F2), and (3) the maximum (individual) feature efficiency (F3). In addition, we designed two extra measures based on the previous ones: (4) the directional-vector maximum Fisher's discriminant ratio (F1v), inspired by F1, and (5) the collective feature efficiency (F4), inspired by F3.

Measures of class separability estimate to what extent the classes are separable by examining the length and the linearity of the class boundary. This category comprises the following measures: (1) the minimised sum of the error distance of a linear classifier (L1), (2) the training error of a linear classifier (L2), (3) the fraction of points on the class boundary (N1), (4) the ratio of average intra/inter class nearest neighbour distance (N2), and (5) the leave-one-out error rate of the one-nearest neighbour classifier (N3).

Measures of geometry, topology, and density of manifolds provide an indirect characterisation of the class separability. They assume that the problem is composed of several manifolds spanned by each class. The shape, position, and interconnectedness of these manifolds give some hints on how well the classes are separated and on the density or population of each manifold. This category comprises the following measures: (1) the non-linearity of a linear classifier (L3), (2) the non-linearity of the one-nearest neighbour classifier (N4), (3) the fraction of maximum covering spheres (T1), and (4) the average number of points per dimension (T2).

The next three subsubsections, corresponding to each category of complexity measures, describe all the measures and provide their formal definition.

3.3.2.1 *Measures of overlap in the feature values from different classes*

We start off with the description of the five measures that estimate different complexities related to the discriminative power of the attributes.

Maximum Fisher's discriminant ratio (F1). This measure computes the maximum discriminative power of each attribute, that is,

$$F1 = \max_{j=1}^{m} FDR_j, \tag{3.16}$$

where m is the number of input attributes, and FDR_j is the Fisher's discriminant ratio of each attribute. FDR_j is calculated differently depending on whether the data set has two classes or more than two classes.

1. For two-class data sets, the ratio for each attribute j is computed as

$$FDR_j = \frac{(\mu_1^{(j)} - \mu_2^{(j)})^2}{(\sigma_1^{(j)})^2 + (\sigma_2^{(j)})^2}, \tag{3.17}$$

   where, for continuous attributes, $\mu_k^{(j)}$ and $(\sigma_k^{(j)})^2$ are the mean and the variance of the attribute j for class k respectively. For nominal attributes, each value is mapped onto an integer number—e.g. the set {red, green, blue} would be mapped as $red = 0$, $green = 1$, and $blue = 2$[1]. Then, μ_k is the median value of the attribute j for class k and $(\sigma_k)^2$ is the variance of the attribute j for class k computed as the variance of the binomial distribution, that is,

$$\sigma_k = +\sqrt{p_{\mu_k}(1 - p_{\mu_k}) * n_k}, \tag{3.18}$$

   where p_{μ_k} is the frequency of the median value μ_k, and n_k is the total number of instances of class k.

2. For m-class data sets ($m > 2$), the ratio for each attribute j is computed as

$$f_j = \frac{\sum_{k=1}^{C} \sum_{l=k+1}^{C+1} p_k p_l (\mu_k - \mu_l)^2}{\sum_{k=1}^{C} p_k (\sigma_k)^2}, \tag{3.19}$$

   where C is the maximum number of classes and p_k is the proportion of instances of class k[2].

F1 ranges in the interval $[0, \mu_k]$. High values of the Fisher's discriminant ratio indicate that at least one of the attributes enables the learner to separate the instances of different classes with partitions that are parallel to an axis of the feature space. Low values of this measure do not imply that the classes are not linearly separable, but that they cannot be discriminated by hyperplanes parallel to one of the axis of the feature space.

The directional-vector maximum Fisher's discriminant ratio (F1v). This measure complements F1 by searching for an oriented vector which can separate instances of two different classes.

It computes the two-class Fisher's criterion [Malina, 2001], which takes the following form:

$$R(d) = \frac{[\vec{d}^T(\vec{\mu_1} - \vec{\mu_2})]^2}{\vec{d}^T \Sigma \vec{d}} = \frac{\vec{d}^T B \vec{d}}{\vec{d}^T \Sigma \vec{d}}, \tag{3.20}$$

where

[1] Indeed, this ordering implies that green is in-between red and blue, which in this case is not correct and also affects the mean computation. An alternative implementation could be to expand the nominal variables to a set of ν variables each representing one of the ν-many distinct values. The variable then will be coded as a binary vector with ν components, having a one only in the position corresponding to its original value.

[2] Although we kept the original F1 formulation for two-class problems, such proportional weights for classes of different sizes may also be needed for this measure if the classes are not balanced.

- $\vec{d}$ is the directional vector on which data are projected
- $\vec{\mu}_k$ is the mean vector of class k
- $\overline{\Sigma} = \alpha\Sigma_1 + (1-\alpha)\Sigma_2,\ 0 \leqslant \alpha \leqslant 1$
- Σ_i is the scatter matrix of patterns for class k
- $B = (\vec{\mu}_1 - \vec{\mu}_2)(\vec{\mu}_1 - \vec{\mu}_2)^T$ is the between class scatter matrix

The directional vector $\vec{d}$ is calculated as

$$\vec{d} = \overline{\Sigma}^{-1}\Delta, \tag{3.21}$$

where $\Delta = \vec{\mu}_1 - \vec{\mu}_2$ and $\overline{\Sigma}^{-1}$ is computed as the pseudo inverse [Moore, 1920; Penrose, 1955] of $\overline{\Sigma}$.

This measure is only implemented for two-class problems.

F1v ranges in the interval $[0, \mu_k]$. High values of the directional-vector maximum Fisher's discriminant ratio indicate that there exists a vector that can separate instances belonging to different classes after these instances are projected onto it.

The overlap of the per-class bounding boxes (F2). This measure computes the overlap of the tails of distributions defined by the instances of each class.

The definition of this measure for two-class data sets is the following. For each attribute, it computes the ratio of the width of the overlap interval (i.e. the interval that has instances of both classes) to the width of the entire interval (see Fig. 3.3). Then, the measure returns the product of the ratios calculated for each attribute, which is defined as

$$F2 = \prod_{j=1}^{m} \frac{MIN\_MAX_j - MAX\_MIN_j}{MAX\_MAX_j - MIN\_MIN_j}, \tag{3.22}$$

where m is the number of input attributes and,

$$MIN\_MAX_j = \min(\max(j,1), \max(j,2)), \tag{3.23}$$

$$MAX\_MIN_j = \max(\min(j,1), \min(j,2)), \tag{3.24}$$

$$MAX\_MAX_j = \max(\max(j,1), \max(j,2)), \text{ and} \tag{3.25}$$

$$MIN\_MIN_j = \min(\min(j,1), \min(j,2)), \tag{3.26}$$

where max(j,k) and min(j,k) are, respectively, the maximum and minimum values of the attribute j for class k. Nominal values are mapped to integer values (for details, see Sect. 3.3.2.1 on page 28) to compute this measure.

For m-class data sets ($m > 2$), we compute F2 for each pair of classes, get the absolute value of all them, and return the sum of all these values.

F2 ranges in the interval $[0, 1]$. Low values of this measure mean that the attributes can discriminate the instances of different classes.

The maximum (individual) feature efficiency (F3). This measure computes the discriminative power of individual features and returns the value of the attribute that can discriminate the largest number of training instances.

For this purpose, the following heuristic—a heuristic of local continuity—is employed. For each attribute, we consider the overlapping region (i.e. the region where there are instances of both classes) and return the ratio of the number of instances that are not in this overlapping region to the total number of instances (see Fig. 3.4). Then, the maximum discriminative ratio is taken as measure F3.

Note that a problem is easy if there is one attribute for which the ranges of the values spanned by each class do not overlap (in this case, this would be a linearly separable problem). F3 ranges in the interval $[0, 1]$. High values of this measure indicate that there is an attribute which is able to discriminate between instances of different classes.

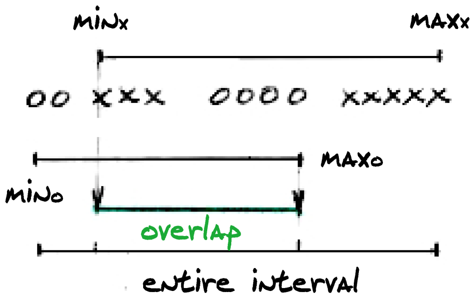
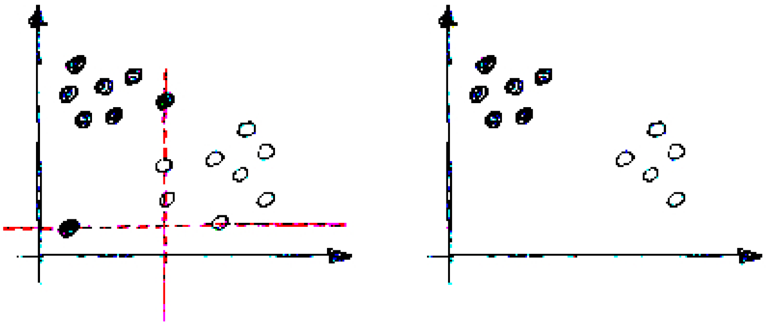


Figure 3.3: Example of overlap interval for one dimension.



(a)

(b)

Figure 3.4: Example of data set disambiguation: (a) original set of samples and (b) modified set.

The collective feature efficiency (F4). This measure follows the same idea presented by F3, but now it considers the discriminative power of all the attributes—therefore, the *collective* feature efficiency.

To compute the collective discriminative power, we apply the following procedure. First, we select the most discriminative attribute, i.e. the attribute that can separate the majority of instances of one class. Then, all the instances that can be discriminated are removed from the data set, and the following most discriminative attribute (with regards to the remaining instances) is selected. This procedure is repeated until all the instances are discriminated or all the attributes in the feature space are analysed. Finally, the measure returns the proportion of instances that have been discriminated. Thus, it gives us an idea of the fraction of instances whose class could be correctly predicted by building separating hyperplanes that are parallel to one of the axes in the feature space.

Note that the measure described herein differs slightly from F3, which only considers the number of instances discriminated by the most discriminative attribute, instead of all the attributes. Thence, F4 provides more information by taking into account all the attributes, and highlighting the collective discriminative power.

F4 ranges in the interval $[0,1]$. Like F3, high values of this measure indicate that there is an attribute which is able to discriminate between instances of different classes.

3.3.2.2 *Measures of class separability*

In the following, we describe five measures that examine the shape of the class boundary, to estimate the complexity of separating instances of different classes.

