# UNIVERSITY OF LONDON THESIS

Degree PhD Year 2007 Name of Author TROTTER Matthew William Burnell.

# Support Vector Machines for Drug Discovery

Matthew William Burnell Trotter

Department of Computer Science
University College London

A thesis submitted to the University of London in the
Faculty of Science for the degree of Doctor of Philosophy





ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

I, Matthew William Burnell Trotter, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Support vector machines (SVMs) have displayed good predictive accuracy on a wide range of classification tasks and are inherently adaptable to complex problem domains. Structure-property correlation (SPC) analysis is a vital part of the contemporary drug discovery process, in which several components of the search for novel molecular compounds with therapeutic potential may be performed by computer (*in silico*). Inferred relationships between molecular structure and biological properties of interest are used to eliminate compounds unsuitable for further development. In order to improve process efficiency without rejecting useful compounds, predictive accuracy of such relationships must remain high despite a paucity of data from which to infer them.

This thesis describes the application of SVMs to SPC analysis and investigates methods with which to enhance performance and facilitate integration of the technique into present practice. Overviews of contemporary drug discovery and the role of machine learning place the investigation into context. Computational discrimination between compounds according to their structures and properties of interest is described in detail, as is the SVM algorithm. A framework for the assessment of supervised machine learning performance on SPC data is proposed and employed to assess SVM performance alongside state-of-the-art techniques for *in silico* SPC analysis on data provided by GlaxoSmithKline.

SVM performance is competitive and the comparison prompts adaptations of both data treatment and algorithmic application to explore the effects of data paucity, class imbalance and outlying data. Subsequent work weights the SVM kernel matrix to recognise heavily populated regions of training data and suggests the incorporation of domain-specific clustering methods to assist the standard SVM algorithm. The notion that SVM kernel functions may incorporate existing domain-specific methods leads to kernel functions that employ existing pharmaceutical similarity measures to treat an abstract, binary representation of molecular structure that is not used widely for SPC analysis.

I would like to thank my supervisors, Sean Holden and Bernard Buxton, for their advice, encouragement and support; fellow members of the INTErSECT Faraday Partnership's RoCKET project - in particular, Robert Burbidge, David Corney and Bill Langdon (UCL), Darko Butina, Anne Hersey, Chris Luscombe and Stephen Barrett (GSK) and Tom Khabaza (SPSS), all of whom made significant contributions to my research and development; colleagues at UCL, who offered help and distraction in equal measure; my parents, relations and friends, for maintaining contact when I did not; and my wife, Anna, and children, Tabitha and Emmanuelle, for their love, patience and understanding.

# Contents

**Abstract** 3

**Acknowledgements** 4

**Contents** 5

**List of Figures** 7

**List of Tables** 8

**1 Introduction** 9

1.1 Motivation & Hypotheses . . . . . 9
1.2 Application . . . . . 10
1.3 Thesis Structure . . . . . 11
1.4 Conclusion . . . . . 12

**2 Background** 14

2.1 Contemporary Drug Discovery . . . . . 14
2.1.1 Identification of a Therapeutic Target . . . . . 14
2.1.2 High Throughput Screening . . . . . 15
2.1.3 Lead Generation & Combinatorial Chemistry . . . . . 17
2.1.4 Structure-Property Correlation Analysis . . . . . 18
2.1.5 *In silico* Molecular Representation . . . . . 22
2.1.6 Lead Optimisation . . . . . 25
2.1.7 Performance Evaluation of SPC Classifiers . . . . . 26
2.1.8 Research Focus and Conclusion . . . . . 28
2.2 Contemporary Machine Learning for Drug Discovery . . . . . 29
2.2.1 Introduction . . . . . 29
2.2.2 Linear Classification . . . . . 31
2.2.3 Non-Linear Classification . . . . . 35
2.2.4 Impediments to Generalisation . . . . . 38
2.2.5 Support Vector Machines . . . . . 44
2.2.6 Machine Learning in Drug Discovery . . . . . 53
2.2.7 Conclusion . . . . . 68

**3 ADMET Data & Experimental Practice 70**
3.1 GlaxoSmithKline Data . . . . . 70
3.1.1 Blood-Brain Barrier . . . . . 70
3.1.2 P-glycoprotein Substrate Binding . . . . . 71
3.1.3 Acute Toxicity . . . . . 72
3.1.4 Bioavailability . . . . . 72
3.1.5 Protein Binding . . . . . 73
3.2 Experimental Practice . . . . . 74
3.2.1 Data Partitioning . . . . . 75
3.2.2 Accuracy Measures for Imbalanced Class Populations . . . . . 76
3.2.3 Parameter Selection . . . . . 77
3.2.4 Performance Comparison . . . . . 78
3.3 Model Building . . . . . 79
3.3.1 SVM Parameters . . . . . 80
3.3.2 ANN & RBF Parameters . . . . . 80
3.3.3 C5.0 Parameters . . . . . 81
3.3.4 Nearest-Neighbour Parameters . . . . . 81
3.4 Conclusion . . . . . 81

**4 Support Vector Machines for ADMET Property Classification 82**
4.1 Machine Learning Comparison - SVM vs. State-of-the-Art . . . . . 82
4.1.1 Results . . . . . 83
4.1.2 Discussion . . . . . 87
4.2 Balancing Generalisation Performance . . . . . 94
4.2.1 Results . . . . . 97
4.2.2 Discussion . . . . . 97

**5 Neighbourhood Influence on Support Vector Machine Classification 107**
5.1 Neighbourhood Weighting the SVM Kernel Matrix . . . . . 108
5.2 Results . . . . . 110
5.3 Discussion . . . . . 112

**6 Tanimoto Kernels for Support Vector Machine Classification 118**
6.1 Tanimoto Similarity Kernels for Binary Data . . . . . 125
6.2 Results . . . . . 127
6.3 Discussion . . . . . 130

**7 Conclusion 137**
7.1 Summary & Contributions . . . . . 137
7.2 Suggested Future Work . . . . . 139
7.3 Closing Statement . . . . . 141

**Bibliography 142**

# List of Figures


2.1 The Primal Perceptron Algorithm . . . . . 34
2.2 The Dual Perceptron algorithm . . . . . 36
2.3 The Optimum Separating Hyperplane . . . . . 46
2.4 Slack Variables . . . . . 49

3.1 MDS Plot of BBB Data . . . . . 71
3.2 MDS Plot of P-gp Data . . . . . 72
3.3 MDS Plot of Acute Toxicity Data . . . . . 73
3.4 MDS Plot of Bioavailability Data . . . . . 74
3.5 MDS Plot of Protein Binding Data . . . . . 75

4.1 MDS Plots of BBB Training (left) and Test (right) Partitions . . . . . 83
4.2 MDS Plots of P-gp Training (left) and Test (right) Partitions . . . . . 84
4.3 MDS Plots of Acute Toxicity Training (left) and Test (right) Partitions . . . 84
4.4 MDS Plots of Bioavailability Training (left) and Test (right) Partitions . . . 84
4.5 MDS Plots of Protein Binding Training (left) and Test (right) Partitions . . 85
4.6 Original (left) and Mahal-Reduced (right) BBB Training Data . . . . . 97
4.7 Original (left) and Mahal-Reduced (right) P-gp Training Data . . . . . 100
4.8 Original (left) and Mahal-Reduced (right) Acute Toxicity Training Data . . 100
4.9 Original (left) and Mahal-Reduced (right) Bioavailability Training Data . . 100
4.10 Original (left) and Mahal-Reduced (right) Protein Binding Training Data . . 101

6.1 Output of Tanimoto and Tanimoto-RBF Kernel Functions . . . . . 127
6.2 Estimated Distribution of Bit Content in the GSK Daylight Data Classes . . 129

# List of Tables


4.1 GSK Data: Class Distribution in Training and Test Partitions . . . . . 83
4.2 Algorithmic Performance on GSK Test Data . . . . . 86
4.3 Data Dimension Pre- and Post- PCA Transformation . . . . . 91
4.4 Algorithmic Performance on PCA-Reduced GSK Test Data . . . . . 92
4.5 Weighted Algorithm Performance on GSK Data Test Data . . . . . 98
4.6 Mahalanobis-Reduction: Algorithm Performance on GSK Test Data . . . . 99
4.7 K&S-Reduction: SVM Performance on GSK Test Data . . . . . 104

5.1 $k$NN-SVM vs. SVM Performance on GSK Test Data . . . . . 111
5.2 BC-SVM vs. SVM Performance on BBB Test Data . . . . . 115
5.3 SVM and BC-SVM Performance on Mahalanobis-Reduced BBB Data . . . 115

6.1 Range and Selected Percentiles of Bit Content in GSK Daylight Data Sets. . 128
6.2 Selected Percentiles of Bit Content in GSK Daylight Data Classes. . . . . 128
6.3 SVM Kernel Performance on GSK Daylight Test Data . . . . . 131
6.4 $k$-NN Performance on GSK Daylight Test Data . . . . . 132
6.5 Combined RBF and Tanimoto Kernels on Abraham and Daylight P-gp Data 133
6.6 Tanimoto Kernel Performance on K&S Reduced GSK Daylight Data . . . . 134

# Chapter 1

# Introduction

This thesis concerns the application of supervised machine learning to the analysis of data drawn from the contemporary drug discovery process. Research, in collaboration with the pharmaceutical company GlaxoSmithKline (GSK), demonstrates that *support vector machines* (SVMs) are a suitable technique with which to build classifiers capable of distinguishing discrete classes of biological behaviour according to the structures of novel pharmaceutical compounds. Further investigation suggests improvements, both to the technique and to the practice of its application to drug discovery, that have the potential to increase performance.

## 1.1 Motivation & Hypotheses

Several aspects should be considered upon the introduction of a machine learning technique to a new application. Primary questions asked of both technique and application include whether the application would benefit from the inclusion of another technique in its analysis, whether the technique involved is capable of analysing the application successfully and how success is measured in the context of the application. Secondary questions, such as whether the technique may be adapted to treat the application better or whether the application may be altered in order to improve machine learning performance upon it, should be asked also.

A major motivating factor in the introduction of a machine learning technique to a new area of application is that it provides useful information regarding the behaviour of the technique in practice. However, the majority of real-world applications seldom provide data that describes perfectly the relationship between process and outcome. Small samples of known data are employed to analyse relationships that govern large amounts of unknown data. Accordingly, any deviation in the known data from the wider relationship across all data causes significant problems to machine learning when the aim is to create a relationship that generalises well to unknown data. This unwelcome facet of real-world applications provides machine learning research with contradicting goals. It is important that a new technique is well founded and shown to work on statistically regular data, but if that technique is to be applied to a real-world domain, with its associated difficulties, it must be able to cope with

at least some disruption to the relationship it is employed to learn. From a machine learning perspective, pharmaceutical classification is particularly challenging. There is no universal standard for the computational representation of molecular structure. The data itself is subject to bias during its extraction from the industrial process, contains significant amounts of noise and exhibits complex non-linear relationships between structural attributes and class labels and between the structural attributes themselves.

A sensible approach to such challenges is to take a well-founded technique and adapt it to cope with real-world applications in a manner that affects its analytical strengths as little as possible. The application of machine learning techniques to real-world scenarios, such as drug discovery, provides a test bed for their further development. Therefore, the hypotheses investigated during this work take the form of positive answers to the research questions posed above. In particular that:

1a. the application *does* benefit from the inclusion of another technique in its analysis;

1b. the technique *is* capable of analysing the application successfully; and

2a. the technique *can* be adapted to treat the application better;

2b. the application *can* be altered to improve machine learning performance upon it.

Measurement of successful analysis from the context of the application is answered by necessity during investigation of the two primary hypotheses above. The work performed tests the hypotheses against null hypotheses presented by negative answers to the research questions.

## 1.2 Application

The roles played by pharmaceutical classification within the contemporary drug discovery process are described in Chapter 2. Pharmaceutical classification has risen in importance as the search for new therapeutic products has widened. It is no longer sufficient to examine the range of pharmaceutical compounds known to a pharmaceutical company and develop a new product from within that collection [Beresford et al., 2002]. Novel therapeutic products must compete in an increasingly crowded market place and it is increasingly difficult to discover a novel product using a finite collection of known compounds as the basis for development.

Advances in the contemporary drug discovery process require the replacement of biological compound selection with computational models of biological selection in order to identify novel compounds with potential for development into a therapeutic product. Classifiers that relate molecular compound structure to biological properties of interest (known as *structure-property correlation* (SPC) analysis) are becoming widely used for computational analysis of biological processes and relationships. Support vector machines have hitherto demonstrated greater predictive ability than a number of other supervised machine learning

methods on a wide range of real world applications (see, for example, [Jaakkola et al., 1999; Ward et al., 2003; Hammond et al., 2004]). More importantly, SVMs have demonstrated an adaptability to specialised domains that is lacking in many other such methods. It is expected that SVMs in standard form will provide predictive ability that is competitive with machine learning methods currently used for pharmaceutical classification. Moreover, it is anticipated that SVMs may be adapted to the domain to provide better results than those provided by the standard form of the algorithm.

The approach taken in this thesis towards development of the technique and its application to the domain considers more than the straightforward pursuit of performance increase. An endeavour is made to provide developments via the incorporation of extant methods of pharmaceutical analysis, including relevant methods of similarity assessment and pattern identification, in order to facilitate their eventual incorporation into the drug discovery process.

## 1.3 Thesis Structure

This work provides a focused example of the use of a state-of-the-art machine learning technique in a specific area of the drug discovery process. The expected benefit of this is that the application will profit from the use of another recent technique in its analysis and the applicative practice reported will aid the introduction of future techniques.

Chapter 2, *Background*, provides an overview of the area of application (pharmaceutical classification) and the use of machine learning within it. Section 2.1 guides the reader through the contemporary drug discovery process towards the focus of this research. Section 2.2 introduces machine learning, before discussing its role within contemporary drug discovery. Advantages of the use of machine learning for this problem are described, along with obstacles in the path of successful analysis. Measures available with which to overcome such obstacles are discussed and a perspective from which to measure algorithmic performance when analysing this application is described. The support vector machine algorithm is described alongside recent developments to the technique and their use hitherto for drug discovery and other real-world applications. The background material culminates by matching the strength of support vector machines to the challenges posed by a specific area of the contemporary drug discovery process (SPC analysis).

Chapter 3, *ADMET Data and Experimental Practice*, introduces the real-world data employed to demonstrate the major contributions of this work. Five SPC analysis problems, provided by GSK, are visualised and described in terms of their purpose and the nature of the data involved. Subsequently, an experimental rationale and practice for the principled comparison of several machine learning techniques on pharmaceutical data are outlined. Chapters 4, 5 & 6 employ the practice of Chapter 3 in experimental work undertaken to test the research hypotheses of section 1.1 above.

Chapter 4, *Support Vector Machines for ADMET Property Classification*, is formed by two distinct pieces of work, the first of which approaches the primary research hypotheses.

The performance of several machine learning techniques is compared when they are used to analyse the SPC data provided by GSK. Results of the comparison are discussed and guidelines for the use of support vector machines, and supervised machine learning in general, for small scale pharmaceutical modelling are reported throughout. The results are assessed as to whether they provide positive answers to the primary research questions.

The second section in Chapter 4 considers measures with which to acknowledge imbalanced training data class sizes, a particular challenge posed to classifier creation by many small SPC problems. The effect of existing methods that direct machine learning algorithms towards the provision of balanced classification accuracy during training is employed as a benchmark, against which to assess the use of existing pharmaceutical analysis procedures to alter the training data in order to achieve similar effect. Discussion of the results relates the data balancing measures introduced to other work in the field and suggests several approaches via which the methods may be developed further.

Chapters 5 and 6 investigate whether the SVM algorithm may be adapted to better analyse the application. Two developments of the SVM algorithm are introduced. Chapter 5, *Neighbourhood Influence on Support Vector Machine Classification*, describes the effect of biasing the existing SVM data transformation according to aspects of locality within the training data. Chapter 6, *Tanimoto Kernels for Support Vector Classification*, describes conversion of an existing structural similarity measure in order to allow domain-relevant SVM application to a specific pharmaceutical data representation, Daylight fingerprints, that is not used widely for SPC analysis in the later stages of drug discovery.