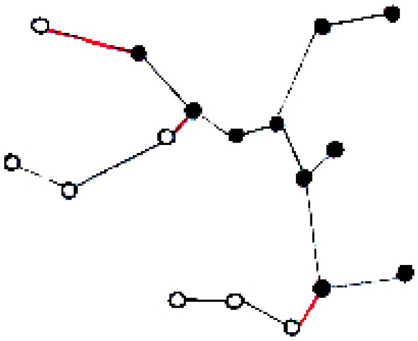


Figure 3.5: Example of an MST. Red lines connect instances that belong to different classes. The sum of these connections is divided by the total number of instances and taken as measure N1.

The minimised sum of the error distance of a linear classifier (L1). This measure determines the linear separability.

For this purpose, it returns the sum of the differences between the prediction of a linear classifier and the actual class value in test formulated as

$$L1 = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n_t}, \tag{3.27}$$

where $\hat{y}_i$ is the class predicted by the linear classifier for instance i, y_i the actual class value of instance i, n is the number of instances in the original data sets, and n_t is the number of instances of the test set. Different from Ho and Basu [2002], in our implementation we use an Support Vector Machine (SVM) [Vapnik, 1995] with a linear kernel, which is trained with the SMO algorithm [Platt, 1999]. We use this learner since the SMO algorithm provides an efficient training method, and the result is a linear classifier that separates the instances of two classes by means of a hyperplane. L1 is highly conditioned by outliers that lie on the wrong side of the best approximation of the hyperplane. This measure is only implemented for two-class problems.

L1 ranges in the interval $[0, n/n_t]$. A zero value of L1 indicates that the problem is linearly separable. In that case, we assume that the problem is simple since it can be solved by a linear classifier. However, not being linearly separable does not imply that the problem cannot be tackled by this type of classifier, which may obtain a solution with low classification error.

The training error of a linear classifier (L2). This measure provides information about to what extent the training data is linearly separable. It builds the linear classifier as explained above and returns its training error defined as the percentage of incorrectly classified instances. As before, the measure is only implemented for two-class data sets. L2 ranges in the interval $[0, 1]$. Low values of this measure indicate that there is a large gap in the class boundary.

The fraction of points on the class boundary (N1). This measure, inspired by the test proposed by Friedman and Rafsky [1979] gives an estimate of the length of the class boundary.

For this purpose, it builds a Minimum Spanning Tree (MST) over the entire data set by first connecting all the points using the Euclidean distance. It returns the ratio of the number of nodes of the spanning tree that connect different classes to the total number of instances in the data set (see Fig. 3.5). If a node n_i is connected to more than one node of a different class, n_i is counted only once.

N1 ranges in the interval $[0, 1]$. High values of this measure indicate that the majority of the points are located near the class boundary, and so, that it may be more difficult for the learner to model this class boundary accurately.

The ratio of average intra/inter class nearest neighbour distance (N2). This measure compares the intra-class spread to the inter-class spread.

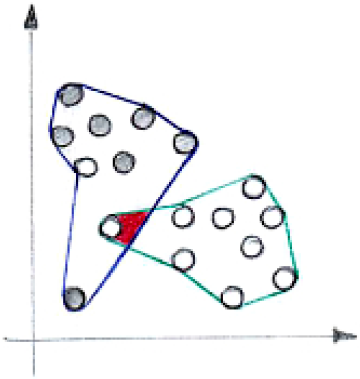


Figure 3.6: Example of an overlap region obtained by L3.

For each input instance $\mathbf{x}_i$, we calculate the distance to its nearest neighbour within the class ($\mathtt{intraDist}(\mathbf{x}_i)$) and the distance to its nearest neighbour of any other class ($\mathtt{interDist}(\mathbf{x}_i)$). Then, the result is the ratio of the sum of the intra-class distances to the sum of the inter-class distances for each input instance, i.e.

$$N2 = \frac{\sum_{i=1}^{n} \mathtt{intraDist}(\mathbf{x}_i)}{\sum_{i=1}^{n} \mathtt{interDist}(\mathbf{x}_i)}, \tag{3.28}$$

where n is the number of instances in the data set.

N2 ranges in the interval $[0, +\infty]$. Low values of this measure suggest that the instances of the same class lie closely in the feature space. High values indicate that the instances of the same class are disperse.

The leave-one-out error rate of the one-nearest neighbour classifier (N3). The measure denotes how close the instances of different classes are. It returns the leave-one-out error rate of the one-nearest neighbour (the k-Nearest Neighbour (k-NN) classifier with $k = 1$).

N3 ranges in the interval $[0, 1]$. Low values of this measure indicate that there is a large gap in the class boundary.

3.3.2.3 *Measures of geometry, topology, and density of manifolds*

Having seen a set of measures that estimate the shape of the class boundary, we now explicate four measures that indirectly characterise the class separability by assuming that a class is made up of single and multiple manifolds that form the support of the distribution of the class.

The non-linearity of a linear classifier (L3). This measure implements a measure of non-linearity proposed by Hoekstra and Duin [1996].

Given the training data set, the method creates a test set by linear interpolation with random coefficients between pairs of randomly selected instances of the same class. Then, the measure returns the test error rate of the linear classifier (the SVM with a linear kernel) trained with the original set. The measure is sensitive to the smoothness of the classifier boundary and the overlap of the convex hulls of the classes (see Fig. 3.6). This measure is only implemented for two-class data sets.

L3 ranges in the interval $[0, 1]$. High values of this measure express a high interleaving between classes.

The non-linearity of the one-nearest neighbour classifier (N4). This measure, proposed by Hoekstra and Duin [1996], creates a test set as proposed by L3 and returns the test error of the 1-NN classifier after using the original data set as training.

N4 ranges in the interval $[0, 1]$. High values of this measure express a high interleaving between classes.

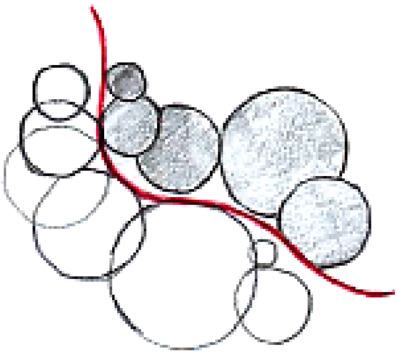


Figure 3.7: Example of adherence subsets required to describe the class boundary between two classes.

Table 3.1: Summary of the complexity measures.

Label	Complexity measure	Bounds	m-class
F1	Maximum Fisher's discriminant ratio	$[0, \mu_k]$	✔
F1v	Directional-vector maximum Fisher's discriminant ratio	$[0, \mu_k]$	
F2	Overlap of the per-class bounding boxes	$[0, 1]$	✔
F3	Maximum (individual) feature efficiency	$[0, 1]$	✔
F4	Collective feature efficiency	$[0, 1]$	✔
L1	Minimised sum of the error distance of a linear classifier	$[0, n/n_t]$	
L2	Training error of a linear classifier	$[0, 1]$	
L3	Nonlinearity of a linear classifier	$[0, 1]$	
N1	Fraction of points on the class boundary	$[0, 1]$	✔
N2	Ratio of average intra/inter class nearest neighbour distance	$[0, +\infty]$	✔
N3	Leave-one-out error rate of the one-nearest neighbour classifier	$[0, 1]$	✔
N4	Nonlinearity of the one-nearest neighbour classifier	$[0, 1]$	✔
T1	Fraction of maximum covering spheres	$[0, 1]$	✔
T2	Average number of points per dimension	$[1, n]$	✔

The fraction of maximum covering spheres (T1). This measure originated in the work of Lebourgeois and Emptoz [1996], which described the shapes of class manifolds with the notion of an *adherence subset*. Simply speaking, an adherence subset is a sphere centred on an instance of the data set which is grown as much as possible before touching any instance of another class. Therefore, an adherence subset contains a set of instances of the same class and cannot grow more without including instances of other classes. The measure considers only the biggest adherence subsets or spheres, removing all those that are included in others. Then, the measure returns the number of spheres normalised by the total number of points (see Fig. 3.7).

T1 ranges in the interval $[0, 1]$. Low values of this measure means that the instances are grouped in compact clusters.

The average number of points per dimension (T2). This measure returns the ratio of the number of instances in the data set to the number of attributes. It is a rough indicator of sparseness of the data set. T2 ranges in the interval $[0, n]$, where n is the number of instances.

Table 3.1 summarises the descriptive information of the complexity measures detailed above. It specifically gathers for each measure: its label, bounds, and whether it can be applied to m-class data sets. These complexity measures are far from exhaustive, but interestingly, their use has turned the understanding of learners' behaviour on its head. This motivates us to provide the research community with a fair implementation of the measures, which they can use in their analyses. It is also desirable since the implementation details of the measures are not supplied by the seminal papers.

3.4 DCOL: DATA COMPLEXITY LIBRARY

Providing a public and common implementation may help researchers to create their own use of these measures and compare their achievements. This section gives the details of our implementation DCoL, made available as open source. After a quick look at the literature, we report the features of the library and draw some further improvements related to the computational efficiency.

3.4.1 *Whence DCoL?*

By sweeping the literature, we notice that complexity measures have been used to: (1) study the sources of problem difficulty that affect particular learners [Bernadó-Mansilla and Ho, 2005; Luengo and Herrera, 2010], (2) compare different learners on collections of problems of bounded difficulty [Orriols-Puig and Casillas, 2010], and (3) guide data preprocessing techniques [Luengo et al., 2010]. These works show that complexity measures have been instrumental to gain understanding of learners behaviour in different domains and to identify their strengths and weaknesses. Nevertheless, despite the added value of the complexity measures, there is still a factor that may hinder researchers from applying them: the difficulty of implementing some of the measures from their original, more conceptual description. The aim of the library, therefore, is to provide a standard, flexible implementation of the different complexity measures that researchers can use for their analyses. So far, the library accumulates a total of 250 downloads since 2009 (for details, see Appendix B).

3.4.2 *SourceForge release*

DCoL provides the implementation of a set of measures designed to characterise the apparent complexity of data sets for supervised learning, which were originally proposed by Ho and Basu [2002]. More specifically, the implemented measures focus on the complexity of the class boundary and estimate (1) the overlaps in the feature values from different classes, (2) the class separability, and (3) the geometry, topology, and density of manifolds. In addition, two other complementary functionalities, (4) stratified k-fold partitioning and (5) routines to transform m-class data sets ($m > 2$) into m two-class data sets, are included in the library. The source code can be compiled across multiple platforms (Linux, MacOS X, and MS Windows) and can be easily configured and run from the command line interface. The latest version (1.1) of the data complexity library in C++ is available at `http://dcol.sourceforge.net/`.

All of these measures were initially designed for two-class data sets described by continuous attributes. However, they have been extended to deal with (1) multiple classes, (2) nominal attributes, (3) missing values, and (4) standard input file formats.

3.4.2.1 *Dealing with multiple-class data sets*

Originally, all the complexity measures were defined for two-class data sets. To extend the measures to multiple-class data sets, we followed the suggestions provided by Ho et al. [2006]. Thus, we redefined all these measures, except the three that involve the construction of a SVM and the directional-vector maximum Fisher's discriminant ratio, to m-class data sets (for details, see Sect. 3.3.2). Notwithstanding, applying these measures to m-class data sets may preclude some key observations on the complexity relative to individual classes. We recommend then to investigate the complexity in the context of binary classification problems. This leads us to supply a complementary function: an automatic transforming method. When this option is specified and a data set with more than two classes is fed to the application, we transform the m-class data set into m two-class data sets by creating a new two-class data set for each class, which contains all the instances of one class and groups all the remaining instances in a new, different class. Therefore, each data set is lined up with the complexity of a single class, considering that we may map each different learning concept onto a class.

3.4.2.2 *Dealing with nominal attributes*

To manage both continuous and nominal/categorical attributes, we considered the most relevant distance functions for these types of attributes [Wilson and Martinez, 1997]—functions which have been shown to enhance the behaviour of instance-based learners. For continuous attributes, we implemented (1) the Euclidean distance function, (2) the normalised Euclidean distance function, and (3) the Euclidean distance function weighted by the standard deviation. For nominal attributes, we implemented (1) the overlap distance function and (2) the Value Difference Metric (VDM).

Regardless of the type of the attributes, the distance between two instances $\mathbf{z}_1$ and $\mathbf{z}_2$ is calculated as

$$\mathrm{distance}(\mathbf{z}_1,\mathbf{z}_2) = \sqrt{\sum_{j=1}^{m} [d(x_j^{(1)},x_j^{(2)})]^2}, \tag{3.29}$$

where $x_j^{(i)}$ is the value of attribute j for the instance i, m is the number of input attributes, and d is the function that computes the distance between two values of the attribute j.
For *continuous attributes*, we implemented the following distance functions:

The Euclidean distance function is defined as

$$d(x_j^{(1)},x_j^{(2)}) = x_j^{(1)} - x_j^{(2)}, \tag{3.30}$$

where $x_j^{(i)}$ refers to the value of the attribute j of the instance i. Note that we do not return the absolute value of the substraction.

The normalised Euclidean distance function (i.e. the distance of each attribute is normalised by its range) is defined as

$$d(x_j^{(1)},x_j^{(2)}) = \frac{(x_j^{(1)} - x_j^{(2)})}{\mathrm{range}_j}, \tag{3.31}$$

where range_j is the range of attribute j.

The Euclidean distance function weighted by the standard deviation is defined as

$$d(x_j^{(1)},x_j^{(2)}) = \frac{(x_j^{(1)} - x_j^{(2)})}{4\sigma_j}, \tag{3.32}$$

where σ_j is the standard deviation of the attribute j. This metric is said to be less sensitive to outliers. Since 95% of values in a normal distribution fall within two standard deviations (2σ) of the mean, the difference between numeric values is divided by 4 standard deviations to scale each value into a range whose width is usually 1.
For *nominal attributes*, we implement the following distance functions:

The overlap distance function is defined as

$$d(x_j^{(1)},x_j^{(2)}) = \begin{cases} 0 & \text{if } x_j^{(1)} = x_j^{(2)}, \\ 1 & \text{otherwise.} \end{cases} \tag{3.33}$$

If two nominal attributes are equal, the distance between them is 0. Otherwise, the distance is 1.

The VDM distance function [Stanfill and Waltz, 1986] is defined as

$$d(x_j^{(1)}, x_j^{(2)}) = \sum_{k=1}^{C} |Pr(j, x_j^{(1)}, k) - Pr(j, x_j^{(2)}, k)|, \tag{3.34}$$

where C is the total number of classes in the problem and $Pr(j, x_j^{(}i), k)$ is the conditional probability that the output class is k given that the attribute j has the value $x_j^{(i)}$. That is, $Pr(j, x_j^{(i)}, k)$ is defined as

$$Pr(j, x_j^{(i)}, k) = \frac{N(j, x_j^{(i)}, k)}{N(j, x_j^{(i)})}, \tag{3.35}$$

where $N(j, x_j^{(1)}, k)$ is the number of instances of class k for which the attribute j takes the value $x_j^{(i)}$, and $N(j, x_j^{(i)})$ is the total number of instances for which the attribute j takes the value $x_j^{(i)}$.

The application also permits mapping the nominal values onto integers (for detail, see Sect. 3.3.2.1 on page 28) and applying a Euclidean distance or a normalised Euclidean distance. Note that by combining the normalised Euclidean distance for continuous attributes with the overlap distance for nominal attributes, we obtain the Heterogeneous Euclidean-Overlap Metric (HEOM) [Aha et al., 1991; Giraud-Carrier and Martinez, 1995]. Moreover, the combination of the normalised Euclidean distance for continuous attributes with the VDM distance for nominal attributes results in the Heterogeneous Value Difference Metric (HVDM) [Wilson and Martinez, 1997].

3.4.2.3 *Dealing with missing values*

As some measures do not accept missing values, we implement the following policy to replace them. For real- and integer-valued attributes, a missing value in the attribute j of an instance that predicts class k is replaced with the average value of the attribute j computed on all the instances that predict class k. For nominal attributes, it is replaced with the median value of the attribute j computed on all the instances that predict class k.

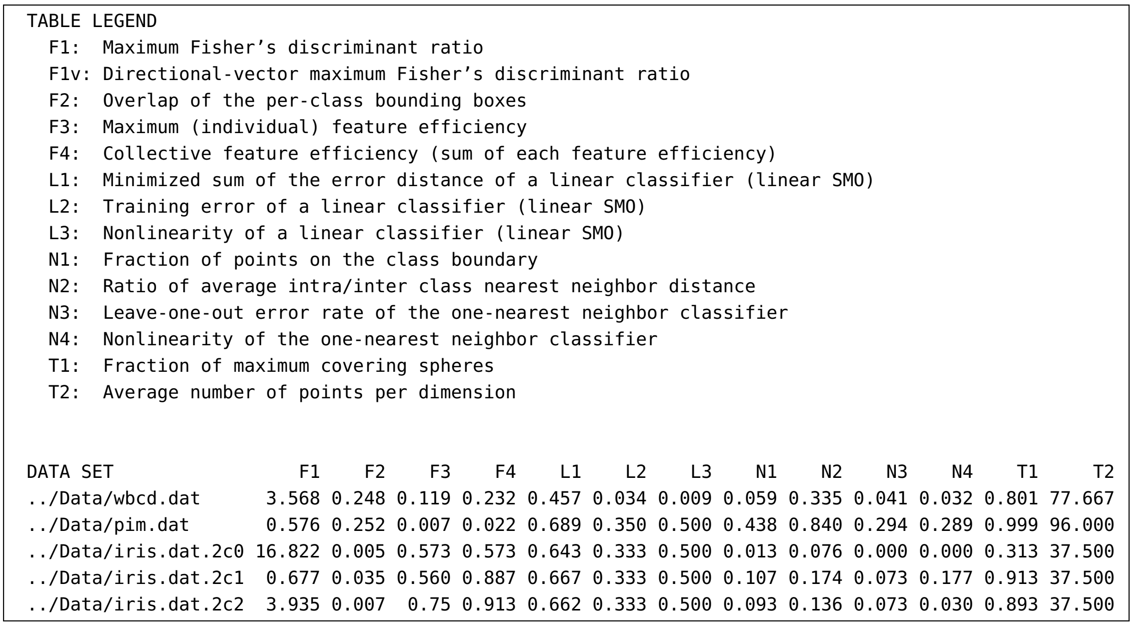
3.4.2.4 *Dealing with standard input file formats*

The application requires an input file which must contain either (1) the data set for a single run or (2) a list of paths for a multiple run with different data sets. The input data sets have to follow either the Weka format [Witten and Frank, 2005] or the KEEL format [Alcalá-Fdez et al., 2008] which is an extension of the Weka's. The data loader checks the syntactic and semantic correctness of the data sets. This is useful to detect anomalies in the data such as attributes with a constant value (i.e. irrelevant attributes) or inconsistent definition of the problem (i.e. defined classes with no instances in the sample set).

3.4.2.5 *Other functionalities*

In order to increase the functionalities of the software, some additional functions such as automatic transformation of m-class data sets into m two-class data sets and k-fold cross-validation partitioning have been included in the library.

Transformation. This functionality requires that the input data set has m classes (m >2). It generates m new data sets (one for each class), which consist of instances of one class against instances of all the other classes. Therefore, it creates two-class data sets in which each concept is discriminated against the rest of the domain.



```
TABLE LEGEND
  F1:  Maximum Fisher’s discriminant ratio
  F1v: Directional-vector maximum Fisher’s discriminant ratio
  F2:  Overlap of the per-class bounding boxes
  F3:  Maximum (individual) feature efficiency
  F4:  Collective feature efficiency (sum of each feature efficiency)
  L1:  Minimized sum of the error distance of a linear classifier (linear SMO)
  L2:  Training error of a linear classifier (linear SMO)
  L3:  Nonlinearity of a linear classifier (linear SMO)
  N1:  Fraction of points on the class boundary
  N2:  Ratio of average intra/inter class nearest neighbor distance
  N3:  Leave-one-out error rate of the one-nearest neighbor classifier
  N4:  Nonlinearity of the one-nearest neighbor classifier
  T1:  Fraction of maximum covering spheres
  T2:  Average number of points per dimension


DATA SET                 F1    F2    F3    F4    L1    L2    L3    N1    N2    N3    N4    T1     T2
../Data/wbcd.dat      3.568 0.248 0.119 0.232 0.457 0.034 0.009 0.059 0.335 0.041 0.032 0.801 77.667
../Data/pim.dat       0.576 0.252 0.007 0.022 0.689 0.350 0.500 0.438 0.840 0.294 0.289 0.999 96.000
../Data/iris.dat.2c0 16.822 0.005 0.573 0.573 0.643 0.333 0.500 0.013 0.076 0.000 0.000 0.313 37.500
../Data/iris.dat.2c1  0.677 0.035 0.560 0.887 0.667 0.333 0.500 0.107 0.174 0.073 0.177 0.913 37.500
../Data/iris.dat.2c2  3.935 0.007  0.75 0.913 0.662 0.333 0.500 0.093 0.136 0.073 0.030 0.893 37.500
```

Figure 3.8: Output of the command `./dcol -i ./list2.dat -o ./Output/list2 -B -A -d`

Partitioning. This functionality runs a stratified cross-validation with *numFolds* folds (where *numFolds* $>$ 1). If *numFolds* is not specified or incorrectly set (i.e. *numFolds* $<$2), it is set to ten by default.