Chapter 7 concludes the work by considering the outcome of its tests upon the research hypotheses. The work described here gives rise to several lines of further investigation that provide interesting platforms from which to continue research in this area. Suggestions are made for improvement of the present work and ideas for future work and new directions in SPC analysis are discussed.

## 1.4 Conclusion

This thesis concerns the application of supervised machine learning to the analysis of data drawn from the contemporary drug discovery process. Research, in collaboration with the pharmaceutical company GlaxoSmithKline, provides the following contributions:

- an experimental framework for the comparison of supervised machine learning algorithms when applied to pharmaceutical data;
- a detailed assessment of the suitability of the support vector machine method for supervised learning from small sets of pharmaceutical data;
- adaptations to the standard formulation of SPC analysis as a supervised machine learning problem that balance the training data to improve algorithmic performance; and

- adaptations to the SVM kernel transformation that may improve performance by incorporating domain-relevance into the treatment of pharmaceutical analysis problems.

The intention of the work, therefore, is to introduce and adapt a relatively new tool to a well-defined area of drug discovery to the prospective benefit of both machine learning and drug discovery communities. The tool, support vector machines, benefits from exposure to a new area of application and the non-standard challenges that come with it. The area of application benefits from a set of guidelines regarding the application of a state-of-the-art technique, which may assist in what is commonly a 'trial and error' introduction of a recent development to an industrial process. In addition, the attempt is made during this work to adapt the technique in a manner that will facilitate its insertion into extant drug discovery processes. Tools and procedures, e.g. similarity measures, data representations and data description techniques, are employed where possible in order to make the developments proposed by this work easier both to interpret and to incorporate within a contemporary drug discovery environment.

This thesis is submitted for examination in computer science and, thus, the primary focus is on the computational technique applied. Nevertheless, it is hoped that the work will be of use and interest to the drug design community in general and to those wishing to classify pharmaceutical data with support vector machines in particular.

# Chapter 2

# Background

## 2.1 Contemporary Drug Discovery

In order to understand the classification task to which support vector machines are applied during the course of this work, it is first necessary to describe the contemporary drug discovery process and the rationale behind it. This section describes major stages of the contemporary drug discovery process.

### 2.1.1 Identification of a Therapeutic Target

The search for a new pharmaceutical product begins with the identification of a therapeutic target, e.g. a protein implicated in some pathogenic process. The first aim of drug discovery is to find a novel compound that reacts with (is biologically active against) the target in the desired manner. For example, an inhibiting compound will bind to the target in a manner that impedes its undesirable effects, e.g. via its unimpeded interaction with receptors on the cell surface [Wang et al., 2004]. The majority of therapeutic products interact with their target on the cellular or molecular level.

The success of the search depends largely upon the amount of information available regarding the target. Information may be acquired from empirical investigation of the target, e.g. examination of protein binding sites and local structure, known crystal structure (if available), or from the large amounts of new therapeutic information arising from the fields of functional genomics and proteomics [Debouck and Metcalf, 2000]. Once the target is described in sufficient detail, the search for compounds that are active against it may begin.

An ability to interact with the target in the manner desired is only the first step towards discovering a novel therapeutic product. Most compounds selected by an initial search for binding affinity to the target, for example, will possess very few of the properties required for eventual sale as a therapeutic product. It is more likely that compounds that are identified as 'suitable' during the early stages of the drug discovery process will become *backbone* compounds that are used as a development platform from which to produce a novel product with properties optimised for the therapeutic aim.

It is important to introduce the notion of *chemical space* [Walters and Murcko, 2002]

as early as possible, as it provides a useful tool with which to describe both drug discovery and the use of machine learning in this context Chemical space is that space inhabited by all possible molecular combinations. Conservative estimates place the number of viable therapeutic compounds[1] in this space to be $10^{60}$. The number of novel compounds from which to select the basis of a therapeutic product is very large. Rough estimates, based on current search throughput capabilities, place the length of time required to assess the suitability of all possible novel therapeutic compounds against a single target to be in the order of the lifetime of the universe. Estimates such as this must be regarded with caution, but the reality is that a practical search for novel compounds of therapeutic worth presents a significant problem to the investigator. An important aid to drug discovery is the prior knowledge that compounds with similar properties, both physical and therapeutic, tend to inhabit similar regions of chemical space [Van Hijfte et al., 1999]. For the purposes of the search, it is valid to say that the space is localised.

### 2.1.2 High Throughput Screening

Since the 1980s, the search through chemical space for novel compounds has been formulated as a 'needle in a haystack' search by elimination [Beresford et al., 2002; Xu and Hagler, 2002]. A key component of the early search process is high throughput screening (HTS).

HTS comprises a series of rapid, batch assays that test the affinity of a large number of compounds against a well-described therapeutic target. Pharmaceutical companies keep large collections of compounds (in the order of $10^6$ - $10^7$) with known molecular structures and binding properties [Xu and Hagler, 2002]. The collections form a record of their work to date and provide a known region of chemical space from which to start a search. When confronted with a new therapeutic target, it is common to assess the entire corporate collection against the target using HTS to identify those compounds that are active against it. The *hits* that emerge successful from HTS provide backbone compounds that can be used to explore useful regions of chemical space.

At this point, it is useful to distinguish *in vitro* and *in silico* methods. Increases in automated production line processes allow thousands of compounds to be synthesised and assayed against a biological target in batch. Industry sources regularly pronounce HTS throughputs in excess of $10^6$ compounds per day, although a realistic estimate across the pharmaceutical industry is closer to $10^5$ compounds per day [Dixon et al., 2000]. When this figure is compared to the above estimate of the number of potential compounds that inhabit useful chemical space ($10^{60}$) it is clear that initiating a search from a relatively small subset of known compounds does not permit a full exploration of chemical space. On occasion, the search is augmented by the inclusion of all compounds that a company is capable of synthesising at the time. Contrary to this, if prior domain knowledge suggests that a small, well-defined area of chemical space is likely to produce compounds that are active against

[1] Viable compounds are commonly specified as those compounds having molecular weight $< 500$ - beyond which metabolism and ingestion become difficult.

the target, a subset of the corporate collection that describes that area may be assayed.

The application of HTS methods to existing compounds, or those synthesised for the purposes of the search, is known as *in vitro* HTS. The term *in vitro* can be roughly translated from the Latin of its origin as 'in glass', thus signifying the creation and evaluation of the compounds physically in laboratory conditions. The *in vitro* HTS processes of today allow a much wider initial search than the 'trial and error' and prior knowledge based methods in place before the inception of HTS [Van Hijfte et al., 1999; van de Waterbeemd, 2003]. There exist drawbacks to *in vitro* HTS, however, which have necessitated a further development, similar in magnitude to the development of *in vitro* HTS when compared to the methods used before it, in the initial search for active compounds.

In order to lead smoothly to the most recent development in high throughput screening, first it is necessary to list the deficiencies of the *in vitro* method.

- to search chemical space using a subset of $10^7$ known compounds limits the diversity of novel compounds that may be discovered when using the results of the search as the basis for future development;
- when all compounds that a company is capable of synthesising are included in the search, an increase in diversity may be noted but the diversity is limited by the range of chemical reagents available at the time of synthesis [Xu and Agrafiotis, 2002];
- as the size of the search increases, so does the cost of performing the search;
- as the size of the search increases, it is likely that the accuracy of results emerging from HTS will decrease. The primary cause of potential accuracy loss is the miniaturization of the assay process required in order to make increasing throughput practical [Panfili, 1999; Bajorath, 2000]. Loss of accuracy results in process inefficiency, as compounds that are not therapeutically useful may remain in the process for further evaluation and potentially useful compounds may be rejected.

It is clear that, as the difficulty of discovering novel products to compete in crowded markets increases, the search must be widened if product output is to be maintained or increased. It is also clear that the primary limitations of *in vitro* HTS are physical. The requirement that compounds are synthesised and assayed in physical form limits throughput and increases cost.

*Virtual* HTS [Bajorath, 2000], also known as *in silico* HTS, is now in widespread use within the pharmaceutical industry. Again, roughly translating from the Latin of its origin, *in silico* refers to a process conducted '*in silicon*' or, to be more familiar, by using a computer. Increases in available computational power, including distributed or grid computing (*www.grid.org*), during the past two decades allow a target to be modelled and represented computationally. An electronic representation is gained via thorough examination of the individual binding sites across the target and their geometric relation to each other. The approximated representation of a typical target may be referred to as a *pharmacophore*.

Similarly, compounds may be synthesised and represented *in silico*. A similar representation to that used for pharmacophore creation combines the binding properties of many sub-molecular fragments that comprise each compound [Joseph-McCarthy, 1999]. *In silico* representations of both target and potential substrates allow HTS to be performed *in silico*.

Unlike *in vitro* HTS, computational power and storage capacity are the only limiting factors of *in silico* HTS, although the time required to search chemical space remains an issue. Material costs do not rise with the size of the search undertaken and accuracy does not diminish with increasing throughput. Accordingly, *in silico* HTS can search over $10^{12}$ compounds synthesised *in silico* and cover a wider range of chemical space in the initial search. *In silico* methods are cheaper, faster, and allow a wider search for novel therapeutic products than *in vitro* methods. However, *in silico* drug discovery involves some significant complications which will become evident in the course of this work.

### 2.1.3 Lead Generation & Combinatorial Chemistry

High throughput screening enables the identification of thousands of molecular compounds that are active against a specific biological target [Van Hijfte et al., 1999]. A series of hits identified by HTS proceed to the next stage of the discovery process, *lead generation*.

Hits emerging from the HTS process are likely to be active against a therapeutic target, but they bear little resemblance to the drugs available in pharmacies, surgeries and hospitals. Hits display a desired activity against the target, but their binding properties are likely to be suboptimal and their structures must be optimised against the target. Hits are considered as developmental backbones, not as eventual products, and are used to identify useful regions of chemical space. The subsequent task is to cover those useful regions with a diverse collection of compounds that display a similar interaction with the target. It is from these collections (*libraries*) that candidates for further development (*leads*) are selected to progress further through the discovery and design process to be optimised for the therapeutic aim [Böhm and Stahl, 2000; Beresford et al., 2002].

Once hits have been identified, the aim is to cover the region of chemical space around each hit with a diverse collection of novel compounds. The rationale behind this practice is simple. Compounds that inhabit the same region of chemical space have similar properties [Van Hijfte et al., 1999], both structural and therapeutic. Hits have been identified as being active against the target and, therefore, potentially capable of fulfilling the design objective. Covering the region of chemical space around each hit with novel compounds should ensure that the novel compounds themselves are likely to be active against the target, and that a new product developed from the collection is itself likely to be novel. The additional requirement that the collection is 'diverse' increases the likelihood that a compound selected from the collection will be different from others that have been selected to date. Sparse coverage of a diverse area ensures that, although a wide region is covered by novel compounds, time and resources are not wasted by the development of compounds that are almost identical in structure or effect.

A collection of novel compounds in the region around a hit is created by a process

known as *combinatorial chemistry* [Bannwarth and Felder, 2000]. To describe the process briefly, combinatorial chemistry involves the synthesis of each hit with combinations of additional monomers or sub-molecular fragments. For example, if a backbone compound is synthesised with all possible combinations of 100 monomers of type A and 100 monomers of type B, 10000 new compounds, or variations about the backbone compound, will be produced. Compounds in the new *combinatorial library* share the same backbone and, thus, are likely to be similarly active against the therapeutic target. They all differ in structure, binding properties and potential for further development. Each library is 'weeded' of compounds that display some measure of structural similarity beyond a certain threshold, to ensure that the region of chemical space around the initial hit is covered only by novel, non-redundant compounds that display potential to fulfill the therapeutic objective. It is from such collections that leads are identified.

### 2.1.4 Structure-Property Correlation Analysis

The interaction between HTS hits and the therapeutic target is seldom optimal and, therefore, neither is the likely therapeutic efficacy of a combinatorial library that covers the region of chemical space around an initial hit. Variations on a hit may result in improved interaction of combinatorial library members with the target, but the question remains as to which of the thousands of compounds synthesised around an initial hit to develop further. Those chosen are known as *development candidates*.

More is required of *candidate* compounds than high levels of desired interaction with the target. In order for a compound to interact with a therapeutic target successfully, it must first reach the target site in the body (*in vivo*). Ability to do so is governed by a large number of properties that relate to the interaction between a compound and the human system.

The design of an orally administered drug (or xenobiotic) intended to interact with a target site in the brain provides a good example. The drug must pass through the gut, avoid being broken down in the liver, be transported to the brain in the blood stream, pass through the membrane separating blood and brain, and arrive at the target site in sufficient quantity to perform the task required of it. The presence of the drug must not result in harmful secondary (side) effects. Candidates that pass all of the above criteria must meet the additional condition that they are novel, i.e. they are not covered by existing patents. This simple example, described in greater detail in [Beresford et al., 2002], demonstrates the complexity of the search space created by combinatorial chemistry, from which suitable compounds for further development are identified by elimination. A model of human-xenobiotic interaction is required in order to identify suitable development candidates from a collection of novel compounds, but modelling the entire system of human-xenobiotic interaction as a single problem is almost impossible in practice. The system is too complex for satisfactory models to be extracted from the relatively small amount of available knowledge regarding the interaction of known compounds and the human system.

An intractable problem can often be decomposed into a series of constituent subproblems that, when solved and recombined in a suitable manner, approximate to solving

the larger problem. Absorption, Distribution, Metabolism, Excretion, and Toxicity (ADMET) properties [Beresford et al., 2002; van de Waterbeemd, 2003] and their relationship to molecular structure provide the constituent sub-problems that comprise a larger model of human-xenobiotic interaction. The example above describes the progress of a drug through the body, which may be decomposed into a series of hurdles that a compound must pass in order for it to be considered a suitable candidate for further development. A screen created for each hurdle, e.g. a screen that identifies compounds able to pass the gut wall, can be used to reject a subset of compounds in the library on the basis that, in spite of being biologically active against the target, they are unlikely to reach it. The deployment of screens to weed combinatorial libraries of compounds that are unsuitable for further development improves process efficiency through the early identification of useful development candidates and the early rejection of compounds that do not possess the properties required of a successful therapeutic product.

*In vitro* screens are employed to observe the behaviour of novel compounds in laboratory conditions designed to replicate the conditions inside the body. Thus, the ADMET properties of a compound may be quantified and used to evaluate development potential. The limitations of *in vitro* screening are similar to those outlined for *in vitro* HTS (section 2.1.2). Many properties and relationships must be tested in order to assure the retention of only those compounds with development potential and the rejection of those that do not. The cost involved in synthesising every member of every combinatorial library created from HTS hits, building *in vitro* screens for several identification criteria and screening the library members in order to ascertain their developmental worth becomes prohibitively expensive as the search space widens.

The successful substitution of *in vitro* HTS by *in silico* HTS (section 2.1.2) is largely dependent upon the availability of sufficient computational resources. Both processes measure the same outcome in a similar manner, but *in silico* HTS allows a wider range of compounds to be assayed more rapidly and with less operational cost than *in vitro* HTS. When considering the substitution of *in vitro* ADMET screens, the situation is more complex and, rather than directly replicating the *in vitro* process, the *in silico* alternative to *in vitro* screening must perform its function in a different manner.

The computational complexity involved in modelling whether each compound in a library can, for example, cross a specific membrane limits its viability as an alternative to *in vitro* screening. Alternately, it is considerably less intensive to encapsulate prior knowledge regarding previous *in vitro* screens in a classification rule that relates compound structure to the screened property. It was impractical, even before the advent of HTS, combinatorial chemistry and the expanded search through chemical space that they make practical, to use *in vitro* screens to examine the binding affinity and ADMET properties of every compound created towards a particular therapeutic aim. Chemical knowledge, dating back to the 19th century, was used to create linear relationships between molecular structure and properties of interest [Hollinger, 1997]. As described in section 2.1.2, binding affinity is directly related to the location and nature of various binding sites across a substrate and the sites to

which it binds. The relationship between compound binding and molecular structure is used to model compound synthesis and binding reactions during *in silico* HTS.