3.4.2.6 *Format of the output*

The output depends on the type of run.

- For runs concerning complexity measures, the application writes an output file which specifies the value for each measure. In the next section there are several examples of output files.

- For runs concerning options of transformation—`-t2class`—and partitioning—`-cv`—, different files are created with the corresponding output data sets.

By default the system saves the computation of the complexity measures in a `.txt` format file (see Fig. 3.8). There are two more formats available: LaTeX(`.tex`) and XML (`.xml`). In any case, information of the process will be displayed on the screen.

Finally, warnings and errors will be reported in file `<output_file>.log` where `<output_file>` is the output file name specified by the user.

Orriols-Puig et al. [2010] supplies more detailed and additional information about the options of the library and how to download, compile, and run the code.

3.4.3 *The structure of the code*

The object-oriented design is composed of three group of classes: (1) application core classes, (2) distance function classes, and (3) auxiliary classes. Fig. 3.9 provides the class diagram. It is worth highlighting that the code does not use third party libraries for compatibility across different architectures.

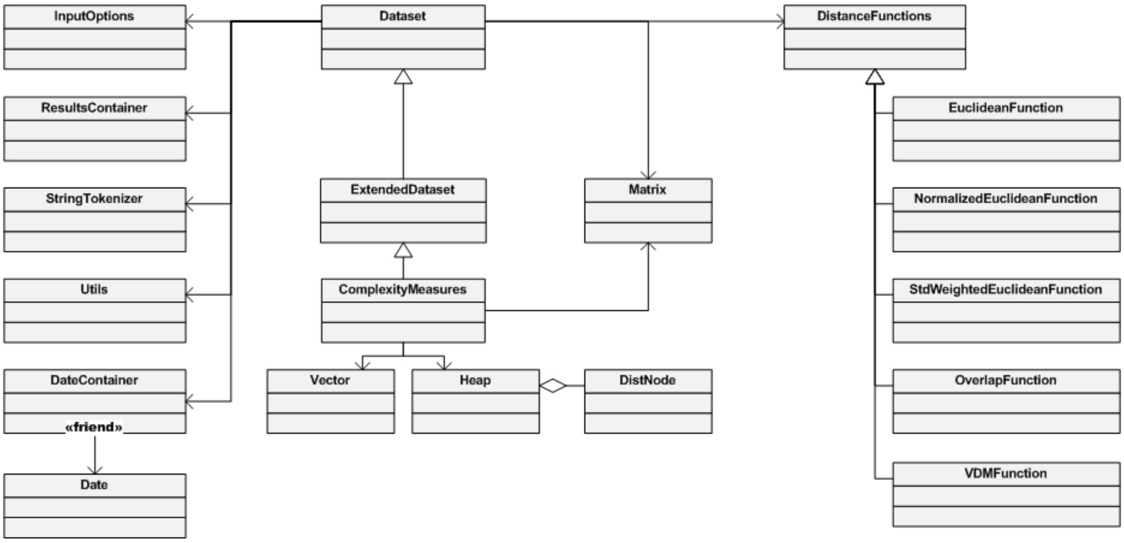


Figure 3.9: Class diagram.

Table 3.2: Computational cost of the complexity measures. n is the number of input instances, n_t is the number of test instances (applicable only to the measures that generate an additional test set), m is the number of attributes, and c is the number of classes. $O(SMO)$ is the cost of building a support vector machine with linear kernel by means of the SMO algorithm.

Label	Complexity measure	Computational cost
F1	Maximum Fisher's discriminant ratio	$O(n \cdot m)$
F1v	Directional-vector maximum Fisher's discriminant ratio	$O(n \cdot m + m^3)$
F2	Overlap of the per-class bounding boxes	$O(n \cdot m)$
F3	Maximum (individual) feature efficiency	$O(n \cdot m \cdot c)$
F4	Collective feature efficiency	$O(n \cdot m^2 \cdot c)$
L1	Minimised sum of the error distance of a linear classifier	$O(SMO)$
L2	Training error of a linear classifier	$O(SMO)$
L3	Nonlinearity of a linear classifier	$O(SMO + n_t \cdot m \cdot c)$
N1	Fraction of points on the class boundary	$O(n^2 \cdot m)$
N2	Ratio of average intra/inter class nearest neighbour distance	$O(n^2 \cdot m)$
N3	Leave-one-out error rate of the one-nearest neighbour classifier	$O(n^2 \cdot m)$
N4	Nonlinearity of the one-nearest neighbour classifier	$O(n_t \cdot a \cdot c + n \cdot n_t \cdot m)$
T1	Fraction of maximum covering spheres	$O(n^2 \cdot m)$
T2	Average number of points per dimension	$O(1)$

3.4.4 *Computational efficiency*

Complexity measures are essential to build a bridge between learners' properties and data characteristics, but they may pose certain difficulties because of their computational cost. Table 3.2 summarises the computational cost of each complexity measure.

In the following, we elaborate the less obvious costs.

L1, L2. Their computational cost is guided by the cost of building the SVM (for further details, see [Platt, 1999]).

L3. Its computational cost is guided by the cost of building the SVM plus the cost of generating the test set which increases linearly with the number of test instances, the number of classes, and the number of attributes.

N1. Its computational cost is determined by the cost of building the MST, which grows with the product of the number of instances power two and the number of attributes.

N4. Its computational cost depends on the sum of (1) the cost of generating the test instances and (2) the cost of computing the k-NN algorithm. The former increases linearly with the product of the number of test instances that are generated, the number of attributes, and the number of classes. The latter depends linearly on the product of the number of training instances, the number of test instances, and the number of attributes.

In the next section, we outline future enhancement to face new trends regarding data streams and large-scale problems by implementing multi-threaded optimisation and distributed computing strategies—in particular, MapReduce [Dean and Ghemawat, 2010].

3.4.5 *Further work*

To improve the efficiency, the plan involves a two-pronged attack: (1) multi-threaded optimisation and (2) distributed computing.

Multi-threaded optimisation. Although the library offers the possibility of individually running each complexity measure, the main option is to sequentially compute the whole set. This should be enhanced by running each measure in parallel, which means to code groups of complexity measures in different threads. The composition of these groups has, however, to consider over which bits of data each complexity measure operates to efficiently coordinate the tasks and avoid bottlenecks in data access. Such drawback could be easily solved by storing the data in a distributed and replicated file system [Stonebraker et al., 2010]. Thus, we are drawn to a profitable situation that allows us to benefit from the latest distributed computing techniques as elaborated below.

Distributed computing. Replication has been designed to overcome classical constraints of centralised systems and provide high scalability when dealing with massive quantities of data [Krikellas et al., 2010]. There are many tools that take the most out of distributed file systems against the decline of consistency. Although as the most data are replicated the performance tends to be zero [Brewer, 2000; Jiménez-Peris et al., 2002], techniques from the cloud paradigm have maximised the scalability. Hadoop is becoming a very appealing tool since it incorporates distributed file systems which automatically manage replication and the MapReduce processor.

Specifically, MapReduce [Dean and Ghemawat, 2010] has emerged as a platform for data intensive computation taking advantage of the classical *divide&conquer* strategy, now *recoined* as *map&reduce*. These two functions permit distributing data across thousands of machines and processing them in a reasonable amount of time, a distribution that implies parallel computing in the form that each CPU performs calculations on different chunks of data. *Map* invocations are spread across the system by the master node, which loads the input data and preprocesses them by automatically cutting them up into sub-jobs to be executed. *Reduce* invocations are the aggregation of partial results and the communication of the final output. Fig. 3.10 reflects the flow of a MapReduce operation. The master node picks idle workers and assigns either map tasks or reduce tasks. All the maps need to finish before the reduce stage can begin since the workers have to access all the values with the same identifier (ID) to perform sequential computations with them. The parallelism is exploited by simultaneously executing the task for the different IDs.

Back to the DCoL and profiling its code, we detected that most of the computational time of N1 is on the construction of the MST which is implemented by the Prim's algorithm [Prim, 1957]; the cost of this standard sequential implementation is $O(n^2)$ where n the number of vertices.

This observation leads to look into the strand of parallel computing and consider the computation model for MapReduce proposed by Karloff et al. [2010], which can compute the MST of a

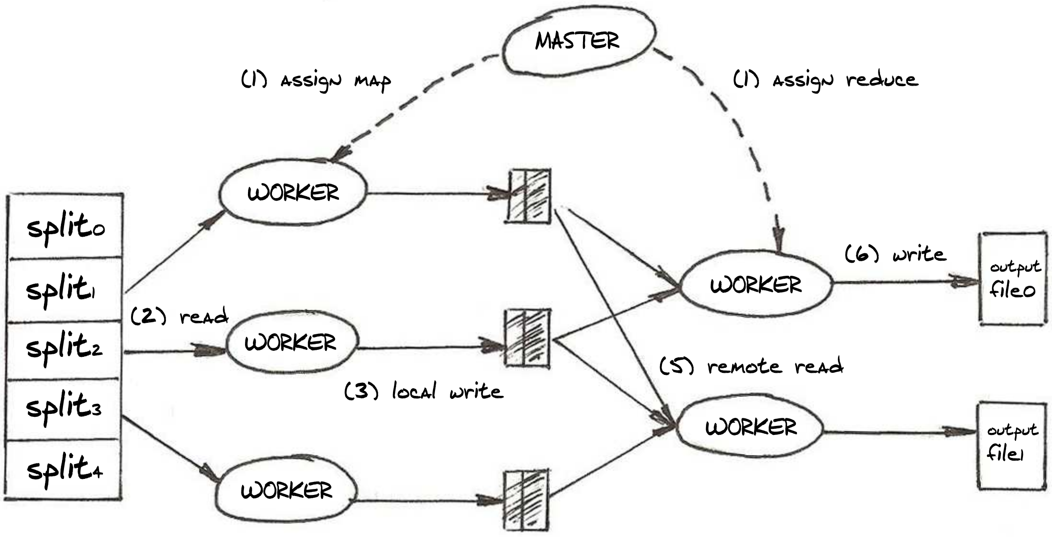


Figure 3.10: Overview of the flow of a MapReduce operation.

dense graph in two rounds—a value more conspicuous than the $\log(n)$ rounds needed in the *parallel random access machine* model.

In fact, the strength of MapReduce is the interleaving of parallel and sequential computation plus the execution on a Hadoop cluster, something that frees us from the scant resources of supercomputers—where we actually carry out our calculations—, the internal configuration of the cluster, and the exact locality of the data.

Finding the MST with MapReduce requires us to divide the process into two steps: (1) generate the similarity matrix of the data set and (2) find the MST in the undirected complete graph, which corresponds to the similarity matrix. Figures 3.11a and 3.11b synthesise steps 1 and 2 respectively. For more details, this approach has been successfully implemented by Chang et al. [2010]. However, we are still vigilant since the testing on large data was based on samples composed of 20,000 objects whereas our experimentation handles problems with up to 250,000 instances and 380 attributes (see Chapter 5).

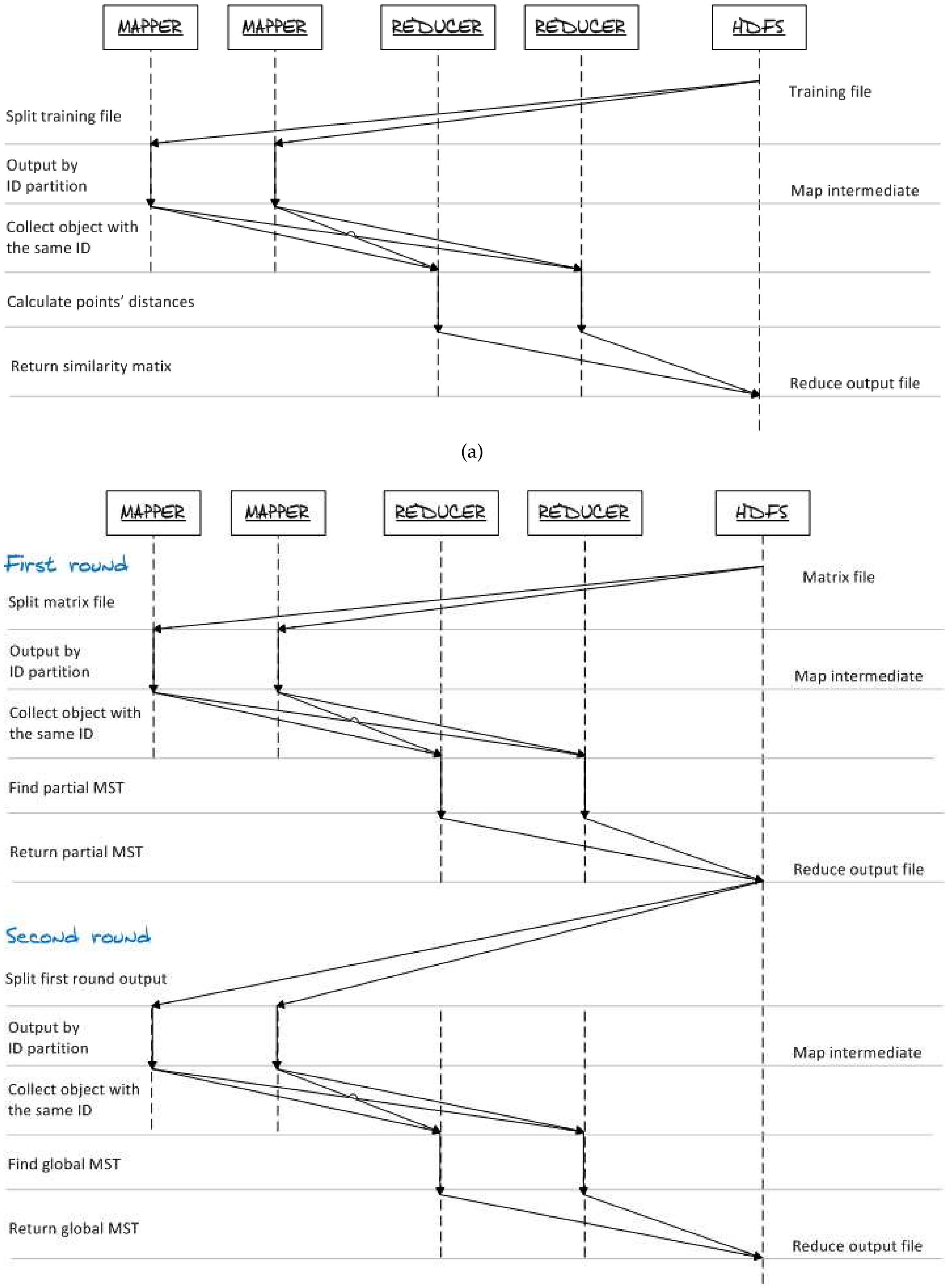
On the other hand, DCoL is implemented in C++ and the open source implementation of Hadoop, in Java. Although there is a package—Hadoop Pipes[3]—, that allows C++ code to use Hadoop DFS and MapReduce, the library translation into Java is not an issue since its basic version is already included in the KEEL software [Alcalá-Fdez et al., 2011]. However, it will be important to measure the gain of the parallelisation and an interpreted language.

On a side note, we admit that there is a big gap between the single-threaded Prim's algorithm—taught in Computer Science 101—and MapReduce for high-performance computing. Intermediate solutions based on multi-threaded implementations such as the one proposed by Bader and Cong [2006] for the MST algorithm should also be developed and studied.

Finally, for further research a pending task is to rethink the complexity measure algorithms in terms of map and reduce operations as well as to connect DCoL to MapReduce and database management systems. This latter suggestion, inspired by Stonebraker et al. [2010], has more to do with broadness of scope than efficiency. It is meant not to restrict the computation of the complexity measures to arff files and be able to directly attack parallel databases.

Our ideal scenario would be to harness the power of the Hadoop framework, as its distributed file system spreads and replicates data across multiple commodity storage servers in a highly scalable way. This would allow us to arbitrarily execute many instances of map and reduce

[3] http://hadoop.apache.org/common/docs/r0.20.2/api/org/apache/hadoop/mapred/pipes/package-summary.html



(a)

(b)

Figure 3.11: Finding the MST with MapReduce in two steps: (a) generation of the similarity matrix and (b) construction of the MST.

functions which would locally access the data. Otherwise, with a conventional database such as MySQL, the data would be remotely retrieved increasing computational cost.

As a notice, we should bear in mind that these solutions, in general, reduce the computational cost to the detriment of legibility of the code and its maintenance.

3.5 COMPLEXITY MEASURES: A REFEREE ELEMENT

Although this set of complexity measures is still preliminary and under study, they provide an estimate of the problem difficulty without suffering the dependence from the knowledge representation. This section points out the interest of data analysis in supervised learning.

Any conclusion cannot be generalised over a small sample of problems, and by using a wider test bed, learners tend to behave similarly on average. This produces a deadlock situation in the experimental design, which we expect to solve by introducing an a priori data analysis. In the first case, complexity measures can give some explanations to why a learner outperforms another learner based on the data characteristics and their influence on the knowledge representation. For large amounts of data sets, complexity measures may help to select a representative sample of problems with enough resolution and diversity to test learners' performance and, consequently, identify domains of competence. Indeed, if we were able to consolidate a set of benchmarks, this would contribute to perform fair comparisons.

Contribution.

1. Revision and update of the complexity measures proposed by Ho and Basu [2002].
2. Release of DCoL which implements the complexity measures and other functionalities.

4 COMPLEXITY MEASURES MAY EXPLAIN SUPERVISED LEARNING BEHAVIOUR

Summary. This chapter describes the usage of the complexity measures covered previously and deals with the two issues disclosed in previous chapters: (1) data set selection and (2) guidance for learner selection. With no doubt, data hold the key of learnable patterns, and the analysis of their complexity—using realistic approaches—can provide guidelines that should help us to perform some meaningful introspection on the two aforementioned issues.

4.1 INTRODUCTION

Indeed, data set selection has a substantial impact on experimental conclusions. Observed learners' behaviours cannot be generalised over too few data sets and tend to be equal, on average, with too many—as empirically shown in Chapter 2. Our principal motivation of using complexity measures and characterising the difficulty of problems is, therefore, to give an answer to the following questions: How many data sets should be used in the experimentation? Is the applied machine learning technique the most suitable to tackle this type of problem? We believe that data complexity analysis can be useful to (1) investigate the test bed design—number of data sets involved, representativeness of the set—in order to test specific properties of the learners and (2) identify learners' domains of competence.

The purpose of this chapter is to present a different usage of the complexity measures to answer the previous queries and discuss our achievements.

In the following, we recite the main contributions using the set of complexity measures under study and bring the reader's attention to specific details which help to unearth the vicious circle between data, complexity, and learning. We then examine the impact of progressive inclusion of problems with increasing complexity on statistical conclusions and consider the ideal size of a test bed. Moreover, we complete the case study in Chapter 2 by providing explanations of why each learner outperforms the other two in a determined collection of data sets. Finally, we present a meta-learner approach that envisages ultimate goals that can be attained once data complexity is under control.

4.2 COMPLEXITY MEASURES USED FOR...

This section goes through the big picture and cites some relevant works that used the complexity measures for four main purposes: (1) data pre-processing, (2) re-sampling, (3) analysis of learners' behaviours, and (4) meta-learning.

Data pre-processing. Some authors have resorted to the complexity measures to analyse the effect of pre-processing techniques. Dong and Kothari [2003] proposed a feature selection algorithm based on the definition of classifiability. The sub-set selection was based on, given a specific pattern, counting the similarities in the neighbourhood. This approach did not need to construct a classifier and saved the high computational burden of *wrapper* methods, which are impractical when addressing a large number of attributes due to the re-sampling procedure required to estimate the classification error. García et al. [2009] studied the relationship between the Fisher's discriminant ratio (F1) and an evolutionary instance selection method in the classification task. The analysis revealed that F1 can help to decide whether it is adequate to improve the classification capabilities of the k-NN classifier. When F1 is low—indicating strong overlapping—, the best accuracy is reached by applying the prototype selection technique whereas when it is high—indicating low overlapping—, the pre-processing does not guarantee any improvement in the classification accuracy. Some of the drawbacks of this approach are that, even if F1 is covered with good resolution in the interval $[0.035, 2.670]$, the collection of problems is composed of only 24 problems—from the UCI repository—and the computation of this measure for m-class problems is still under study. Furthermore, problems cannot be fully characterised by just one complexity dimension.