As the search narrows, models that relate various aspects of molecular structure to its affinity to a target are used to evaluate the success or otherwise of making small changes to the structural features modelled during compound optimisation. Since the 1960s, the relationships modelled have been non-linear as well as linear [Hansch, 1969], which has prompted the introduction of computational modelling techniques. The field of modelling the relationship between compound structure and binding affinity is known as *structure-activity relationship* (SAR) analysis, a popular machine learning application in drug discovery [King et al., 1992; Burbidge et al., 2001].

The premise of relating molecular structure to binding affinity can be extended to a wide range of interesting properties that relate to molecular behaviour. The properties of interest include those most pertinent to ADMET investigation, such as protein binding tendency, bioavailability and toxicity. Other properties, such as the ability of a compound to permeate a membrane, are directly related to properties that are regulated by compound structure. This basic premise enables the creation of relationships between compound structure and biological properties. The practice of creating relationships that describe the effect of compound structure on properties of biological interest is generally known as *structure-property correlation* (SPC) analysis.

*In silico* SAR and SPC relationships are used to identify leads from combinatorial libraries using prior knowledge gained from the *in vitro* and, on occasion, *in vivo* examination of the binding affinities and ADMET properties of compounds with known structure. A primary benefit of *in silico* screening, in which the screens need not be physical, is that a selection of compounds may be assessed against several biological target properties simultaneously (*parallel screening*). This notion may be extended to consideration of a single target property that represents the presence, or otherwise, of several individual properties. Rather than examine the various biological properties that a compound must display in order to be developed into a new pharmaceutical product, it is becoming popular to screen for abstract properties, relevant throughout the discovery process, such as *drug-likeness* (or *lead-likeness*) [Rishton, 2003], in exactly the same manner as for biological properties, such as protein binding or membrane penetration. By doing so, the most pertinent question, i.e. whether a compound demonstrates potential for further development, is answered via the consideration of a single target property, its similarity to other compounds that were developed into novel therapeutic products. There exists a large amount of prior knowledge from which to create *in silico* drug-likeness screens, more so than there exists prior knowledge regarding the relationship between individual biological properties and molecular structure. Moreover, knowledge regarding the structures of commercially available pharmaceuticals is publicly available from sources such as the World Drug Index (Derwent Information, UK; *http://www.derwent.com*) and others listed in [Walters and Murcko, 2002]. However, the selection of compounds that most resemble previously successful development candidates may impair the selection of novel development leads. Nevertheless, the creation of *in sil-*

*ico* screens for drug-likeness is one of the most popular applications of supervised machine learning within the contemporary drug discovery process (§ 2.2.6).

Compounds with known structural attributes and measured ADMET properties provide prior knowledge from which to create *in silico* ADMET structure-property models. Compound samples relevant to the target property are provided via a complex series of *in vivo* and *in vitro* assays [Beresford et al., 2002]. Models are designed to predict a target property (or properties) when presented with new compounds, whose structural properties are known but whose target properties are not. In many cases, there is a limited amount of known data available with which to describe the relationship between a specific property and aspects of molecular structure.

The information required to create SAR and SPC models is fed backwards through the discovery process. *In vivo* and *in vitro* trials that measure properties, such as membrane penetration, bioavailability and toxicity, during the final stages of the discovery process provide qualitative and quantitative measures of ADMET properties. Whether the property modelled is discrete or continuous depends largely on the stage of the discovery process at which the model is used. During lead generation and early lead optimisation (which follows, cf. § 2.1.6), the modelling task is formulated to determine between discrete quantisations of the target property. For example, the modelled property may be represented as 'high / low' or 'good / bad'. As the discovery process focuses upon the optimisation of compound properties for the therapeutic aim, the modelled property becomes more descriptive, e.g. 'high / medium / low', or continuous. Continuous property modelling is employed when investigating the biological effects of small changes to molecular structure.

A limited amount of information is available from which to create structure-property relationships. A typical set of 'known' data will contain hundreds rather than thousands of compounds. The paucity of known data is the result of the difficulty and cost of obtaining precise ADMET properties from relevant compounds. For example, *in vivo* trials that produce precise measurements of compound toxicity in the human system are, for obvious reasons, limited in number. Occasionally, in cases where *in vivo* data are particularly limited, an *in vitro* screen is created to predict *in vivo* results and to classify further compounds as, for example, toxic or non-toxic in order to provide more information from which to create a model. It is envisaged that, as *in silico* screening becomes more prevalent in contemporary drug discovery, the information available from which to create structure-property relationships will increase. Presently, great efforts are made to create models from small collections of known data that are capable of generalising well beyond the range of information presented to them. These efforts, and how their success is measured, are described in section 2.2.

The use of computational modelling techniques to replace *in vitro* assays appears sensible and cost-effective, but requires the computational representation of molecular structure. Knowledge of the geometric relationship between the binding sites of a pharmacophore and potential substrates is sufficient to assay binding affinity in the HTS process. The relationship is explicit. The relationships between the ADMET properties of interest described

above and molecular structure are more abstruse and may prove more difficult to model. A wide variety of structural and chemical attributes may contribute to relationships between molecular structure and ADMET properties.

### 2.1.5 *In silico* Molecular Representation

The association of relevant experimental measurements with aspects of molecular structure is an intuitive concept. Each structural feature employed within an SPC relationship represents an associated axis in chemical space. Thus, *in silico* representations of molecular structure are used to reference chemical space when considering the similarity or otherwise of compounds within it.

An *explicit* representation of molecular structure comprises a vector (one for each compound) of descriptive attribute values, each of which relates to a specific structural property (cf. § 2.2.1). The primary aim of SPC analysis is to relate aspects of molecular structure to biological properties of interest, therefore, one must consider which elements of molecular structure best describe the tendency of a compound to fulfill a particular biological criterion.

Descriptive attributes available with which to describe molecular structure vary in resolution, from properties of the individual atoms and bonds that comprise a molecule [Kier, 1995], to properties of larger sub-molecular fragments [Dominik, 2000] and whole-molecular properties, such as molecular weight [Jurs et al., 1995]. All properties may affect fundamental aspects of pharmaceutical desirability, such as binding, absorption and solubility. Measurements of structural properties may be the results of laboratory assays or, if sufficient knowledge is available, may be calculated deterministically from those of small, well known molecular *fragments* that comprise a larger compound. Whole molecule properties, such as lipophilicity and hydrophobicity (which influence binding and membrane penetration) are employed also. Subsets of the available structural information are selected to provide as much relevant information as possible about the region of chemical space inhabited by a collection of compounds.

The representation of a molecular compound by a number of measured (or calculated) structural or biological properties is known as a *2D molecular representation*. Structural features are quantified, but their geometric relationship to one another is unspecified. A representation that specifies the geometric relationship of structural features, as well as some measure of their magnitude, is known as *3D molecular representation*. 2D molecular representation is used at lead generation and early lead optimisation (cf. § 2.1.6) stages of the discovery process because it has been demonstrated empirically to reference the local neighbourhood characteristics of chemical space in the most suitable manner for the models built at this stage [Van Hijfte et al., 1999].

If there existed a subset of explicit structural attributes that allowed chemical space to be partitioned according to any biological property, regardless of its nature, SPC analysis would not be such a necessary and challenging component of the drug discovery process. In actuality, there is no 'perfect' descriptor set for the creation of SPC relationships. The resolution and nature of the descriptors employed depends upon the biological property

that one attempts to associate with chemical space. Prior knowledge, gained from decades of examination and process development, is commonly employed to focus 2D representations of particular SPC relationships. Well-known SPC applications are accompanied by the knowledge that a well-defined subset of all available molecular information should be involved in the construction of predictive relationships on them, e.g. Lipinski's 'rule-of-five' for drug-likeness [Lipinski et al., 1997; Dominik, 2000]. Focused targets, such as whether a particular compound is able to pass through a specific membrane of the body, may be governed by a smaller subset of attributes that relate directly to the target [Zhao et al., 2003]. For example, it is more likely that small compounds pass through a membrane than larger ones, thus, molecular weight and surface area are likely to play an important role in the description of such a process.

The scenario wherein little prior knowledge exists regarding the relationship between a target property and chemical space results in an undesirable search across a large number of structural descriptors for descriptors related to the target property as measured over a small subset of compounds (cf. § 2.2.4). The results of such an approach are sub-optimal in the majority of cases (the selection problem is NP complete [Fröhlich et al., 2006; Russell and Norvig, 2003]) leading compounds with distinct ADMET properties to occupy overlapping regions of a chemical space defined by the chosen subset of molecular descriptors.

The difficult nature of relevant structural descriptor selection has motivated the development of whole-molecular structural descriptions, designed to encode relevant structural information in a uniform representational schema. Descriptions of three such representational schema, two of which are employed in later chapters of this work, are provided below and more detailed descriptions are provided in the literature, e.g. [Drewry and Young, 1999; Matter et al., 2001; Xu and Hagler, 2002].

*Volsurf* [Cruciani et al., 2000] is a relatively recent framework for the description of molecular structure, which departs from the explicit representation of molecular structure towards a wider, abstract representation. Volsurf descriptors originate from 3D molecular information, produced by measured interactions between a molecule and a number of probes applied uniformly to points across it (cf. GRID representation [Goodford, 1995]). The probes measure hydrophobic and electrostatic potentials, which affect molecular binding propensity, and their multiple application provides whole molecular information, rather than the fragmental extrapolation measures employed by many explicit representations. Other factors that similarly affect binding propensity, such as surface area and molecular weight, may be measured also. Image processing techniques are employed to convert the 3D GRID image of a molecule to a 2D string of real-valued information, some of which may be mapped back to the corresponding structural features. Two of the five GSK ADMET data sets, described in Chapter 3 and used to assess algorithmic performance in Chapter 4, employ Volsurf molecular representation. The other three sets employ subsets of explicit and fragment-based molecular descriptors.

Abstract molecular representations imply a mapping, of sorts, between molecular structural attributes and a simplified vector representation. *Structural keys* arose from the need

for a better, i.e. more computationally efficient, representation of chemical structure than traditional chemical notation, when screening structural information [Hansch, 1969]. A structural key is a binary string in which bits represent the presence of particular combinations of molecular features. For example, part of the string might be chosen to represent the presence of hydrogen bond donors, with attributes such as 'zero', 'one', 'two', '> two', and so on. Continuous attributes, such as molecular weights, are separated into bins.

There are two specific considerations to be made when employing structural keys. The first concerns the number of bits employed to represent each attribute. For example, in the case that a compound has two hydrogen bond donors, should the attribute of having one hydrogen bond donor be represented as well? The compound with two donors also has one, thereby sharing a structural similarity with other compounds that only have one. The second consideration concerns the choice of descriptive attributes contained in the bit string. The more molecular information (attributes) represented by the string, the more sophisticated the comparison between it and others will be. It will also be more computationally expensive to analyse. Conversely, too few descriptors, or the wrong type of descriptors, may lead one to draw an unsatisfactory relationship from the data. Generic structural key representations are used widely [MDL, 1994; Drewry and Young, 1999; Van Hijfte et al., 1999].

*Fingerprints* are a more recent development in abstract molecular representation, an example of which is employed during this work. The Daylight fingerprint method [James et al., 2000; Daylight, 2006] provides structural information over the whole molecule, whereas structural keys may represent attributes chosen by the user. Thus, fingerprints employed by pharmacologists may be compared to those employed by petrochemists [James et al., 2000; Daylight, 2006]. Structural keys employed in the two different disciplines are likely to feature different attributes and, therefore, could not. The generalisation stems from a representation in which individual attribute values are not related to specific structural properties. Data in fingerprint form cannot be used to examine specific structural elements responsible for classification, but may be used effectively to assess whole-molecular similarity rapidly and provide transferable results.

The Daylight method works as follows. Each compound is comprised of several patterns. Patterns are created of each atom, each atom and its nearest neighbours (including the bonds that join them), each group of atoms connected by paths of two bonds and so on, up to a predetermined path length limit. A fingerprint is a bit string of length $N$, in which all bits are initially un-set (set to zero). Each sub-molecular pattern serves as a seed to a pseudo-random number generator, which returns 4 / 5 distinct integers in the range $[1, N]$. The bits at corresponding string locations are switched on (set to one). The individual sub-pattern fingerprints are combined using a logical OR to yield a single fingerprint of length $N$. Each molecule is thus represented by a unique string of ones and zeros, resulting directly from its structure. Each fingerprint is of uniform length and represents all structural patterns contained by the molecule within the path length limit.

When querying such a fingerprint with a sub-structural pattern, e.g. to determine whether the sub-structure may serve as a novel developmental backbone [James et al., 2000],

the sub-structure may be assumed not to exist within the queried compound unless all of the sub-structure fingerprint bits match the bits of the queried fingerprint. This may be expanded to assess inter-molecular similarity in relation to the number of bits shared by the fingerprints under comparison [Dominik, 2000; Xu and Hagler, 2002; Holliday et al., 2002]. Small molecules, comprised of few sub-patterns, may yield sparse fingerprints, i.e. fingerprints with few bits switched on. In order to improve the information content (the ratio of bits switched on to fingerprint length), fingerprints may be 'folded'. The fingerprint is split into two, equally sized, sub-prints, which are combined by a logical OR. It is easy to see that a sub-structure that does not exist in the original fingerprint is also likely to be excluded by its folded counterpart. Of course, care must be taken in order to leave sufficient sparsity to retain structural individuality. A mean information content of $\sim 20\%$ is frequently considered suitable [James et al., 2000].

The concept of abstract fingerprints raises the consideration of how best to assess molecular similarity from an *in silico* representation. When compound structures are represented as vectors of real-valued measurements or descriptors, it is intuitive to employ the Euclidean distance or some other Minkowski distance in order to assess inter-molecular similarity or diversity. In the discrete chemical spaces represented by keys or fingerprints, which encode the presence of sub-structural elements in their bits, several measures exist with which to provide relevant measures of similarity or diversity [Holliday et al., 2002]. The Tanimoto similarity, for example, is a measure of similarity between binary strings of structural information. The Euclidean distance between two binary strings treats identical variable values (be they one or zero) as inter-string similarities. The interest, here, lies in existing structural properties that the two compounds have in common. The Tanimoto similarity only computes similarity based on what the compounds possess, i.e. the ones. It does so by dividing the number of ones in common between the two compounds by the number of ones that *could* be in common. This gives a normalised similarity ratio, which accounts for the number of ones that might be in common relative to those that are. Daylight fingerprints and the Tanimoto similarity are the subjects of work described in Chapter 6 of this thesis.

The identification of a relevant *in silico* representation of molecular structure is fundamental to the success of SPC analysis. There does not exist an 'ideal' subset of the available explicit molecular descriptors that performs well for all analysis problems of this nature. Thus, methods to select relevant descriptors, or even to obviate this requirement, are very much required in order to advance the field [Fröhlich et al., 2004, 2006].

### 2.1.6 Lead Optimisation

HTS produces hits, which are used to create combinatorial libraries of potential leads. The libraries are weeded of compounds that do not possess the properties required of a therapeutic product using *in vitro* and *in silico* screens. By this stage of the discovery process, the number of compounds under consideration for development has been reduced from $10^{12}$ (*in silico* HTS) to approximately $10^2$ development candidates.

The lead optimisation process contains many stages and involves significant compound

attrition. Once leads have been identified during lead generation, a rigorous series of compound synthesis, *in silico*, *in vitro*, and *in vivo* screening is undertaken in order to optimise whole molecular properties for the therapeutic aim. The contemporary lead optimisation process is described well by [Beresford et al., 2002]. To summarise, contemporary lead optimisation is a smaller, more focused, repetition of the combinatorial chemistry used to create candidates during lead generation. Instead of making single, specific changes to the structure of each lead and observing the results, as was done previously, multiple changes are made around each lead in order to cover the region of chemical space around the lead in a focused and detailed manner. The combinatorial libraries created at this stage of the process differ greatly in size when compared to those created directly after HTS. Libraries created during lead optimisation commonly contain in the order of $10^3$ compounds, whereas libraries created for lead generation can contain in the order of $10^6$ compounds.

After leads have been optimised, compound properties are optimised to produce maximum therapeutic effect whilst remaining suitable for ingestion. Virtual screens may be employed to select library members with properties suited to the therapeutic aim. The selected compounds are synthesised and evaluated with both *in vitro* and *in vivo* methods. Further compounds are synthesised around those tested and the process is repeated until no further improvement is apparent. It is likely that any remaining compounds will be active against the target, optimised for ADMET suitability, and not covered by existing patents. Once it is thought that a compound is ready for market, clinical trials begin to test the hypothesis and satisfy medical and safety regulations.

### 2.1.7 Performance Evaluation of SPC Classifiers

An improvement in SPC classification accuracy may involve an increase in one or more of the following categories:

1. generalisation accuracy;
2. classification throughput;
3. rapidity of classifier creation;
4. intelligibility of classification; and
5. ease of use for the non-expert.

Each application area within the discovery process may place different emphasis on any of the categories listed, but generalisation accuracy is frequently at the top of the list. It is a little misleading to list individual categories of improvement, because advances often involve a combination of the above. For example, a new method for SPC analysis may be regarded as successful if able to predict a target property with greater accuracy than the existing state-of-the-art, from the same amount of information, while taking the same amount of time, or less, to do so. The individual treatment of improvement types does help, however, during consideration of their importance to a particular process. This work

treats the creation of SPC relationships from the ADMET properties of small compound collections during early lead optimisation. It is from this perspective that the categories listed above are considered below.

The outright predictive accuracy of an SPC relationship may be misleading in certain scenarios that occur frequently when attempting to classify biological properties from small collections of known data. In many cases, it is not desirable that a classifier treats all data examples with equal importance. A good example arises in the creation of *in silico* screens. Should a new compound be incorrectly described as drug-like (a *false positive* classification), the result will be that the library it inhabits will be less efficient and the compound (and dependents, if used to combinatorially construct new compounds) will remain in the development process until potential rejection at the next synthesis-screening repetition. The magnitude of such a reduction in process efficiency is dependent on the number of compounds misclassified in such a manner and on how the combinatorial chemistry process acts upon those compounds once they remain in the process. Should a new compound be incorrectly classified as not drug-like (a *false negative* classification), a potential new drug is disregarded at an early stage of development, with an enormous potential cost. It is clear that accuracy, in this case, should not be calculated on the basis that both classes involved are of equal importance. It is unclear, however, how best to 'cost' misclassifications of each class. For example, it is of primary importance to retain leads when mining the output of HTS to identify backbone compounds for further development, but the false identification of non-leads affects the efficacy of the combinatorial chemistry process that ensues subsequently. During the creation of ADMET SPC relationships for lead optimisation, it is certainly important both to retain candidates for further optimisation and to reject non-viable compounds in order to prevent their continuation through the process of further combinatorial creation and, eventually *in vitro*, optimisation.

A complementary consideration to that of misclassification cost is that of balanced predictive performance assessment. As discussed above, the discrete, binary classification scenario requires that the SPC solutions obtained must deliver high classification accuracy on both the 'good' and 'bad' compound classes. For example, a solution that classifies everything as 'good' will not serve the intended purpose, as it is required to form part of a search by elimination. Likewise, if everything is classified as 'bad', the search ends and no compounds will be selected for further development. This obvious requirement may be challenged in the case of one compound class being far greater in size (both in terms of available, known data and in terms of potential encounters during classification) than the other. In such a circumstance, the assessment of a trained classifier must acknowledge the existence of a majority class. Otherwise, an apparently high predictive accuracy may mask poor performance on the minority class. Communication with collaborators at GSK during the course of this work suggested an 'industry-desired' prediction accuracy of $> 80\%$ on both classes, beyond which limitations of the data make further performance increase unlikely. From this brief consideration, it is clear that firm notions of 'useful' classification accuracy must be employed in order to evaluate the performance of SPC models created at

this stage of the design process. These issues are discussed in greater detail in Chapter 3 (§ 3.2.3).

The time taken to build a discriminant model (at this stage) is not particularly significant, largely because the amount of data available for training is, in general, small ($< 1000$ examples). Moreover, a classifier learned from the available data is likely to be applied to larger numbers of unknown examples repeatedly. Predictive performance is a key issue, thus extra time taken to construct an accurate classifier is likely to be rewarded several-fold. Nevertheless, the developmental process behind the construction of an SPC classifier requires creation and validation to take place within a reasonable period and thus, the time taken to select classifier architecture and the parameters that define it must also be taken into account.

The time taken to create a classifier for eventual use is not of primary importance, although time taken during R & D may well be. Ease of use during R & D by non-expert users is of similar importance. A similar and important area, in which improvement would be of great benefit, is that of intelligibility. Through detailed research, the computer scientist may understand a lot about how a technique has made its prediction. The end user, however, may not possess such knowledge and be unable to verify how the decision is constructed. If a new classification method is to be used by many different types of user, it must be able to deliver its results in a manner and format that all can understand. For example, this may include information regarding which compounds are primarily responsible for the form of an SPC relationship, or which descriptive attributes play the largest role during classification of further compounds.

Regardless of the nature of an advance provided by a new SPC classifier, it is important that it may be incorporated easily into the extant process framework. Any improvement in performance is welcome in such a challenging and potentially inefficient domain, but the domain itself is unlikely to change wholesale unless the improvement is of such magnitude as to render previous practices obsolete, e.g. high throughput screening. A new SPC classification method is more likely to provide such a paradigm shift in combination with wider changes to the domain and should be designed, therefore, to be integrated in a practical and *domain-relevant* manner.

### 2.1.8 Research Focus and Conclusion

As stated in the introductory chapter, this work examines the application of a recent technique, SVMs, to a specific area of the drug discovery process. The above overview of the contemporary drug discovery process allows the application to be placed in context. The area of the drug discovery process that provides the focus for this work is the creation of *in silico* SPC relationships during the early lead optimisation stage. The relationships are used to discriminate between compounds that belong to discrete classes of ADMET properties.

The description of the contemporary drug discovery process provided in this section suggests clearly that the replacement of *in vitro* processes with their *in silico* equivalents is paramount to the expansion of the drug discovery search. Machine learning is becom-

ing widely used within the pharmaceutical industry to perform tasks previously performed by *in vitro* screening, with the ultimate aim of an automated *in silico* discovery process that initiates upon the identification of a therapeutic target [Beresford et al., 2002; van de Waterbeemd, 2003]. The increasing use of data mining packages, such as Clementine [SPSS, 2002], and dedicated pharmaceutical modelling packages, e.g. Pharma Algorithms (*http://www.ap-algorithms.com/*), display the current trend towards *in silico* discovery. The aim will not be attained, however, unless *in silico* structure-property predictors are able to overcome the challenges posed by the domain. Before proceeding to the application of support vector machines to the prediction of ADMET properties, it is useful to place support vector machines in the context of contemporary machine learning as ADMET structure-property relationship analysis has been placed in the context of contemporary drug discovery here.

## 2.2 Contemporary Machine Learning for Drug Discovery

It is assumed that most readers are familiar with machine learning and many techniques described in this section. One of the aims of this thesis, however, is to provide useful information to those readers who may be from outside the computer science community, or those who have not dealt specifically with the application of machine learning to pharmaceutical data analysis. Accordingly, this section introduces machine learning, and supervised machine learning in particular, before proceeding to describe the application of machine learning techniques in general to the contemporary drug discovery process of section 2.1.

Machine learning involves the induction of relationships from sets of descriptive data examples. The relationships (or solutions) obtained may be applied subsequently to predict the behaviour of further examples drawn from the same data distribution. This section begins with an introduction to classifier inference, before describing a selection of machine learning techniques and their application to contemporary drug discovery.

### 2.2.1 Introduction

Throughout this work, vectors of descriptive values are referred to as data *examples* (or points). Descriptive values are referred to as data *attributes* and correspond to the individual pieces of information that comprise an example. Data attribute values may be numerical, textual, continuous and / or discrete. A useful concept in the description of machine learning is that of an *input space*, within which all possible data examples reside. For example, a collection of examples, each possessing three descriptive attributes, may be seen to exist as points within a 3-dimensional sphere, the diameter of which relates to the maximum distance between any two examples in the collection. Here, input space is denoted $X \subseteq \mathbb{R}^m$, where $m$ is the dimensionality of real-valued data examples drawn from input space. The notion of input space encourages the involvement of associated similarity metrics (e.g. Euclidean distance) with which to reference the space. This concept is analogous to the *chemical space* described in § 2.1.1 (p. 14).

The purpose of supervised machine learning is to infer a mapping of input space to a *target attribute*, $Y$, that describes (or categorises) its contents, $f : X \rightarrow Y$. As above, the target attribute can be numerical, textual, continuous and / or discrete in nature. Examples referred to as *known* examples have a corresponding target attribute value (or *class label*) that designates the class of data to which they belong, i.e. they are drawn from $X \times Y$. Known examples represent a finite data subset, e.g. of size $n$, from which the wider relationship between input space and the target attribute may be inferred.

$\mathbf{x} \in X$ An example vector of $m$ attribute values, $x_j, j = 1, \ldots, m$.

$\mathbf{y} \in Y$ A vector of $n$ class labels, e.g. $y_i \in \{-1, +1\}, i = 1, \ldots, n$.

A subset $S \in X \times Y$ of $n$ labelled data examples is defined as $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$. In the presence of a labelled data subset (or *training data*), the task of supervised machine learning is to infer the mapping $f : S_{\mathbf{x}} \rightarrow S_{\mathbf{y}}$ under the assumption that it reflects a wider mapping $X \rightarrow Y$. An *unknown* example does not possess an associated class label, or rather, the mapping function is not party to the class label. A mapping $f : X \rightarrow Y$, inferred from a collection of training data, may be applied to unknown examples in order to predict their target attribute values. Such a process represents *classification* when the target attribute is discrete, or *regression* when the target attribute is continuous.

Supervised learning is often referred to as *pattern recognition*, because most scenarios in which it is employed involve the recognition of significant relationships within a body of data drawn from input space. The supervisor, or target attribute, is used to represent prior knowledge regarding the form of the relationship in question. *Unsupervised* learning concerns the induction of significant, latent relationships from a body of data in the absence of a supervisor. Unsupervised learning methods are often employed to reduce the size or complexity of a body of data in order to elucidate an underlying distribution across its members. This may be via a reduction in the number of data examples (to provide a sparser, phenotypic representation of dense or clustered data), or a reduction in the number of data dimensions (to provide a more compact representation of highly descriptive data, e.g. for visualisation).

The research described in subsequent chapters concentrates on an application located early in the lead optimisation stage (cf. § 2.1.6) of drug discovery. Complex, small-scale SPC relationships are investigated to focus the discovery search towards those novel compounds most likely to fulfill a given therapeutic aim. Relationships are created from small collections (typically hundreds) of compounds, primarily due to a paucity of known data relevant to the areas under investigation. The classification required at this stage remains 'select / reject' rather than the prediction of a continuous property, which is often required later in the discovery process, e.g. when optimising molecular binding properties against a target. Thus, the target attribute is discrete and binary. A relationship is required to discriminate between those compounds that should remain in the design process for further investigation and development, and those that should be rejected in order to improve the efficiency of the design process. Despite a focus on binary classification, the majority of

methods described hereon can be applied to both multi-class discrete and continuous target attributes.

A generic SPC analysis problem with binary target attribute presents a set $S$ of $n$ example vectors, $\mathbf{x}_i \in \mathbb{R}^m$, $i = 1, \ldots, n$, each labelled according to a target attribute, $y_i \in \{-1, +1\}$. Each example vector contains $m$ attribute values that correspond to structural features or properties of a molecular compound. As for a generic supervised learning problem with binary target attribute, the aim is to map the example vectors to their corresponding class labels under the assumption that the mapping will *generalise* to the remainder of chemical space. The relationship between example attributes (molecular structure) and the target attribute (biological property) induced by the mapping is used subsequently to associate target attribute values with examples (compounds) drawn from the remainder of chemical space.

Formulated thus, *in silico* screening of compounds according to a target biological property is dependent upon:

- the identification of an adequate vector representation of molecular structure;
- an accurate relationship between a vector representation of molecular structure and a target property;
- the computational means by which to describe such a relationship; and
- a representative sample of labelled training data from which to derive the relationship.

From section 2.1, these requirements are not encountered easily and sub-section 2.2.4 describes the potential effects of this upon the creation of useful SPC classifiers. First, sub-sections 2.2.2 & 2.2.3 describe means by which SPC classifiers may be obtained via supervised learning.

### 2.2.2 Linear Classification

When attempting to partition input space according to a binary target attribute, a discriminant function may employ a threshold constant to relate training data examples to the target attribute. If the value of the discriminant function is greater than the threshold constant, one class is assigned. If it is lower, the other is assigned. A common form of linear discriminant function, which creates a linear hyperplane to separate two data classes in input space, is shown in equation (2.1). A set, $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, of linearly separable, binary labelled data examples may be separated according to their associated class labels ($y_i \in \{-1, +1\}$) by the following equation

$$y_i = \text{sgn}\left(\mathbf{w}^T\mathbf{x}_i + w_0\right) \quad \text{where} \quad \text{sgn}(\mu) = \begin{cases} +1 & \mu \geq 0 \\ -1 & \text{otherwise.} \end{cases} \tag{2.1}$$

Equation (2.1) provides a discriminant threshold that assigns the binary class label ($\mathbf{y} \in Y$) to the examples in $S$. The structure of the discriminant function is determined by the

example vectors ($\mathbf{x_i} \in \mathbb{R}^m$), a vector of *weights* ($\mathbf{w} \in \mathbb{R}^m$) and a constant *bias* term ($w_0$) that provides the threshold. The weights and bias of the discriminant function may be used to associate the same binary target label ($y \in \{-1, +1\}$) with an example, $\mathbf{x} \in X$, using the classification function

$$f(\mathbf{x}) = \text{sgn}\left(\mathbf{w}^T\mathbf{x} + w_0\right) \quad . \tag{2.2}$$

The classification function, $f(\mathbf{x})$, employs a hyperplane in the $\mathbb{R}^m$ input space that is oriented by the perpendicular weight vector $\mathbf{w}$ and located w.r.t. the origin of input space by the constant $w_0$. At

$$\mathbf{w}^T\mathbf{x} + w_0 = 0$$

the discriminating hyperplane separates the training data according to $\mathbf{y} \in Y$. Training examples with $y_i = +1$ lie on one side of the hyperplane and those with $y_i = -1$ lie on the other side. The notion of a discriminating hyperplane (or *decision boundary*) in input space is fundamental to the following description of supervised machine learning and its use for SPC analysis.

It remains to infer a set of weights and a bias term that orient the hyperplane to separate data in input space according to the target attribute - assuming, here, that it is possible to do so. An input space hyperplane that separates examples according to an associated binary target attribute records an *empirical error* of zero when its decisions are compared to the corresponding target attribute labels, or

$$E\left[f(S)\right] = \sum_{i=1}^{n} |y_i - f(\mathbf{x}_i)| = 0 \tag{2.3}$$

where $E\left[f(S)\right]$ represents the number of errors encountered in the application of $f(\mathbf{x})$ to the training set, $S$, of $n$ labelled training examples. A generic measure of the empirical error of $f(\mathbf{x})$ is provided by

$$E\left[f(\mathbf{x})\right] = \frac{E\left[f(S)\right]}{n} \quad . \tag{2.4}$$

The decision hyperplane is defined by the weights and bias that orient it. Accordingly, it is convenient to describe the associated empirical error in relation to the weights and bias term, $E(\mathbf{w}, w_0) \equiv E\left[f(\mathbf{x})\right]$. Of greater convenience is a small change to the present notation, in which the bias term, $w_0$, is added to the weight vector and a corresponding constant input attribute equal to one is added to all examples,

$$\mathbf{x}^T = (1\, x_1\, x_2\, \ldots\, x_m) \;, \quad \mathbf{w}^T = (w_0\, w_1\, w_2\, \ldots\, w_m) \quad .$$

Thus, the error associated with a set of weights becomes $E(\mathbf{w})$ and the linear decision

function becomes

$$f(\mathbf{x}) = \text{sgn}\left(\mathbf{w}^T\mathbf{x} + w_0\right) = \text{sgn}\left(\mathbf{w}^T\mathbf{x}\right) \quad .$$

*Hypothesis space* represents the collection of classifiers that map $f : X \rightarrow Y$. When applied to data drawn from an $m$-dimensional input space, the linear classification function employed thus far has an associated hypothesis space inhabited by all linear classifiers defined by a weight vector $\in \mathbb{R}^{(m+1)}$. The axes of hypothesis space are provided by the individual weights that determine classifier structure. A change to the dimensionality of input space, or to classifier structure, redefines the number of classifiers available with which to map $X \rightarrow Y$ and, thus, redefines their hypothesis space. Constraints on classifier structure, e.g. that all weights must be positive integers, affect coverage of hypothesis space by a family of classifiers. Classifiers, or hypotheses, that record zero empirical error on a set of training data represent the *version space* w.r.t. the training data and the hypothesis space that they are drawn from.