Re-sampling. Subsequently, Luengo et al. [2010] applied a similar methodology to predict whether the application of re-sampling techniques would be beneficial in domains with class imbalance. The study used 44 problems—from the UCI repository again—and only F1, N4, and L3. In particular, they obtained two rules to describe *good* and *bad* performance for C4.5 and PART, which help to indicate the most adequate pre-processing method as well. Approximately, F1 $\geqslant$ 1.5 (or F1 $\geqslant$ 0.6), N4 $\leqslant$ 0.2, and L3 $\leqslant$ 0.3 are intervals that indicate a high performance of the techniques and F1 $\leqslant$ 0.3 and N4 $\geqslant$ 0.2, low performance. Basically, this means that for non-overlapped and linearly separable problems, the classifier will be able to correctly learn the concept and classify the instances.

Learners' behaviours. Other authors focused on analysing the behaviour of learners on different problem complexities. Bernadó-Mansilla and Ho [2005] studied the reaction of the eXtended Classifier System (XCS), an influential genetic-based machine learning technique, on real-world problems with different complexities and linked the behaviour of XCS to the problem difficulty estimated by the complexity measures, obtaining what was referred to as the domain of competence of XCS. Two opposite regions of complexity were identified: (1) Difficult problems were characterised by a high number of points laying near the class boundary (N1), a high percentage of adherence sub-sets (T1), a high interleaving in cluster of classes (N2), and high non-linearities (L3 and N4). These values describe problems as interleaved classes and many spread small clusters, which forced XCS to evolve a huge amount of rules to cover these small regions of the space and leave little room for generalisation. (2) Easy problems were characterised by the same complexity measures—N1, N2, N3, L3, and T1—in their low values. In addition, the work indicated in which cases the XCS outperformed a nearest neighbour classifier, a linear classifier, and a decision tree method. The main limitation announced by the authors was, however, the size and representativeness of the test bed. Although 392 problems were involved in the experimentation and their selection was not biased to any of the learners, the sample could not be representative enough and, therefore, limited the scope of the conclusions.

Sánchez et al. [2007] analysed the effect of the data complexity in the nearest neighbour classifier. They attempted to relate class overlapping (F2, N2), feature space dimensionality (F1) and class density (D2, D3) to the practical accuracy of this learner. In their conclusions, they pointed out that k-NN classification performance is strongly sensitive to class overlap and class density, and remarked that N2, D2, and D3—mainly the latter two—are able to predict the k-NN classification accuracy. Nonetheless, despite the small experimentation—nineteen synthetic problems and three real-world from the UCI repository—, these achievements are not so impressive since these measures rely on the same Euclidean distance as the classifier uses.

Luengo and Herrera [2010] examined how a fuzzy system—Fuzzy Hybrid Genetic-Based Machine Learning (FH-GBML)—modelled some classification problems. They discovered that the problems that led to poor results—considering poor results as those whose test accuracy was much lower than the training accuracy—followed a certain pattern in the complexity space. The study only considered F2, F3, L1, L2, N2, N3, N4, T2 over 438 binary classification problems generated from pairwise combinations of 21 data sets from the UCI repository.

Meta-learning. Lastly, Orriols-Puig and Casillas [2010] compared the performance of several fuzzy rule representations within an online learner on the complexity space, and used this characterisation to design a meta-learner that, given a new classification problem, automatically decided which representation should be used to maximise the test accuracy—the decision was based on the problems' apparent complexity reported by the complexity measures. This latter approach is discussed in Sect. 4.4. The conclusions reached in the contribution were that F3, N1, and N2 are not meaningful to guide the selection of the knowledge representation. The automatic set of rules was based on T1, N3, F4, and F2, which mainly estimate the number of clusters present in the data and the discriminative power of the attributes. As a note, the test set consisted of thirty problems from the UCI repository.

Across the review, we observe that complexity measures can provide some hints and explanations about the behaviour of machine learning techniques. We explore this aspect in more detail, as well as initiate a new research line with respect to data set selection analysis.

4.3 DATA SET SELECTION BASED ON COMPLEXITY MEASURES

As reported in Chapter 2, the number of data sets used in experimentation has tended to increase, since researchers aim to demonstrate the perfection of new techniques across a larger variety of domains. Nevertheless, the size of these collections has not been incremented to a feasible extent due to, as warned by the NFL theorem, the risk of not finding significant conclusions if too many domains with different characteristics are considered. In any case, it is not clear how the conclusions change as new data sets are considered in multiple learner comparisons. This section analyses how the collection size affects the conclusions through the prism of data complexity by examining the effect of progressively including problems of bounded difficulty into learner comparisons. In the following, we describe the experimental methodology and then present some results.

4.3.1 *Experimental methodology*

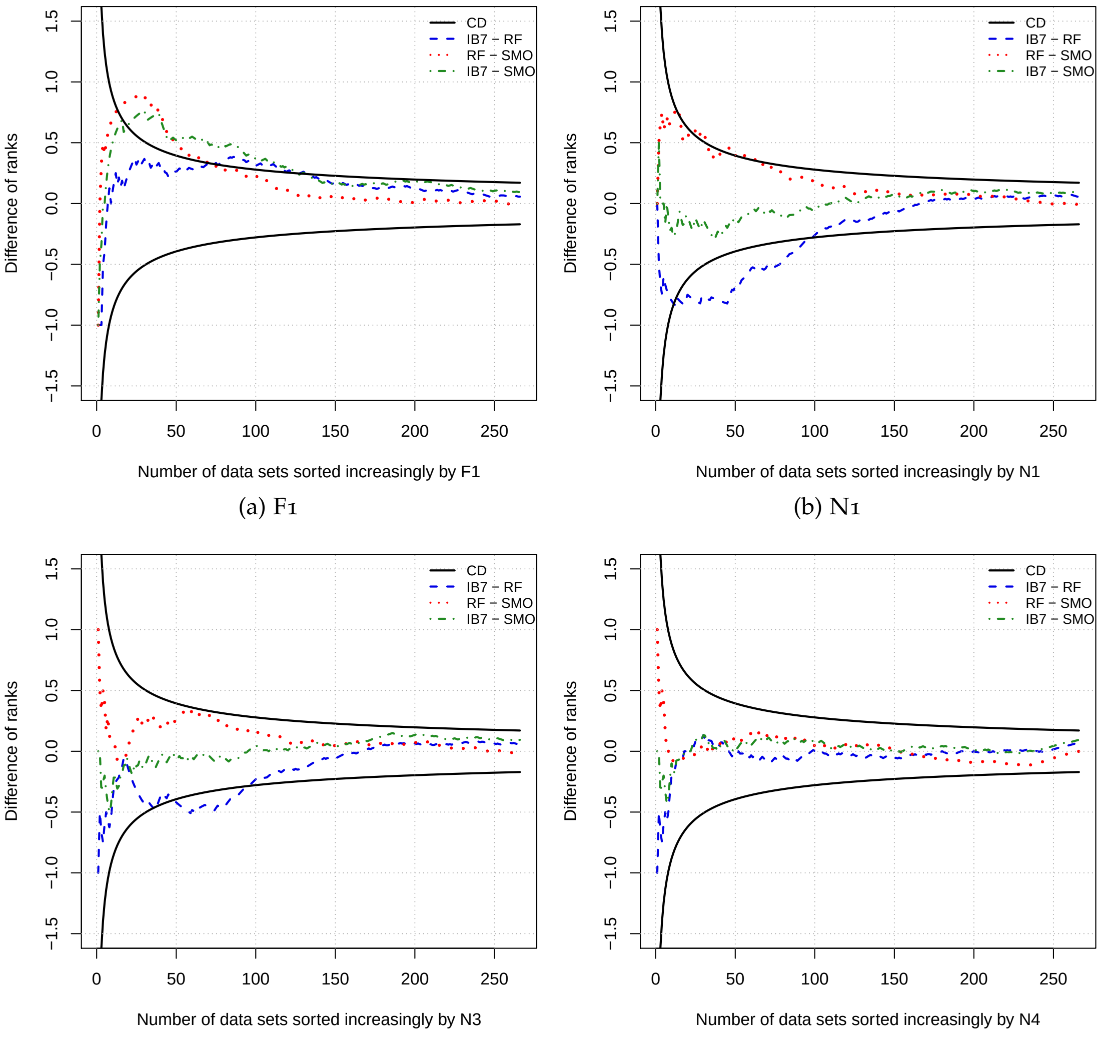
The analysis is based on the initial collection of 308 data sets, used in Chapter 2, which were characterised by the complexity measures proposed by Ho and Basu [2002]. For each complexity measure, we sort the data sets from the easiest to the most complex one according to that particular dimension of complexity. The same process is repeated but sorting the data sets in the reverse way, from the most complex to the easiest one.

For each collection sorted according to a complexity measure, we build 308 collections of data sets, where the first collection contains only the first data set, the second collection contains the first two data sets, and so forth, until the 308th collection contains all the data sets. Then, we evaluate the classification accuracy of the three learners in each collection and apply the Bonferroni-Dunn test to identify which learners significantly degrade the results obtained with the best ranked learner. This analysis is carried out for each complexity measure.

4.3.2 *Increasing complexity and size*

In the first set of experiments, we sort the data sets from the easiest to the most complex one for each complexity measure. For the sake of compactness, Fig. 4.1 gathers the most representative results by showing the experiments of the following complexity measures: (a) F1, (b) N1, (c) N3, and (d) N4. Each plot represents the pairwise difference between the ranks of the learners. The horizontal axis moves along the 308 collections of data sets, built for a particular complexity with increasing level of complexity. The CD, according to the Bonferroni-Dunn test at $\alpha = 0.05$, is depicted with a solid line. When the rank difference falls into the region delimited by the two lines, the learners are statistically equivalent.

The plots show how the differences between learners progressively decrease as new data sets are introduced into the comparison, providing empirical evidence of the NFL theorem. When all the collections of data sets are considered, the rank of all the learners is approximately the same, which means that the accuracy of the three learners is equivalent. On the other hand, note that significant differences are not found for small collections of data sets. In these cases, the critical distance required by the statistical tests is large due to the low number of data sets, and the difference of ranks between learners is small; as a consequence, the statistical analysis cannot detect significant differences. Thence, if significant differences exist, they can be found in collections that contain, approximately, from 20 to 150 data sets.