A typical classifier training algorithm draws an initial classification function at random from hypothesis space and aims to update the function parameters to minimize an error, or loss, function across the training data. For example, a training algorithm may begin with a classification function defined by a random set of weights and assess a measure of its performance over a set of labelled data examples. Upon the application of classifier to training data and subsequent performance evaluation (one training iteration), the structure of the solution is altered in a manner that is likely to improve performance after the next iteration.

The 'sum of squares' loss function provides a useful example of an error function with which to assess the separation, or otherwise, of data according to a target attribute. The sum of squares loss is the sum of squared errors between predictions made by a classifier and the corresponding target variable

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - f(\mathbf{x}_i)\right)^2 = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2 \quad . \tag{2.5}$$

Classifiers that populate hypothesis space and their associated empirical errors may be seen to define an error (or loss) 'surface' over hypothesis space. The location on the surface at which the minimum error occurs is known as the *global minimum*. Over a linearly separable data set, $S$, the error surface of a linear classifier is parabolic and has a single global minimum that lies in version space. An iterative training algorithm that updates the classification function (via its weights) in a manner proportional in magnitude to the associated empirical error and which alters the function to reduce future error behaves as if following a path along the error surface that, at each step (training iteration), proceeds in the direction of maximum negative gradient

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \eta\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}\bigg|_{\mathbf{w}_i} \quad \text{where } \eta = \mathbb{R}^+, \quad \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \left(\frac{\partial E(\mathbf{w})}{\partial w_0}, \ldots, \frac{\partial E(\mathbf{w})}{\partial w_m}\right)^T \tag{2.6}$$

$\eta \in \mathbb{R}^+$, $\mathbf{w}^{\text{init}} \leftarrow 0$, $w_0^{\text{init}} \leftarrow 0$, $k \neq 0$, $R = \max_i ||\mathbf{x}_i||$

while $(k \neq 0)$ // if errors are made on the training data
{
  $k = 0$
  for $(i = 1$ to $n)$ // iterate over the training data
  {
    if $(y_i (\mathbf{w}^T \mathbf{x}_i + w_0) \leq 0)$ // an error occurs
    {
      $\mathbf{w} = \mathbf{w} + \eta y_i \mathbf{x}_i$ // update weights
      $w_0 = w_0 + \eta y_i R^2$ // update bias
      $k = k + 1$
    }
  }
}

**Figure 2.1:** The Primal Perceptron Algorithm

In plainer language, the analogy is of a ball, which rolls down the steepest slope in the error surface, $-\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}$, until it comes to rest in the global minimum. A parameter optimisation scenario of this nature is known as *gradient descent*. The sum-of-squares error is favoured by many, as it is easily differentiable. Thus, the minimum of the loss term can be found easily and an appropriate optimisation procedure employed to find it.

Rosenblatt's linear *perceptron* [Rosenblatt, 1958] ties the above concepts of linear classification, empirical error, training and gradient descent together nicely. The linear perceptron accepts a real-valued input vector, $\mathbf{x} \in \mathbb{R}^m$, and compares the weighted sum of its elements to a threshold, $\theta$, in order to assign a binary class label. Accordingly, the perceptron classification rule can be described by equation (2.2) above. To classify an example, $\mathbf{x} \in \mathbb{R}^m$, with binary attribute label $y \in \{-1, +1\}$, a perceptron with input weights $\mathbf{w}$ computes

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{j=1}^{m} w_j \mathbf{x}_j - \theta\right) \equiv \text{sgn}\left(\mathbf{w}^T \mathbf{x} + w_0\right) \quad . \tag{2.7}$$

The weights of the perceptron solution are found by iterative updates to a random set of initial weights. Given a set of binary labelled training data, the primal perceptron training algorithm calculates weights as shown in figure 2.1.

A positive real-valued constant, $\eta$, controls the size of steps by which the weight update seeks to minimize empirical error and is referred to as the *learning rate*. This update procedure parallels the gradient descent method discussed above. If the data are linearly separable according to the target attribute, the error surface defined by all possible classifier weights and the corresponding classifier errors is parabolic and has a single global minimum. Thus, the perceptron algorithm obtains a solution that separates the data within a finite number of

weight updates (from a theorem of Novicoff, a proof of which is given in [Cristianini and Shawe-Taylor, 2000]).

### 2.2.3 Non-Linear Classification

Before proceeding to describe non-linear classification, it is useful to consider the *dual* form of the perceptron algorithm. The dual form arises from the following observation. If the constant learning rate is set to $\eta = 1$, upon encountering a misclassification, the primal perceptron algorithm update procedure adjusts the weight vector, $\mathbf{w}$, with the corresponding misclassified attribute vector (or training example). Accordingly, when the algorithm terminates successfully, the solution may be represented as

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \, y_i \, \mathbf{x}_i \tag{2.8}$$

where the $\alpha_i$ are positive and proportional to the number of times each $\mathbf{x}_i$ is misclassified during training. The vector $\alpha^T = (\alpha_1 \; \alpha_2 \; \ldots \; \alpha_n)$ may also be seen as an alternative to the weight vector, $\mathbf{w}$, when defining the classification function. For example, the binary classification function (equation 2.2) can be re-written as

$$\begin{aligned} f(\mathbf{x}) &= \text{sgn}\left(\mathbf{w}^T\mathbf{x} + w_0\right) \\ &= \text{sgn}\left(\left(\sum_{i=1}^{n} \alpha_i \, y_i \, \mathbf{x}_i\right)^T \mathbf{x} + w_0\right) \\ &= \text{sgn}\left(\sum_{i=1}^{n} \alpha_i \, y_i \, \left(\mathbf{x}_i^T\mathbf{x}\right) + w_0\right) \quad . \end{aligned} \tag{2.9}$$

The dual form of the perceptron training algorithm is shown in figure 2.2.

The binary classification scenario described thus far assumes that input space may be partitioned according to a target attribute in a linear manner. When the training data are not linearly separable according to the target attribute, no separating hyperplane exists in version space. A change to the form of classification function employed and, hence, the hypothesis space, is required in order to encounter a mapping that partitions input space according to the target attribute. Consider, for example, a polynomial decision function of the form

$$f(\mathbf{x}) = (\mathbf{w}'')^T\mathbf{x}^2 + (\mathbf{w}')^T\mathbf{x} + w_0 \tag{2.10}$$

where $\mathbf{w}'$ is the original $m$-dimensional weight vector of the linear classification function and $\mathbf{w}''$ is a vector containing an additional $k = \binom{(m-1)+2}{2}$ weights, which correspond to the components of the squared example vector $(\mathbf{x}^2)$. Now, the total number of weights that define the non-linear classification function is $d = m + k$ (with $k > m$). Let the input dimension $m = 2$ in the above polynomial classification function, thus $k = 3$ and $d = 5$.

$\alpha^{\text{init}} \leftarrow 0,\ w_0^{\text{init}} \leftarrow 0,\ k \neq 0,\ R = \max_i ||\mathbf{x}_i||$

```
while (k ≠ 0)
{
    k = 0
    for (i = 1 to n)
    {
        if (y_i (w^T x_i + w_0) ≤ 0)
        {
            α_i = α_i + 1
            w_0 = w_0 + y_i R^2
            k = k + 1
        }
    }
}
```

**Figure 2.2:** The Dual Perceptron algorithm

From equation (2.10), we have

$$\mathbf{w} = \mathbf{w}' \wedge \mathbf{w}'' = \left(w_1',\ w_2',\ w_1'',\ w_2'',\ w_3''\right) \quad .$$

The enlarged weight vector, in tandem with a composite input vector

$$\mathbf{v} = \mathbf{x} \wedge \mathbf{x}^2 = \left(x_1,\ x_2,\ (x_1)^2,\ (x_2)^2,\ 2x_1x_2\right)$$

yields a linear classifier that performs its calculation on a non-linear transformation of input space

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{v} + w_0 \quad .$$

A neater form maps the $m$-dimensional input attribute vector to the $d$-dimensional expansion employed to provide non-linearity. Consider $d$ mapping functions,

$$\phi_i(\mathbf{x}),\ \text{where } i = 1, \ldots, d\,,$$

each of which provides one component of the composite input vector, $\mathbf{v}$,

$$\begin{aligned}
\phi_1(\mathbf{x}) &= (x_1)^2 \\
\phi_2(\mathbf{x}) &= (x_2)^2 \\
\phi_3(\mathbf{x}) &= 2x_1x_2 \\
\phi_4(\mathbf{x}) &= x_1 \\
\phi_5(\mathbf{x}) &= x_2\,, \\
\mathbf{v} &= \mathbf{x} \wedge \mathbf{x}^2 \\
&= (x_1\ x_2\ (x_1)^2\ (x_2)^2\ 2x_1x_2) \\
&= (\phi_1(\mathbf{x})\ \ldots\ \phi_5(\mathbf{x}))
\end{aligned}$$

The $d$-dimensional space employed to describe the original input space in higher dimension is referred to as *feature space*. The $d$-dimensional *mapping vector* $\Phi(\mathbf{x})$, contains the individual mapping functions, $\Phi(\mathbf{x}) = (\phi_1(\mathbf{x})\ \ldots\ \phi_d(\mathbf{x}))$, and maps data residing in input space to feature space. A non-linear decision function with linear form may now be rewritten as

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{d} w_i\phi_i(\mathbf{x}) + w_0\right) = \text{sgn}\left(\mathbf{w}^T\Phi(\mathbf{x}) + w_0\right) \quad . \tag{2.11}$$

This representation of a linear decision function in feature space can be applied to the perceptron dual representation described earlier in equation (2.9) to yield

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}) + w_0\right) \quad . \tag{2.12}$$

The comparison of equations (2.11) & (2.12) reveals a potential trade-off between the number of training examples and the dimensionality of the mapping used to create a non-linear decision function. Equation (2.11) involves a sum over $d$ individual mapping functions of $\Phi(\mathbf{x})$, whereas equation (2.12) involves a sum over $n$, the number of training examples. It is clear from equation (2.10), above, that an increase in either the dimensionality of input space, $m$, or the complexity of the desired non-linear solution, e.g. the degree of polynomial used in equation (2.10), will increase the number of weights that define the classifier and, thus, the size of the sum in equation (2.11). A further consideration is that the number of examples required to describe a particular space in sufficient detail so as to associate a target attribute across it increases with the dimensionality of the space (cf. the *curse of dimensionality*, described well in [Bishop, 1995a]). A common by-product of complex non-linear representations of input space and an associated target attribute is an error surface in hypothesis space that contains *local minima*. Convergence at local minima causes an algorithm to cease training before the global minimum is reached, thereby creating a suboptimal solution. It is also of note that, despite being able to represent non-linear relationships between input space and a target attribute, the non-linear functions described above can only

approximate the true mapping, as their form is chosen prior to weight optimisation. The additional complications of non-linear classification highlight an important practical barrier to the statistical analysis of large descriptive data sets, which is of particular consideration on the high-dimensional data often encountered during *in silico* screening (cf. § 2.1).

A *kernel function* computes $K(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x})^T \Phi(\mathbf{z})$ for all vectors $\mathbf{x}$ and $\mathbf{z}$. Valid kernel functions are defined by Mercer's conditions [Mercer, 1909], which state the following validity condition for a kernel expansion of two vectors $\mathbf{x}$ and $\mathbf{z}$. There exists a mapping vector $\Phi$ and a kernel expansion

$$K(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x})^T \Phi(\mathbf{z}) \tag{2.13}$$

iff, for any function $g(\mathbf{x})$ such that $\int g(\mathbf{x})^2 d\mathbf{x}$ is finite, then

$$\int K(\mathbf{x}, \mathbf{z}) g(\mathbf{x}) g(\mathbf{z}) d\mathbf{x} d\mathbf{z} \geq 0 \quad .$$

In practice, the above condition requires that a kernel expansion of two vectors be symmetric and positive semi-definite. A valid kernel function represents implicitly the components of the mapping vector $\Phi$, i.e. the individual mappings, $\phi_1(\mathbf{x}) \ldots \phi_d(\mathbf{x})$, need not be calculated explicitly. Accordingly, equation (2.12) becomes

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + w_0\right) \quad . \tag{2.14}$$

With the addition of a valid kernel function, non-linear classification becomes more practical. A linear separating hyperplane may be created in any valid space, even one of infinite dimension. Furthermore, the error surface of a linear classifier has a single minimum. It is important to remember, however, that the calculation of $K(\mathbf{x}_i, \mathbf{x})$ in equation (2.14) should be affected as little as is possible by the value of $d$, as its computation is required $n$ times in order to evaluate the decision function. Otherwise, that advantage of an implicit representation of $\phi_1(\mathbf{x}) \ldots \phi_d(\mathbf{x})$ is lost.

### 2.2.4 Impediments to Generalisation

Hitherto, the treatment of classifier inference has assumed a linear separation of input space according to a binary target attribute. Data not linearly separable by a binary target attribute may be mapped to a higher-dimensional feature space in which linear separation is possible. Furthermore, classifier inference from a set of training data has been described under the assumption that $f : S \rightarrow \mathbf{y} \equiv f : X \rightarrow Y$ and that a measure of the empirical error is sufficient to suggest classifier performance across the rest of input space.

The training data frequently comprise a small subset of data drawn from input space. Thus, there may exist a disparity between the information afforded by the available training data and the 'true' relationship between input space and the target attribute. A paucity of

known data translates to a lack of knowledge regarding how a particular property partitions chemical space. Separation in chemical space of a small collection of compounds according to an ADMET property of interest may not be reflected outside the small region that these known compounds occupy, for reasons outlined below.

The production of training data represents transition from a hypothetical relationship between chemical space and target property to the existence of labelled data. Input space provides a descriptive framework for the examples of a particular process or population. Chemical space, defined by a particular representation of molecular structure (cf. § 2.1.5) and populated by all possible molecular compounds, provides a good example. The target attribute often represents the outcome of an experiment involving examples that populate input space, e.g. whether a molecular compound is toxic to humans, or whether it passes an *in vitro* ADMET screen. Prior to sampling, the relationship between chemical space and target property is either hypothesised, or obtained via prior observation of the relationship under examination. In either case, the relationship is not quantified in entirety prior to classifier creation (or there would be little need to apply the classifier to new examples). Thus, training data is often the result of experimentation, with the experimental parameters drawn from chemical space and the outcome of the experiment drawn from the target property. Under these circumstances, it is difficult to design experiments or conditions that sample $X \times Y$ uniformly. For example, there is a paucity of data to describe molecular compounds known to be toxic to humans (see below).

Viable molecular compounds may not populate chemical space in a regular manner, with compounds appearing in groups, or regions, rather than uniformly across the space. Similarly, the target property may relate to chemical space in a *localised* manner. That is, the target property may not partition chemical space smoothly. Rather, certain values of the target property may relate to discrete regions of chemical space. Combined with a non-uniform population of chemical space, this prompts consideration of how representative data are to be drawn from $X \times Y$. A complex relationship between chemical space and a target property may not be represented well by a small amount of data drawn uniformly from chemical space. Drawing more examples increases the likelihood that the data will reflect the true relationship between chemical space and target property. Limited training data, therefore, represents a challenge to successful prediction by an SPC classifier.