(a) F1

(b) N1

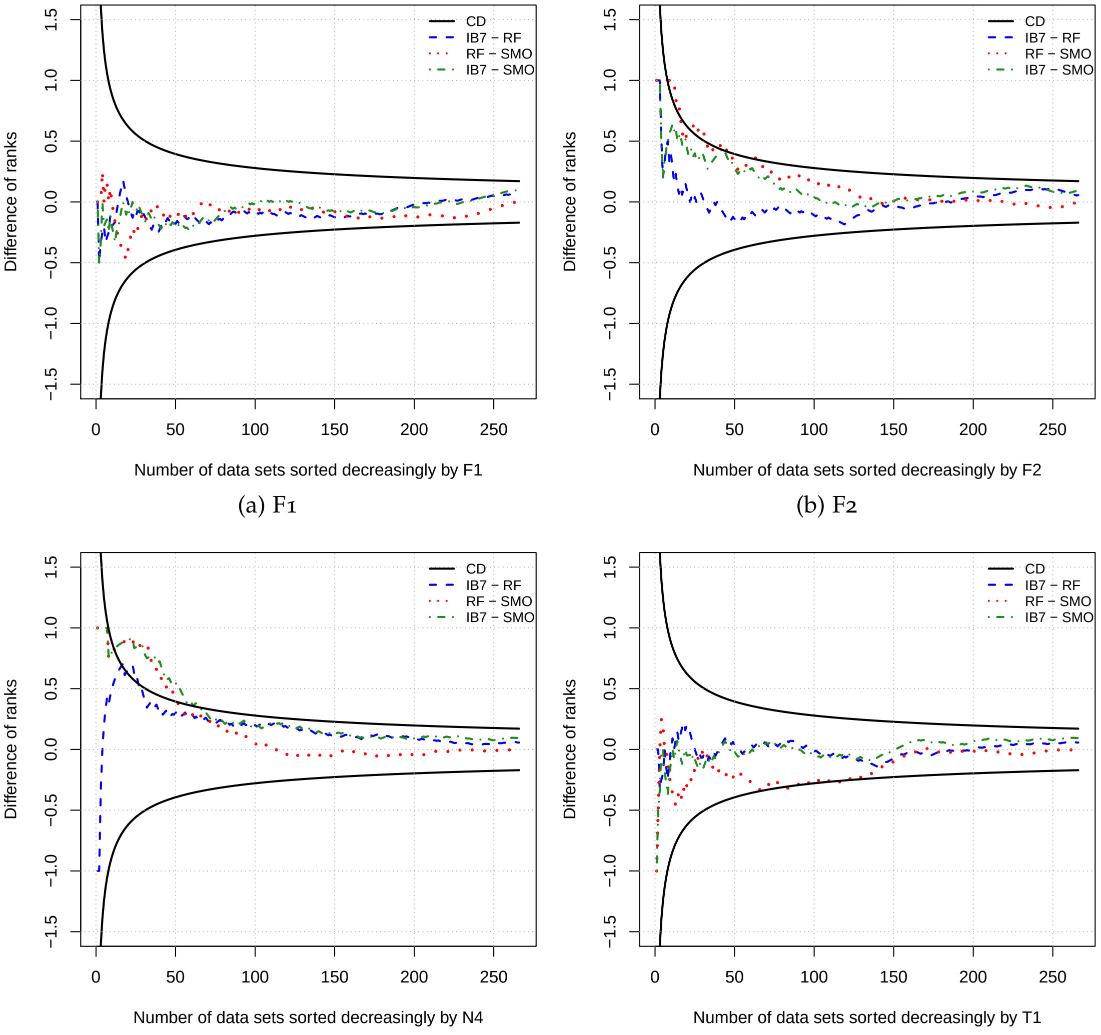
(c) N3

(d) N4

Figure 4.1: Absolute difference between the rank of pairs of learners considering the problems sorted increasingly by each complexity measure.

A more detailed analysis permits the identification of the problem characteristics that pose more difficulties to the different learners. For instance, Fig. 4.1a illustrates that SMO outperforms the other two learners in collections of data sets that have a low F1. This type of problem is complex for RF since the different classes cannot be discriminated by means of partitions that are parallel to an axis of the feature space. The results also indicate that these problems are difficult for IBk. Figures 4.1b and 4.1a point out that the problems with a low N2 and N3 especially benefit IBk. Moreover, RF builds significantly more accurate models than those of SMO in this type of problem. Finally, Fig. 4.1d shows an example in which the inclusion of problems, with progressively increased complexity with respect to N4, does not affect the statistical conclusions.

Fig. 4.2 corresponds to the same study but sorting the problems from the most complex to the easiest one. In this case, we provide the results for (a) F1, (b) F2, (c) N4, and (d) T1. Fig. 4.2a shows no significant difference between pairs of learners for collections of data sets that consist of problems with high values as reported by the measure F1. Note the differences of these results with those reported in Fig. 4.1a. Whereas SMO is the best method when problems with small F1 values are considered, now, when the data sets are sorted in the converse order of complexity, these significant differences disappear. Figures 4.2b and 4.2c illustrate that SMO significantly outperforms the other methods on collections of data sets consisting of problems with large N4



(a) F1

(b) F2

(c) N4

(d) T1

Figure 4.2: Absolute difference between the rank of pairs of learners considering the problems sorted decreasingly by each complexity measure.

and large F2 values, i.e, problems whose boundary is highly non-linear and whose individual features cannot effectively discriminate instances of different classes. Therefore, as also indicated by the previous analysis, SMO appears to be the most robust method for problems with complex or non-linear class boundaries. Finally, Fig. 4.2d shows that, among the three learners, IBk is the least suited to problems with large-moderate T1 values. This means that, as expected, IBk performs poorer than the other techniques on problems where a lot of spheres are required to cover the clusters of instances of different classes.

The analysis conducted emphasises again the importance of studying the data complexity in multiple method comparison and illustrates that, once a collection contains a certain number of data sets, the progressive inclusion of new domains with similar characteristics results in a systematic decrease of the rank differences between learners. Furthermore, we also detect important particular aspects such as that low F1 values benefit SMO or that small N1 values favour IBk. Nonetheless, some problems may have both characteristics, which may benefit both learners. Therefore, the individual analysis of each complexity measure may not be able to fully capture the relationship between combinations of complexity measures and learner ability. Thus, characterisation of problems is, as we have seen in Chapter 3, ongoing research.

4.4 TOO MANY MYSTERIES ABOUT DATA COMPLEXITY

Data complexity can be estimated in different ways. However, all of them allude to the same: working from the details outwards should help us to move from data complexity analysis to the understanding and improvement of supervised learning techniques. This section just reinforces the need for investigating this direction. We analyse the characterisations manually and then define a recursive meta-learning problem.

Why do learners' rankings vary according to the test bed? Characteristics of data sets and their complexity may provide an answer to this question. We believe that data complexity analysis is an important phase in the design of experiments to get reliable conclusions. Indeed, data characterisation should enable us to design experiments and denote for which type of problems—or complexities—a learner—or a group of learners—is the most suitable among the methods involved in the comparison. In order to gain comprehension of which problems are harder for the different learners, the case study in Chapter 2 is taken to the complexity space. For this purpose, we use the DCoL to characterise each domain of the three data set collections. In the following, we describe the experiments and provide some clues of how complexity dimensions explain the learners' behaviours.

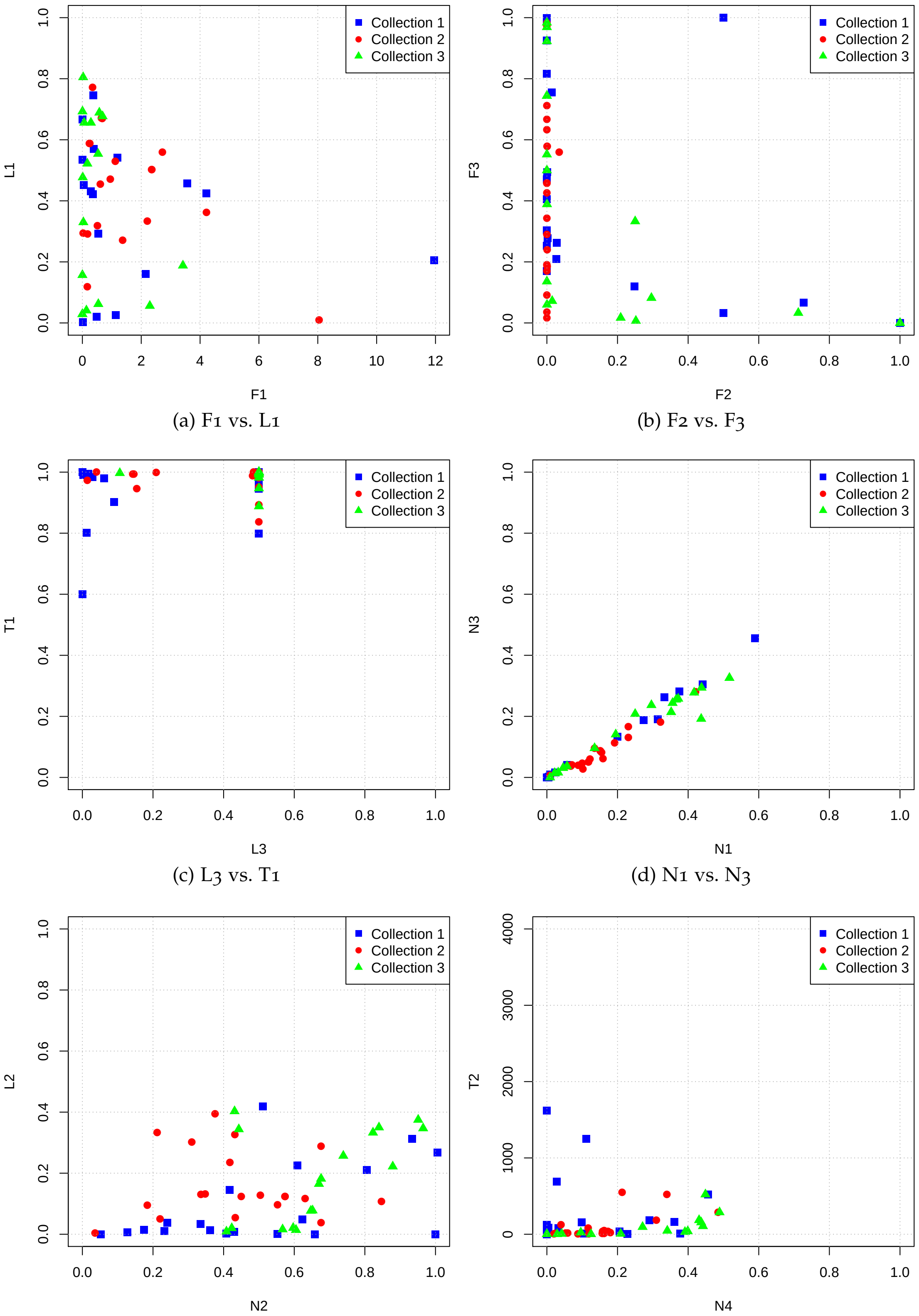
Fig. 4.3 plots all the 60 data sets projected on the complexity space. More specifically, each scatter plot is a projection of data sets on two complexity measures. The data sets that belong to each one of the three collections are depicted in different symbols and colours: (1) blue squares for the data sets within the first collection, where IBk is the best learner, (2) red points for the data sets within the second collection, where RF is the best learner, and (3) green triangles for the data sets within the third collection, where SMO is the best learner. From all the plots, we observe that only two projections separate the three collections in the complexity space: Fig. 4.3d, which projects the data sets on N1 and N3, and Fig. 4.3e, which projects the data sets on N2 and L2. The others do not provide such clear separations; therefore, the following discusses how the three learners perform through the complexity space formed by N1, N2, N3, and L2.