Sample irregularity may be thought of as sampling from $X \times Y$ in a manner that does not reflect the true relationship between chemical space and target property. For example, uniformly sampled compounds may not reflect input space locality as described above. The problem of class imbalance, which occurs when the majority of training examples belong to a single target attribute class, is particularly prevalent in SPC analysis. The requisition of training data via experimentation limits the sampled region of chemical space and constrains availability of the target property. For example, a true measure of toxicity must be measured *in vivo*. It is difficult to convince human subjects, and the regulatory authorities, to approve an assay in which the subjects may ingest toxic material. Accordingly, compounds known to be toxic to humans are in short supply because experimental restrictions prevent a bal-

anced sample of labelled chemical space. So difficult is it to obtain known toxic examples from physical assays that many toxicity training sets are themselves labelled according to models created previously from very small amounts of known data. Reliable training data at this stage of drug discovery is expensive to obtain and in short supply, therefore, the data available is used regardless of population balance. Furthermore, the outcome of a screen to label a collection of compounds is unknown at the time of their selection. Hence, many SPC data sets provide a wider sample of one (majority) class during training when it may be no more important to the process than the corresponding minority class (cf. Chapter 4). The production of data upon which to build *in silico* screens is described further by such sources as [van de Waterbeemd, 1995; Drewry and Young, 1999] and [Dominik, 2000].

The training data may represent a small region of the target property distribution across chemical space. Certain structural attributes may vary little over the training set and, hence, provide little information regarding the distribution of the target property across the sampled region of chemical space, i.e. they appear *redundant*. The information that they contain does little to relate the training compounds present to the class distinction. This may be the result of employing an irrelevant structural feature to relate chemical space to the target property, but it is more likely that the feature in question does not correlate with the target property over the sampled region. Regardless of the source of redundancy, such features should be removed from the training data prior to the creation of a relationship, as they contribute little to the representation of $X \times Y$ and lead the available training data to reference the space less well (cf. the curse of dimensionality [Bishop, 1995a]). The removal of redundant features is often referred to as *feature reduction* and suggests further removal of features which, although not redundant, either do not correlate with the target or which correlate with other features, thereby introducing a redundancy of sorts. The co-linear nature of seemingly unhelpful molecular attributes may limit the sensible application of feature reduction in order to remove them. Removing a feature altogether may remove valuable information provided by its combination with another. This situation is outlined practically in a paper by Gillet et al. [1998], in which sub-structural analysis is used to provide a relationship between a descriptor subset and biological activity against a molecular target.

The redundancy that results from descriptor co-linearity is a primary reason for the increasing use of machine learning techniques over traditional statistical techniques for SPC and SAR analysis. Well-known approaches to solving the problem (or lessening its effects) include methods to identify a subset of informative, non-correlated descriptors with which to represent the original data (feature selection) and data transformation methods, such as PCA, which transform the data to a representative set of orthogonal (uncorrelated) axes in a manner that retains the information inherent in the original data. The former approach is described well in [Böhm and Schneider, 2000] and several machine learning texts, e.g [Bishop, 1995a; Mitchell, 1997]. The latter approach is introduced later in this chapter (2.2.6) and is a popular data pre-processing and visualisation step [Franke and Gruska, 1995]. The application of feature reduction and feature selection prior to the creation of an SPC relationship is referred to as data *pre-processing* and forms part of the data treatment

process prior to analysis.

The effects of an erroneously sampled training data subset are described above, along with the notion that training data samples are the experimental embodiment of a hypothetical relationship between all of input space and the target attribute. The incorporation of erroneous attribute values in the sampled embodiment of $X \times Y$ is arguably the most common impediment to successful generalisation. This may be seen as sampling from an erroneous representation of input space, rather than erroneously sampling from an accurate representation of input space. Erroneous measurements of both the target attribute and input space attributes are commonly referred to as *noise*, or as being *noisy*. An algorithm that is evaluated solely according to its performance over the training data may be led to model erroneous features of the data that do not correctly reflect the underlying distribution from which they are drawn. Noisy, or incomplete, data sets are common and methods used to analyse the data must be able to consider this noise when making subsequent predictions based on the information provided. Noisy examples may be either incorrectly labelled with the target attribute or contain erroneous descriptive attribute values. Algorithms capable of creating complex, non-linear solutions risk the incorporation of training data noise into the inferred representation of $X \times Y$ and suffer poor generalisation accordingly. Although the classifier should base its prediction on the data put before it, it should not follow the data so closely that anomalies cause false classification. In other words, the algorithm and its solution should be *robust* to at least a small amount of false or missing information. When noisy data attributes are present within the training data, the notion of balancing empirical error with predictive generalisation has an easily interpretable form. That is, it may be necessary for a classifier to 'ignore' certain training examples to improve its inference of the mapping.

Unlike many other machine learning applications, e.g. microarray analysis [Tu et al., 2002], there is little noise within the attribute values employed to describe molecular structure, because many are calculated deterministically (see p. 17). Nevertheless, if the rules used to calculate compound attribute values from those of constituent fragments were developed for compounds dissimilar to those under investigation, the properties calculated may be erroneous. Conversely, compounds are frequently mislabelled in this scenario, because the property classes to which they belong are open to interpretation. For example, in order to make a binary classification, e.g. reject / retain, it is likely that a continuous property, e.g. a measured assay of membrane penetration, will be partitioned in order to do so. Threshold selection involves uncertainty and the continuum on which the threshold is based may also contain noise. Common sources of noise include procedural variation, such as measurements recorded on different subjects in differing quantities, and analytical variation, such as measurements recorded and analysed in different laboratories by different scientists. Differing compound classification conventions, themselves thresholds of sorts, may also complicate matters.

An increase in structural information, to allow a small collection of compounds to describe a clear separation of the chemical space that they inhabit, is not a particularly desirable approach in the majority of cases. A partition derived from such information may

overfit characteristics of the known data in relation to a larger unknown space, to the detriment of further generalisation (cf. § 2.2.2). An increase in structural information increases the dimensionality of chemical (input) space, which leads the available data to cover the space less well and which may require greater algorithmic complexity to describe separation of the known data.

The *capacity* of a classifier measures its stability on training data. A classifier with high capacity is able to label a set of $n$ binary labelled data examples in many of their $2^n$ possible label configurations. A classifier with low capacity is less flexible and will only be able to classify the examples when labelled in a smaller number of configurations. Classifiers with high capacity can separate more complex classification problems than those with lower capacity. One disadvantage of this is that it they become unstable as a result. The decision boundary of a classifier with high capacity is likely to change significantly if any aspect of the data on which it was trained is changed. Thus, classifier capacity plays an important role in the creation of successful SPC classifiers. One must balance the capacity required to create complex, non-linear partitions of chemical space with the restriction of capacity required so as not to produce partitions that only apply to the available known data and not to chemical space at large. A full discussion of classifier stability and its effects may be found in [Skurichina, 2001]. The over-reliance of a learned classifier on poorly sampled or uninformative training data, to the detriment of generalisation performance on further examples, is known as *overfitting*. The balance between drawing a useful representation of $X \times Y$ from the training data (minimising empirical error via high algorithmic capacity) and avoiding an over-reliance on the training data representation of $X \times Y$ (maximising generalisation, possibly via a lower capacity) is embodied by the process of *regularisation*.

A measure of expected performance on the unseen region of input space is required in order to quantify the generalisation ability of a learned classifier. The distribution of target attribute values across unlabelled data is unknown *a priori*, therefore, generalisation ability must be estimated. The expected generalisation error is the probability that a classifier, $f(\mathbf{x})$ will misclassify an input-target pair drawn at random from $X \times Y$. Estimates of generalisation performance may be obtained by partitioning a collection of labelled data into independent training and test sets. A classifier is created on the training partition, without reference to the test partition. The expected generalisation performance is obtained by evaluating an error function upon the application of the classifier to predict the target attribute on the test examples. Methods of obtaining independent data partitions for the estimation of generalisation performance are discussed later in Chapter 3. The drawback of the above estimation procedure is that, while the classifier is assessed on data unseen during its construction, the validation data itself represents a (small) subset of input space and, therefore, is similarly vulnerable to the sources of disparity described for training subsets above. Moreover, performance assessment is largely dependent on the error function with which error on the validation data is assessed.

Both compound and feature noise mask the 'true' separation of the known data according to the target property, therefore, the presence of noise within a small collection of known

data can reduce the generalisation performance of a relationship created from this description. To compensate, ADMET models must be robust to mislabelled compounds and noisy data, because a model that incorporates erroneous information in a small region of chemical space is likely to make erroneous classifications when applied to the rest of chemical space. An alternative to requiring robustness of the classifiers used on noisy ADMET data is to 'clean' the data of compounds that appear to have been labelled or described erroneously. Outlier detection methods (cf. 2.2.6) may be employed to weed a collection of labelled training compounds of those that appear distinct from the majority, which may be viewed as an attempt to 'typify' any relationship drawn from the data (cf. 4.2).

Just as attribute co-linearity impedes the obvious and sensible corrective measure of feature reduction, a similar impediment presents itself to the use of robust techniques via the potential existence of 'outliers' or 'singletons' in the data. Such examples appear individually or in small groups, normally some distance away from other compounds of similar activity. They may appear to be mislabelled or noisy, but they may be extremely important in the context of combinatorial chemistry. When designing a combinatorial library (§ 2.1.3, p. 18), the aim is to cover as much chemical space, with as few compounds, as possible. Outliers, therefore, offer an increase in the chemical space covered by fellow compounds of the same target attribute class. In addition, a lead developed from an outlying point may exhibit different behaviour to that expected from other compounds of similar target property, thus further potential for a novel drug. Outliers in known data present an impediment to generalised partitions of chemical space, but it is clear that outliers must not be ignored at classification time by any model used to classify the contents of a combinatorial library.

To summarise, when formulated as a machine learning problem, ADMET modelling for compound screening presents the following challenges:

- there is a paucity of available training examples, requiring that a model must be able to generalise well beyond the range of data from which it is created. Despite the shortage of training data, the resulting classifier may be required to screen large ($> 10^6$) numbers of compounds;
- there is no 'perfect' descriptor subset, with which to relate molecular structure successfully to biological properties under consideration;
- data examples are often represented by a large number of descriptive attributes. It is often the case that there are fewer examples in the available training data than there are dimensions of the chemical space that they inhabit;
- the data available from which to build predictors of a target property often contain erroneous, or noisy, measurements;
- there exist complex, non-linear relationships between molecular structure and target properties; and
- the nature of the target properties under examination (see § 2.1.4, p. 21), results in

there often being a significant difference in the number of examples available to represent each class of data.

The linear and non-linear classifiers described in § 2.2.2 and § 2.2.3 respectively appear vulnerable to these challenges, without the inclusion of methods designed to counter them. The next sub-section returns to the consideration of linear classifiers and introduces support vector machines, which may be seen to provide such methods to the classifiers considered thus far.

### 2.2.5 Support Vector Machines

Further to the impediments to generalisation outlined in § 2.2.4, two important aspects of linear perceptron training should be noted:

- perceptron training ceases as soon as the training data are separated according to the target attribute, which amounts to arbitrary placement of the decision boundary between two classes of data; and
- perceptron training does not converge if the training data are not linearly separable by the target attribute.

Linear classifiers, like the perceptron, that reduce empirical error suffer from arbitrary placement of the decision boundary. In work on statistical learning theory during the 1960s, Vapnik and colleagues formulated a confidence interval on the expected generalisation of such classifiers [Vapnik, 1995, 1998]. The confidence interval depends upon the Vapnik-Chervonenkis (VC) dimension, which is a measure of classifier capacity. The capacity of a set of functions, used to separate a finite set of data, is a measure of the maximum number of points that the set is able to 'shatter'. If a set of $n$ points with two class labels may be separated in all of their $2^n$ label configurations, they are shattered. A full derivation is available in [Vapnik, 1995] and described well in [Burges, 1998]. The VC dimension, $h$, of linear classifiers in an $m$-dimensional space is, therefore, $h = m + 1$.

A subset of linear classifiers, $\Delta$-margin hyperplanes, have VC dimension $h = min\left(\left[\frac{R^2}{\Delta^2}\right], m\right) + 1$, where $R^2$ is the radius of a hypersphere that encloses the training data and $\Delta$ is the margin of separation, or the minimum distance between a separating hyperplane and the examples either side of it. Therefore, a linear hyperplane that maintains maximal distance between itself and the nearest training examples on either side of it, the *optimum separating hyperplane* (OSH), is likely to generalise best to new examples drawn uniformly from $X \times Y$ [Vapnik, 1995, 1999].

An illuminating upper bound on expected generalisation error (equation 2.15) describes the balance between empirical error and structural error. The same bound is also highlighted by [Burges, 1998] and provides a worst-case bound on the number of errors made by a classifier on data drawn at random from the same distribution as the training data. For $0 \leq \eta \leq 1$, the bound on expected generalisation error of a linear classifier holds with

probability $1 - \eta$:

$$R(\mathbf{w}) \leq R_{emp}(\mathbf{w}) + \sqrt{\frac{h(\log(2n/h) + 1) - \log(\eta/4)}{n}} \tag{2.15}$$

where $\mathbf{w}$ are weights that define the classifier on $n$ training examples. $R(\mathbf{w})$ is the expected generalisation (test) error of the learned classifier, $R_{emp}(\mathbf{w})$ is the empirical (training) error and $h$ is the VC dimension of the classifier. The second term on the right-hand side of equation (2.15) is the VC *confidence* of the classifier and is minimised via minimising the VC dimension. By selecting the classifier structure with lowest VC dimension, i.e. a $\Delta$-margin hyperplane with largest margin of separation, statistical bounds on generalisation error are balanced against the empirical error in a process known as *Structural Risk Minimisation.*

The reduction of a bound on the estimated generalisation error of a classifier departed radically from the training paradigms of the time, such as those used to train linear perceptrons, which tended to minimise a cost or loss function on the training data alone (*Empirical Risk Minimization*) in the expectation that generalisation error would follow. It may be seen from the VC confidence term in equation (2.15) that this is true when training data are abundant, $h/n \rightarrow 0$, but not when limited data are available. When linear separation is not possible, a balance must be struck between reduction of VC-dimension and reduction of empirical error.

Both weaknesses of the perceptron algorithm are overcome by structural risk minimisation. By selecting the classifier that separates the training data maximally the placement of a separating hyperplane is no longer arbitrary. By balancing empirical and structural risks, a classifier may be trained on data that is not linearly separable according to the target attribute in a manner that reduces expected generalisation error. The following description outlines the practical implementation of the above, i.e. the process of OSH determination via the SVM algorithm on a generic binary classification task.

Boundaries parallel to the decision boundary and upon which the closest examples lie are referred to as *margin* hyperplanes. The perpendicular distance between them is referred to as the *margin of separation.* Maximising the margin of separation across a linear decision boundary (figure 2.3) creates an optimum separating hyperplane.
A linear decision boundary,

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0\,,$$

is flanked at unit distance by parallel margin hyperplanes

$$\mathbf{x}_i.\mathbf{w} + w_0 \geq +1 \qquad y_i = +1 \tag{2.16}$$

$$\mathbf{x}_i.\mathbf{w} + w_0 \leq -1 \qquad y_i = -1 \tag{2.17}$$

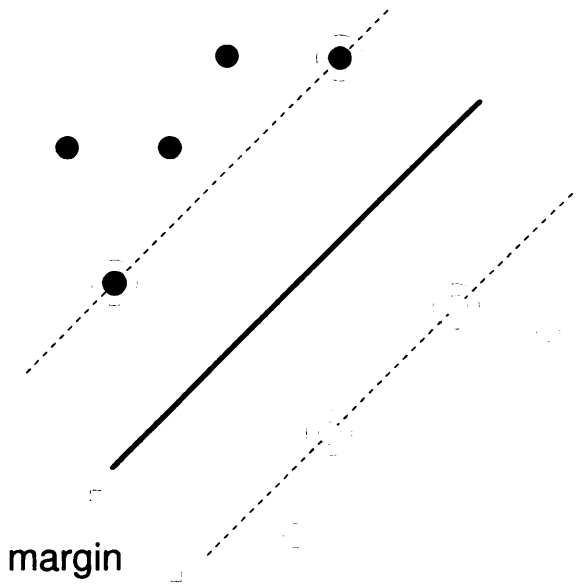


**Figure 2.3:** The Optimum Separating Hyperplane

Equations (2.16) and (2.17) form a single inequality

$$y_i(\mathbf{x}_i.\mathbf{w} + w_0) - 1 \geq 0 \quad \forall\, i\,. \tag{2.18}$$

The distance between margin hyperplanes is

$$\frac{2}{||\mathbf{w}||}$$

Thus, minimising $||\mathbf{w}||^2$ subject to the constraint of equation (2.18) maximises the margin of separation. This constrained optimisation is easier to treat when represented in Lagrangian formulation. Burges [1998] provides two reasons for the conversion. First, the constraints are easier to handle when placed on the Lagrange multipliers themselves. Second, the Lagrange dual treats training data as inner product pairs, the importance of which becomes apparent when treating non-linear SVM classification (cf. § 2.2.3). The constraints are multiplied by positive Lagrange multipliers $\alpha_i, i = 1, \ldots, n$ and subtracted from the objective function to provide the primal Lagrangian formulation,

$$L_P = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i y_i(\mathbf{x}_i.\mathbf{w} + w_0) + \sum_{i=1}^{n} \alpha_i \quad . \tag{2.19}$$

$L_P$ is minimised w.r.t. to $\mathbf{w}$ and $w_0$. The derivatives of $L_P$ w.r.t. $\alpha_i$ must vanish and $\alpha_i$ must remain positive. The constraints on the optimisation become:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{2.20}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.21}$$

The dual formulation of the above (known as the *Wolfe* dual) is solved by maximising $L_P$ subject to the gradient of $L_P$ w.r.t. $\mathbf{w}$ and $w_0$ vanishing and subject to the $\alpha_i$ remaining positive. The maximum of $L_P$, subject to the dual constraints, is reached at the same values of $\mathbf{w}$, $w_0$, and $\alpha$ as the minimum of $L_P$ when minimised subject to the previous constraints. When equations (2.20) and (2.21) are substituted into equation (2.19), the dual becomes:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i . \mathbf{x}_j \tag{2.22}$$

The solution for $\mathbf{w}$, gained by maximising the dual, is again

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \ .$$

The bias of the hyperplane is computed using the Kuhn-Karush-Tucker conditions, inequalities that hold for any convex, quadratic optimisation problem such as this. Details of the calculation may be found in [Burges, 1998; Cristianini and Shawe-Taylor, 2000].