The most relevant information is supplied by the projection on N1, which estimates the length of the class boundary, and N3, which evaluates how close the examples of different classes are by giving the leave-one-out error rate of the 1NN classifier (see Fig. 4.3d). In this complexity space, the problems are grouped into two almost-disjoint regions that lead us to the following observations:

1. The data sets from the second collection have higher values of N1 and N3. So, there is a greater interleaving among instances of different classes, and instances of the same class are more disperse. Thus, other approaches that are able to approximate complex class boundaries, sparsity, and noise may be a better option for these type of problems. Actually, this is the conclusion in the case study regarding the second collection, where RF achieves the most accurate classification models.

2. The data sets from the third collection present the highest values of N1 and N3. In most of these problems, more than half of the instances lay on the class boundary, drawing a dense class boundary and increasing the difficulty of the problem. The experimental analysis demonstrates that SMO significantly outranks the other two methods. For these type of problems, the high interleaving between instances of different classes seems to prevent decision trees from identifying the decision boundaries accurately. Yet, the flexibility provided by the polynomial kernel used enables SMO to accurately approximate complex class boundaries.

The projection on N2, which measures the interconnectedness of examples of the same class, and L2, which evaluates to what extent the training data is linearly separable, also supplies valuable information about the problem characteristics that affect each learner. IBk outperforms the other learners in the first collection, which is composed of data sets that are linearly—or almost linearly—separable (that is, a low value of L2). RF is the best performer in the second collection, which contains problems with a moderate value of N2 and a medium value of



(a) F1 vs. L1

(b) F2 vs. F3

(c) L3 vs. T1

(d) N1 vs. N3

(e) N2 vs. L2

(f) N4 vs. T2

Figure 4.3: Projection of the problems where IBk (blue squares), RF (red points), and SMO (green triangles) are the best performers, respectively, on different complexity spaces.

Table 4.1: Training and test performance—percentage of correctly classified instances—of the domains of competence of IBk, RF, and SMO.

	IBk	**RF**	**SMO**
Training	89.00%	85.22%	84.54%
Test	81.79%	79.04%	78.60%

```
L2 ⩽ 0.041: 0 (141.0/10.0)
L2 > 0.041
|   N2 ⩽ 0.485
|   |   F2 ⩽ 0.000048
|   |   |   N3 ⩽ 0.002: 0 (2.0)
|   |   |   N3 > 0.002: 1 (35.0/5.0)
|   |   F2 > 0.000048
|   |   |   N3 ⩽ 0.166: 0 (12.0/2.0)
|   |   |   N3 > 0.166: 1 (4.0/1.0)
|   N2 > 0.485: 0 (97.0/25.0)
```

Figure 4.4: Tree built by C4.5 for the problem of learning the domain of competence of RF.

L2. Finally, SMO is the best method in the third collection, which contains problems with moderate-high values of both N2 and L2.

4.5 META-LEARNING PROBLEM

The analysis in the complexity space provided herein shows the close relation between the learner behaviour and the complexities of the selected data sets—something already observed in the literature. This leads to the ultimate goal of providing a set of rules to select the appropriate learning techniques. After having studied each complexity dimension manually, we consider all the dimensions to define the domain of competence of each learner with respect to the other techniques, i.e. the *sweet spot* where a particular learner outperforms its rivals. To achieve this, we transform the typical manual inspection of results into a problem of automatic learning and automatically identifying the domain of each learner, as also suggested in Orriols-Puig and Casillas [2010]. This section elaborates upon this idea to learn such domains for the three learners used in the case study.

Let us define a classification problem $\mathcal{D}_i$ for each learner ℓ_i that describes the domain of excellence of learner ℓ_i. Each instance of the problem $\mathcal{D}_i$ corresponds to one of the domains, which is characterised by the complexity measures. The label of each instance is 1 if the given learner ℓ_i obtained the best performance and 0 otherwise. With this definition, we can use any machine learning technique to model the domain of excellence of each learner with respect to the others. Note that this approach seems to be useful not only to analyse under which problem characteristics each technique excels, but also to predict the best method for a new unknown problem described by its apparent complexity.

We perform the experiments on the collection of 308 data sets and use C4.5 [Quinlan, 1995] to learn the domain of competence of the three learners included in the comparison, i.e. IBk, RF, and SMO. We adopt C4.5 because of its tree representation, which can easily be interpreted by human experts. As we also aim to analyse the predictive capabilities of the resulting models, we apply the ten-fold cross-validation procedure [Dietterich, 1998] to estimate the accuracy.

Table 4.1 summarises the training and test accuracies obtained for the models that predict (1) when IBk is the best suited learner (first row), (2) when RF is the best suited learner (second row), and (3) when SMO is the best suited learner (third row). The results show that all the models have a test accuracy above 78%.

```
F1v ⩽ 0.418
|   N4 ⩽ 0.38
|   |   F2 ⩽ 0.000012
|   |   |   F1v ⩽ 0.049: 0 (8.0)
|   |   |   F1v > 0.049
|   |   |   |   F3 ⩽ 0.16: 1 (2.0)
|   |   |   |   F3 > 0.16: 0 (3.0/1.0)
|   |   F2 > 0.000012: 1 (45.0/13.0)
|   N4 > 0.38: 0 (13.0)
F1v > 0.418
|   T2 ⩽ 8.2: 0 (71.0/3.0)
|   T2 > 8.2
|   |   T2 ⩽ 185.5
|   |   |   T2 ⩽ 125
|   |   |   |   N2 ⩽ 0.718: 0 (76.0/11.0)
|   |   |   |   N2 > 0.718
|   |   |   |   |   T2 ⩽ 30.2: 1 (7.0)
|   |   |   |   |   T2 > 30.2: 0 (3.0)
|   |   |   T2 > 125: 1 (11.0/4.0)
|   |   T2 > 185.5
|   |   |   T1 ⩽ 0.801
|   |   |   |   F4 ⩽ 0.543: 1 (2.0)
|   |   |   |   F4 > 0.543: 0 (8.0)
|   |   |   T1 > 0.801: 0 (42.0)
```

Figure 4.5: Tree built by C4.5 for the problem of learning the domain of competence of IBk.

This indicates that, although the complexity measures used cannot fully represent all the sources of problem complexity, they supply enough information to build models that predict the best suited learner to a new problem with moderate-high reliability. Fig. 4.4 shows a piece of the tree built by C4.5 that explains the domain of excellence of RF. In this case, the model indicates that RF may be the best performer for moderate values of N2, N3, and L2, which was already detected in our previous analysis. Fig. 4.5 shows the tree built to explain the domain of excellence of IBk. We observe that the number of rules substantially increases and the model provides a more refined results than the manual analysis. However, this model contains some of the patterns previously observed. For instance, IBk is not suited for large values of T1.

We can see that the meta-learner problem is unbalanced, and the minority class corresponds to the dominance of the learners under study. This is due to the fact that the accuracies of the three learners range in the same, short interval (see Fig. 4.6). The little variation in the performance highlights the need for a collection of problems that offer more resolution in terms of accuracy and data complexity.

4.6 data characterisation and diversity: meta-learner weaknesses

The meta-learning approach is not a new idea; projects STATLOG [Michie et al., 1994] and MetaL[1] were pioneers in this aim. The latter was an attempt to develop an assistant system that guided users through the space of experiments either providing support with the learner selection or data transformation. This was built upon experience, upon the past usage of machine learning systems. The project ended in 2002; nearly ten years later no active research is visible from it. Interestingly, the success (or failure) of such initiatives depends on the description of the problem, i.e. the characterisation of the complexity.

In the project MetaL, the characterisation was grounded on simple measures—e.g. the number of attributes, the number of classes, etc.—, statistical measures—e.g. mean and variance of numerical attributes—, and information theory-based measures—e.g. entropy of classes and

[1] http://www.metal-kdd.org/

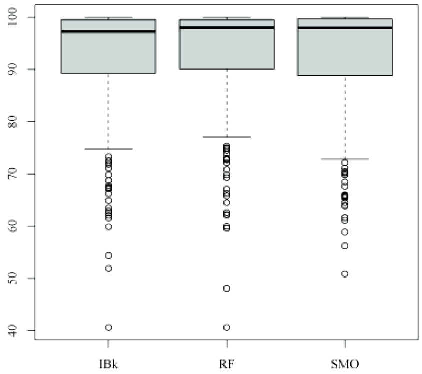


Figure 4.6: Accuracy of IBk, RF, and SMO over the whole collection of problems.

attributes as in [Brazdil et al., 1994]. Later, Peng et al. [2002] proposed fifteen characterisation measures based on the description of the decision tree that represented the data. In our approach, the novelty is the use of intrinsic complexity estimates to characterise the problem. But, the most important thing is to find the way to condense all this information in a succinct set of measures. Controversially, we attain a situation where the understanding of learners' performance is intrinsically bound to the characteristics of the data sets which are still under study. In both cases, learning and characterisation, the experimental research is limited by the availability of a representative sample of problems, as seen in the next chapter. Actually, there is a heavy dependence between these three elements—learners, complexity estimates, and data sets—which demands to work on controlled environments first to grasp solid knowledge, and then expand this knowledge to real-world problems in order to get real insights about the suitability of particular algorithms, particular structures, particular learning biases, etc. Therefore, in the following we study artificial data sets and their generation.

Contribution.

1. Proposal of the test bed size required in the experiments.
2. Justification of learners' behaviours based on complexity analysis.
3. Construction of a meta-learning problem that automatically discovers domains of competence of learners based on complexity estimates.