Lagrange multipliers that remain greater than zero after optimisation represent those points, referred to above, that lie on the margin hyperplanes. The remainder lie further away from the OSH, but such that the original inequality of equation (2.18) holds, i.e. they lie on the correct side of the OSH. Points that lie on the margin hyperplanes are referred to as *support vector* (SV) points and are the only training examples required to support the decision boundary. The boundary would remain the same were the rest of the data removed and the algorithm re-run.

SVM classifiers may be created in feature space in the same manner as described for perceptrons in § 2.2.3. The second reason, given above, for use of the Wolfe dual formulation of SVM optimisation is that the training data are treated as inner product pairs. The original, linear formulation may be performed in a higher dimensional feature space thanks to an appropriate kernel transformation of inner products to feature space with a single function. The previous constraints on margin maximisation remain valid, because a linear separation occurs, albeit in a different space. The Wolfe dual becomes:

$$L_D = \sum_{i} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i . \mathbf{x}_j) \tag{2.23}$$

The constraints remain the same and the solution, when classifying an unlabelled vector $\mathbf{z} \in \mathbb{R}^m$, is of the form:

$$f(\mathbf{z}) = \sum_{i=1}^{nsv} \alpha_i y_i \Phi(\mathbf{x}_i) . \Phi(\mathbf{z}) + w_0 = \sum_{i=1}^{nsv} \alpha_i y_i K(\mathbf{x}_i, \mathbf{z}) + w_0 \tag{2.24}$$

where $\Phi(\mathbf{x})$ is, as before, a mapping from input space to feature space and $nsv$ is the number of support vector points, i.e. those with non-zero $\alpha_i$ after training. Unlabelled points are mapped into feature space using the kernel expansion of the inner product between each unlabelled point and the support vector points. One disadvantage of using Mercer kernels is that the data transformation to feature space is no longer explicit. Thus, it is particularly difficult to map the solution back to the original input space. Several 'standard' kernel functions enable the creation of both linear and non-linear classifiers and represent alternative formulations of other, familiar, supervised machine learning techniques. The following are all valid Mercer kernels which act on vectors $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{z} \in \mathbb{R}^m$.

| | |
|---|---|
| Linear: | $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z})$ |
| Polynomial: | $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z} + c)^p$ |
| RBF: | $K(\mathbf{x}, \mathbf{z}) = \exp -(\|\|\mathbf{x} - \mathbf{z}\|\|^2/2\sigma^2)$ |
| Sigmoid: | $K(\mathbf{x}, \mathbf{z}) = \tanh(\kappa\mathbf{x}^T\mathbf{z} - \delta)$ [only valid for certain values of $\kappa$ and $\delta$] |

For example, the generalised polynomial kernel function, $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z} + c)^p$, maps to a feature space constructed by ${}^{(m+p)}C_p$ monomials of the original data attributes, where $m$ is the cardinality of input space and $p$ the degree of polynomial employed. It is important to note that kernel functions are not limited to those displayed above and domain-specific kernels have been developed for several applications, of which [Watkins, 1999; Zien et al., 2000; Lodhi et al., 2000; Vert, 2002] and [Fröhlich et al., 2006] provide examples and are described and discussed in Chapters 5 & 6. Further discussions on kernel design may be found in [Cristianini and Shawe-Taylor, 2000].

As discussed in § 2.2.4, mapping a small subset of labelled data to a feature space of higher dimensionality in order to obtain example separation according to the target attribute is a recipe for overfitting the training data. In order to limit this occurrence, it may be preferable to attempt the creation of a separating hyperplane in the presence of noise or class overlap, i.e. to ignore certain training examples in data that is not linearly separable according to the target attribute and to create an OSH on the remaining data. *Slack variables*, positive variables that only exist when required to do so, place points that prohibit an OSH on the correct margin of an OSH created without them. Equations (2.16) and (2.17) are altered to include slack variables

$$\mathbf{x}_i.\mathbf{w} + w_0 \geq +1 - \xi_i \qquad y_i = +1 \tag{2.25}$$

$$\mathbf{x}_i.\mathbf{w} + w_0 \leq -1 + \xi_i \qquad y_i = -1 \tag{2.26}$$

$$\xi_i \geq 0 \, \forall \, i$$

A drawback of allowing points to breach margin constraints is that margin maximisation becomes unconstrained. A suitable constraint is provided by adapting the previous quadratic optimisation problem to minimise the objective plus an additional term, comprising the sum

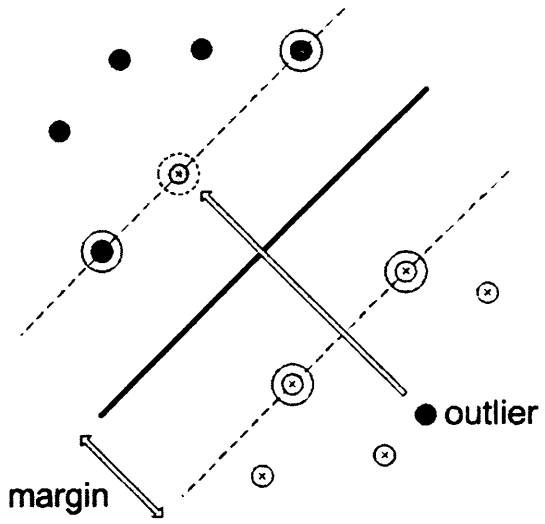


**Figure 2.4:** Slack Variables

of slack variables multiplied by a weighting factor.

$$||\mathbf{w}||^2 \quad \rightarrow \quad \left( ||\mathbf{w}||^2 + C \left( \sum_{i=1}^{n} \xi_i^k \right) \right) \tag{2.27}$$

Minimising the revised objective is a quadratic optimisation problem for $k = 2$ and $k = 1$. If $k$ is chosen as 1, neither the slack variables nor their Lagrange multipliers appear in the Wolfe dual. The sole change is that the $\alpha_i$ are upper-bounded by the constant, $C$. Constraints on the solution and the sum of Lagrange multipliers remain unchanged.

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i . \mathbf{x}_j \qquad 0 \leq \alpha_i \leq C \tag{2.28}$$

The constant, $C$, acts to allow a *tolerance to misclassification* to regularise faith in the data and generalisation to further data. the balance between margin maximisation and the empirical error involved on linearly non-separable data is embodied by the upper bound on expected generalisation of linear classifiers shown earlier (equation 2.15). Thereby, the SVM algorithm retains the powerful ability to produce robust solutions when trained on erroneously labelled, or linearly inseparable, data. Without such balance between empirical and expected generalisation error, the technique would risk overfitting the training data. If slack variables are weighted heavily (low tolerance / high $C$), a decision boundary closely related to the training data is created at the potential cost of diminished generalisation performance. Greater use of slack variables (high tolerance / low $C$) induces a decision boundary less faithful to the training data, with the resulting possibility of greater generalisation performance. The tolerance to misclassification that delivers best generalisation performance may be found empirically by cross-validation or via investigation of an upper bound on the expected generalisation error [Joachims, 1998a].

Full tutorials on the SVM method and the statistical learning theory that underpins it are provided in fundamental texts by Vapnik [1995, 1998] and an excellent introductory text on the subject by Cristianini and Shawe-Taylor [2000]. For quick reference, good descriptions of the SVM method from first principles may be found in [Burges, 1998], [Vapnik, 1999] and [Trotter and Holden, 2003]. The technique has been developed considerably over the past ten years, both theoretically and to improve its the practicality of its application to a variety of machine learning tasks. A selection of such developments, the majority of which are relevant to this research, are cited and described below.

SVMs have been successfully adapted to treat both regression [Vapnik, 1998] and one-class (outlier detection) [Tax and Duin, 1999, 2004] problems. Support vector domain description (SVDD) [Tax and Duin, 1999] is an SVM-based technique that can be used for data description, regardless of class labels, for the purpose of noise and outlier detection. The SVM quadratic optimisation is employed to minimize a sphere that contains the 'true' members of a data set. In a similar manner to a separating hyperplane, the sphere is supported by a subset of the data. The centre of the sphere may be calculated from the subset of 'support objects' that support the sphere, allowing fast comparison of new points with the training set. The SVDD method is a fast and accurate method of domain description and may be extended to non-linear scenarios using kernel functions, to which the Gaussian RBF kernel is well-suited. A soft margin is introduced via the use of slack variables. Applications of this method to real-world data are provided by [Tax et al., 1999; Tong and Svetnik, 2002] and [Morris, 2004].

Formulation of the algorithm for the classification of multiple discrete classes represents an area of ongoing research [Weston and Watkins, 1999; Hsu and Lin, 2001]. It is far more common to encounter the use of SVMs for discrete binary classification than for the classification of more than two classes. Unlike an ANN, it is not an easy task to train an SVM classifier to recognize more than two classes of data simultaneously, due to its linear formulation. Many multi-class SVM applications employ a one-against-all classification scenario, in which a classifier is built to discriminate each class from all others, with decisions combined upon classification of new examples [Hsu and Lin, 2001].

The representation of training data as inner products and the use of appropriate kernel functions for non-linear SVM classification reduce effects of the curse of dimensionality. The time dependence of the algorithm upon the number of data attributes treated is approximately linear, although this depends upon the kernel expansion employed for non-linear classification. A corresponding disadvantage is that the optimisation scales poorly ($\geq O(n^2)$) with the number of training examples. Accordingly, an SVM may take longer to train on large ($> 1000$) numbers of training data than many other supervised learning algorithms. Several methods have been developed to reduce the complexity of SVM optimisation and its time dependency upon the number of training examples. The most widely used method is that of *decomposing* the Hessian matrix of inner product values ($H = \alpha^T \alpha \ \mathbf{y}^T \mathbf{y} \ K(\mathbf{x}, \mathbf{x})$) over which the optimisation is performed. Much early (mid-1990s) SVM research concentrated on reduction of SVM training time. Several fast SVM

formulations have since been developed, the majority of which *decompose* the optimisation problem [Osuna et al., 1997]. The reduced training algorithm selects a subset of training data upon which an SVM is trained. The remainder of the data is classified by the resulting decision boundary. Examples that violate the boundary are added to the working subset at the expense of working subset examples that the boundary classifies correctly and which do not support the boundary. The algorithm is iterated until the working set correctly describes the optimum solution (no further replacement is required). Decomposition makes advantage of the fact that SVM classification only makes use of the SV points. It is, therefore, quicker to train on a small amount of training examples and optimise by classifying the rest than to optimise over the whole training set. Chang et al. [2000] assess decomposition methods for SVM approximation and reference the majority of work on the subject. A method of selecting strategically the training subset employed during decomposition is described by Hsu and Lin [2002].

Mangasarian and collaborators provide a different approach to computational resource reduction. Instead of decomposing the optimisation problem *per se*, an attempt is made to reduce computational demand via the use of inner products between the training data and a small, representative, sub-sample of the training data when calculating the kernel matrix. This 'slims' the kernel matrix considerably and is shown not to affect classification accuracy, even when the subset used contains as little as 1% of the original training data. Full descriptions of the method may be found in Lee and Mangasarian [2001]; Fung et al. [2002] and Lin and Lin [2003]. Further comment on this method is made in Chapter 4.

As will be discussed in Chapter 3 and demonstrated in Chapter 4, it is often desirable to treat one class of data in a different manner to another. Methods have been developed to allow this, the majority of which include an alteration to the upper bound on the Lagrange multipliers (equation (2.28)). Variable misclassification costs have been incorporated into the SVM framework by Lin et al. [2000]. A relatively simple theoretical approach that considers only the binary classification case adds variable costs to the slack variables in equation (2.27). In practice, the same effect may be achieved by associating a different value of the regularisation parameter, $C$, to the examples of each class [Osuna et al., 1997]. Applications of SVM weighting may be found in [Lee et al., 2001; Weston et al., 2003] and [Shin and Cho, 2003].

SVMs may be adapted for use in specialist domains by the deployment, or design, of suitable kernel functions. Polynomial and RBF kernels have been demonstrated to provide good results on a wide variety of non-linear classification problems [Blanz et al., 1996; Schölkopf et al., 1997; Hearst, 1998]. Methods for the design of domain-specific kernel functions are becoming increasingly popular as SVMs are applied to a wide variety of real-world applications. Schölkopf et al. [1998] describe methods of kernel function design that incorporate the effects of transformational invariance and local correlation into an image classification task. Jaakkola et al. [1999] introduces kernel functions that represent similarity between protein sequence, Vert [2002] to analyse phylogenetic profiles and Lodhi et al. [2000] to classify text documents (cf. Chapter 6). The benefit of involving local, rather than

global, correlation between members of the kernel matrix is demonstrated further by [Zien et al., 2000], when it is applied to an SVM used for the identification of particular regions in protein sequence. A similar technique was developed at the same time by Brailovsky et al. [1999] and is developed further to tackle correlation between examples as well as attributes. Local classification methods (such as the $k$-NN algorithm) and global training methods (SVM) are combined to produce an algorithm that recognises local correlation between training examples, without the complex solutions normally associated with lazy algorithms. This method is applied to structure-property correlation data in Chapter 5.

There exist optimisation methods for the selection of kernel function parameters, most notably by Chapelle et al. [2002]. By estimating a continuous bound on the expected generalisation error of an SVM in kernel space, a gradient descent procedure can be used to select the kernel parameters that offer the lowest expected generalisation error. This technique removes redundancy in feature space, thereby offering a method of feature selection. This method may also be used to estimate the regularisation parameter, $C$. The substitution of $C$ with a bounded regularisation parameter, $\nu$, is described in [Cristianini and Shawe-Taylor, 2000]. Although still a free parameter, $\nu$ is much easier to interpret, as it relates to both the fraction of SV points and the fraction of margin errors made on the training set. Although, as with $C$, cross-validation may be used to calculate a suitable value of $\nu$, bounds based on the fact that it can be used to control the number of margin errors may also be employed. Bounds on SVM generalisation have also been employed for free parameter selection by Joachims [2000]. Alternatively, heuristics for kernel free parameter selection are provided by Jaakkola et al. [1999] and Burbidge [2004].

SVMs are a *black box* technique and deliver little information about their predictions other than the predicted class label. Error bars on predictions are not readily available without reworking the algorithm within a probabilistic framework. If kernel functions are used in order to make a non-linear classification, the solution is not mapped simply back to input space. This lack of intuitive visualisation can make the SVM algorithm appear rather obscure to an unfamiliar user. Sollich [2000, 2002] reinterprets the standard SVM formulation, in order to visualise and optimise SVM kernel functions. Probabilistic interpretation allows the incorporation of approximate error bars on SVM outcome. In the probabilistic interpretation, it is sensible to attach class membership probabilities of less than one to points that fall inside the margin of the SVM solution. Points falling outside the margin get conventional probabilities of one. Further probabilistic treatments of the SVM algorithm are provided in [Bishop and Tipping, 2000; Chu, 2003] and [Shin and Cho, 2003].

Support Vector Machines have been introduced as a powerful, theoretically well-founded supervised machine learning algorithm, capable of dealing with large, high-dimensional, non-linear classification problems. Despite some operational weaknesses, which are outlined above, SVMs have consistently achieved performance competitive with the state-of-the-art in a wide range of challenging, real world applications. They have also generated a large and continued research following. To translate the theoretically high generalisation accuracy of an SVM to the complex, non-standard applications involved in drug

discovery is not an easy task, however, and one that has proved problematic to other machine learning techniques. Manallack and Livingstone [1999] ask whether ANNs live up to their promise for drug discovery (a question answered in part by Schneider [2000]) and the same question must be asked here of SVMs. The answer lies in the identification of an area, or areas, of the drug discovery process in which the high predictive ability of SVMs is a requirement and their relative disadvantages do not detract from their successful application. To this end, the above-listed range of developments to the basic SVM algorithm, all within the last 10 years, suggest a flexibility in the SVM framework that may assist the adaptation of its strengths for application to specialist domains.

A suitable application for which to use an SVM for data analysis may involve the following:

- binary classification (multiple classes and continuous target acceptable);
- high generalisation accuracy required;
- presence of noise and class overlap;
- little information required regarding classifier decisions;
- high dimensional data and potential class imbalance;
- classifier used to classify large numbers of new examples from relatively small number of training examples;
- supervised machine learning already used as state of the art; and
- the potential for domain-relevant treatment to improve performance.

The construction of SPC relationships to provide *in silico* screens during the lead optimisation stage of the drug discovery process appears to involve the majority of the above circumstances. With both the requirements of SPC analysis and the abilities of the SVM algorithm placed in context, it becomes clear that SVMs *should* be a useful tool for both this area of drug discovery and those beyond. Before placing this observation to the test (cf. Chapter 4), there follows a brief review of the use of several familiar machine learning methods, both supervised and unsupervised, within the contemporary drug discovery process.

### 2.2.6 Machine Learning in Drug Discovery

The use of machine learning within the drug discovery process has proliferated since the 1980s. The main reasons for the increase, and the gradual replacement of traditional statistical analysis methods by machine learning, are threefold. First, the computational power required to compute the complex, non-linear solutions produced by machine learning techniques has only become readily available over the past twenty years. Second, the use of non-linear modelling techniques in drug discovery has been made obligatory by HTS and

combinatorial chemistry, which themselves arose from technological developments made during the past twenty years. Before HTS and combinatorial chemistry widened the search for novel compounds and formulated it as a search by elimination, *in silico* screening was not required to the extent that it is today. Finally, the development of techniques such as neural networks, decision trees and genetic algorithms, all capable of modelling complex, non-linear relationships that are intractable when attempted by traditional statistical modelling techniques, has allowed those not involved in computer science research to apply machine learning to a wide range of problem domains.

The role of machine learning and *in silico* library screening in combinatorial drug discovery is described by a large number of discussion papers on the subject. Relevant recent publications include [Drewry and Young, 1999; Joseph-McCarthy, 1999; Van Hijfte et al., 1999; Böhm and Stahl, 2000; Matter et al., 2001; Xu and Hagler, 2002; Böcker et al., 2004] and the special journal issue of Schneider and Downs [2003]. All contain a wide range of references on the subject and provide good explanations of the techniques involved in combinatorial chemistry and the motivation behind current library design methods. Informative applications of machine learning techniques (there are many poor applications) to *in silico* screening and library design are reported by Manallack and Livingstone [1999]; Schneider [2000]; Sadowski [2000]; Burbidge et al. [2001]; Trotter and Holden [2003] and, most recently, Fröhlich et al. [2006]. These papers are set apart from the growing number of such publications because they provide clear descriptions of state-of-the-art screening techniques alongside information regarding the application that remains comprehensible to those outside the immediate field of drug discovery. Compound classification as a machine learning application was examined during the 2001 KDD Cup competition, in which competitors investigated the application of machine learning to the prediction of compound binding to Thrombin (*http://www.cs.wisc.edu/ dpage/kddcup2001/*) .

Chapter 4 describes SVM performance alongside a selection of other supervised learning methods, when applied to ADMET classification tasks provided by GlaxoSmithKline. Prior to their use in the comparison and further reference later in this work, brief descriptions of several machine learning techniques, both supervised and unsupervised, are provided below. Several other techniques, which are not the focus of this research, and their application to areas of drug discovery are cited also. Citations of published work that describes their application to pharmaceutical classification are provided.

During the course of this work, SVMs have been applied to data drawn from the drug discovery process on an increasingly regular basis. Drug discovery has been recognised by the machine learning community as a source of challenging data upon which to test machine learning algorithms and it remains apparent that *in silico* drug discovery requires input from a diverse range of sources in order to achieve its full potential within the drug discovery process. The application of the latest machine learning methods to solve drug discovery problems, such as effective structure-property relationship analysis, has a large contribution to make towards this objective.

An early application of SVMs to drug discovery data, and the first to emerge from this

work, was reported by Burbidge et al. [2001]. The SVM$^{light}$ implementation of Joachims [1998a] was applied to the publicly available [Blake and Merz, 1998] QSAR data of King et al. [1992] and its performance compared to other supervised machine learning techniques. The benchmarks included three neural network architectures (cf. p. 57), a radial basis function network (cf. p. 59), a C5.0 decision tree (cf. p. 62) and a one-nearest-neighbour classifier (cf. p. 63). The SVM comfortably outperformed all except a neural network with manual capacity control. The difference in estimated generalisation performance between an SVM with RBF kernel function and the neural network was negligible (0.3%), but the network parameter tuning and training time was an order of magnitude larger, due to the extensive manual tuning required to control the number of hidden nodes and prevent overfitting by the ANN. The conclusion reached was that the SVM demonstrated great potential for QSAR analysis.

Trotter et al. [2001] introduced the application of SVMs to SPC analysis, using BBB data similar to that described in Chapter 3. This paper, intended as an introduction to both technique and application for industry consumption, employed an early version of the RoCKET project [Buxton et al., 2002] SVM implementation, also described in Chapter 3, and compared performance against ANN, RBF, C5.0 and $k$-NN methods. Overall predictive accuracy of the RBF-SVM was higher than that of the other methods, but the increase was gained via better generalisation performance on a class that represented the majority of the training data. The class in question was that which would be most desirable to retain in the selection process, therefore, the effect of class imbalance was noted but neither evaluated nor discussed further, as it is in Chapter 4 of this work. In retrospect, the experimental practice employed to set SVM free-parameters (a single constant was used to set RBF width) and to assess generalisation performance (only overall generalisation accuracy was considered) lead the high-capacity RBF kernel SVM to concentrate more than is wise on the majority data class. Nevertheless, the work introduced the potential benefits of SVM application to problems of SPC analysis and, in reporting performance on both data classes, lead to the improved treatment of data imbalance seen in later chapters of this thesis.

A further, comprehensive introduction to the SVM technique and its potential role in SPC analysis was provided by Trotter and Holden [2003], who combined a concise introduction to the SVM technique from the perspective of optimal linear classification, with a review of machine learning for SPC analysis and a performance comparison similar to that of Trotter et al. [2001] but on the BBB, Bioavailability and Protein Binding data described in Chapter 3 and using an experimental practice similar to that also described in Chapter 3, which aims to assess generalisation performance in a manner that acknowledges the often large imbalance in training data class populations. The sizeable effects of training data class population imbalance were observed and the use of PCA data reduction (cf. p. 66) suggested as a means of improving predictive accuracy on a minority data class via its stronger representation in input/feature space. Interestingly, this approach is demonstrated not to be particularly effective in Chapter 4, section 4.1 of this work.

These methods, and others like them [Doninger et al., 2002; Byvatov et al., 2003; Yap

et al., 2004], follow a familiar pattern, often observed during the introduction of a new method to an existing application domain. The new method is applied alongside other methods that constitute a state-of-the-art to data drawn from the application domain, but measures to adapt the methods applied to the specific challenges of the domain are seldom taken. In other words, the methods assessed are applied to the domain in an 'off-the-shelf' manner and such early works provide a proof-of-concept that encourages further consideration of how best may the new method be successfully applied to the application domain. The research hypotheses of this thesis, which are stated in Chapter 1, reflect this procedure. A further pattern is that all of the work described above assesses the potential contribution of the SVM algorithm solely according to the predictive accuracy of its classifiers, over other potential benefits shown during its application to other domains, such as the incorporation of domain-relevance into the kernel transformation [Watkins, 1999; Jaakkola et al., 1999; Lodhi et al., 2000; Vert, 2002].

In recent years, work that applies SVMs to the drug discovery process in a less straightforward manner has begun to appear. A particularly interesting example is the application of an SVM within a query learning framework [Campbell et al., 2000] to QSAR data by Mathieson [2001] and Warmuth et al. [2002]. An SVM is shown to perform well on heavily class imbalanced, sparsely represented Thrombin binding data set - related to that used during the KDD Cup 2001 (cf. p. 54) - by iteratively requesting the class labels of unlabelled data examples identified as important to generalisation (those within the margin hyperplanes of an SVM trained on a subset of labelled data at each iteration). The idea that the learning algorithm is trained in the presence of unlabelled compounds, synthesised *in silico*, and directs which of those compounds should be labelled by *in vitro* screening in order to join the training set and improve generalisation performance would represent a radical departure from present methods of training data production in the design cycle of combinatorial chemistry [Eriksson and Johansson, 1996; Drewry and Young, 1999]. Algorithms that employ unlabelled examples alongside labelled examples during training, so as to create a classifier more likely to generalise well to unlabelled data in future, are known as transductive, rather than inductive, and several formulations exist for the transductive inference of SVM classifiers [Vapnik, 1998; Bennett and Demiriz, 1998; Joachims, 1999; Jaakkola et al., 2000]. Further work on transductive approaches to drug discovery are introduced by Weston et al. [2003]. An SVM is assessed against a method of transductive inference on a the same sparse, binary representation of 3D molecular structure as used in [Mathieson, 2001] (Thrombin binding data; KDD Cup 2001; see p. 54), in a manner designed to overcome the high cardinality, binary representation of molecular structure and extreme class imbalance in the training data (only 42 of 1909 compounds bind to Thrombin as required). Further application of transductive, or semi-supervised, learning methods to analyse large combinatorial libraries at lead generation stages of the drug discovery process is the subject of work by a research consortium at the Rennselaer Polytechnic Institute, New York, USA [Embrechts et al., 2003].

Another development of standard SVM practice for specific application to pharmaceu-

tical data introduces a kernel transformation that maps directly from a labelled graph representation of molecular structure to a feature space constructed from relevant similarities between molecular structures [Fröhlich et al., 2005, 2006], which obviates the traditional descriptor selection problem that may afflict QSAR and SPC analysis [Eriksson and Johansson, 1996; Drewry and Young, 1999]. This method is described in detail during Chapter 6 of this thesis, alongside the introduction of new work on the kernel representation of relevant molecular similarity. The suggestion that present levels of structure-property classifier performance may be improved via the representation of molecular structure in a feature space constructed from relevant measures of inter-molecular similarity is subscribed to by this work and, alongside transductive approaches, provides a particularly promising avenue of further investigation. Additional work of potential interest includes that of Burbidge [2004], which concerns heuristic methods of SVM application, including online adaptation of kernel free-parameters during training and early-stopping criteria, to data of various molecular representations drawn from lead generation in work performed under the same research consortium as the work in this thesis [Buxton et al., 2002]. The recent work of Wilton et al. [2006] also describes the role of SVMs in lead generation.

Machine learning was applied to structure-property analysis in drug design for at least two decades prior to the advent of support vectors machines. The remainder of this chapter describes several widely used machine learning methods, both supervised and unsupervised, and their application to drug discovery. **Artificial neural networks** (ANNs) have become a particularly popular method of constructing relationships between molecular structure and properties of biological interest [Manallack and Livingstone, 1999]. The most widely used ANN methods in drug discovery are *backpropagation* neural networks and *radial basis function* (RBF) networks. Unsupervised neural network methods, e.g. *Kohonen* networks, are also applied to data visualisation tasks. The history of ANN development and further details on a variety of neural network architectures and training methods can be found in texts by Hertz et al. [1991] and Bishop [1995a]. A useful text by Devillers [1996b] describes the use of all three ANN architectures for SAR analysis.

A typical ANN architecture consists of a several *nodes*, analogous to neurons in the human brain, connected by a series of weights. Each node receives a weighted sum of inputs, as described for perceptrons in § 2.2.2. In the majority of implementations, the binary perceptron threshold is replaced by a continuous *sigmoidal* function that, for a data example $\mathbf{x} \in \mathbb{R}^m$, receives the weighted sum of $m$ inputs

$$f(\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x})}} \tag{2.29}$$

The output range of a sigmoidal decision function is $[0, 1]$ and functionally dependent on the magnitude of the weighted input sum. An additional bias weight, $w_0$, with an associated constant input, $x_0 = 1$, provides a bias term.

The sigmoid unit is differentiable, therefore, *layers* of sigmoids, which feed their output forward from one layer to the units of the next, provide a differentiable non-linear error

surface. Internal (or 'hidden') layers encode input information and provide non-linear interconnection between input and output layers.

Output layer weights are updated in a manner similar to that employed in the linear perceptron, by an error measure calculated via comparison with the target attribute labels of the training data. The output of internal nodes, which encode the inputs rather than map them directly to the target attribute, may not be compared to the target attribute labels to obtain a measure of error with which to update them. Accordingly, a measure of error at the output layer is fed backwards (*back-propagated*) through preceding layers to provide the basis for internal weight updates. The ability to update hidden layer weights without explicit comparison of their output with the target attribute labels of the training data, thus providing a continuous error surface across all network weights, represents the solution to a problem that almost halted neural network research during the 1970s [Mitchell, 1997].

Gradient descent optimisation may be employed to update network weights according to an appropriate loss function over the training data. It is less complex to update network weights after each training example has passed through the network, instead of allowing the entire training set to pass through the network before updating the weights (cf. sum of squares, p. 33). This approach, *stochastic gradient descent*, provides an acceptable approximation to gradient descent across network weights, while accelerating convergence and reducing problems associated with local minima in the network error surface [Bishop, 1995a; Mitchell, 1997]. The network is trained repeatedly until a predefined *stopping criterion* is fulfilled. The stopping criterion may be, for example, a preset error threshold, time limit, or number of training iterations, i.e. the point beyond which any further performance improvement is expected to become negligible.

The flexibility of ANN architecture and, thus, their potential capacity requires painstaking and heuristic architecture and free parameter selection (*capacity control*) in order to avoid overfitting the training data and provide good generalisation performance within a reasonable limit of complexity. The complex, non-linear solutions available from a multi-layer network correspond to high algorithmic capacity. Hence, the possibility arises that an ANN may overfit the training data, by including noise, to a considerable degree. The capacity of a multi-layer neural network increases with the number of nodes in internal layers. Regularisation is performed by balancing the number of nodes necessary to map input examples to their target attribute labels with an observed measure of overfitting. If, despite high accuracy on the training data, performance is poor on an independent validation set during training, the network may be pruned of some hidden layer nodes prior to the resumption of training.

The total number of weights in most networks is much greater than the number of input weights. The corresponding error surface over all possible ANN classifiers available from the network weights (hypothesis space) may be complex and non-parabolic in consequence. Accordingly, ANNs may converge to local minima in the error surface during training, which may trigger a performance-related stopping criterion. This problem may be overcome by adding a 'momentum' term to the weight update. Within a gradient descent optimisation

framework, the new update simulates a ball, rolling down the error surface, the momentum of which carries it through local minima. Other potential concerns regarding the practical application of neural networks include the reproducibility of results, due largely to random initialisation of the network weights and variation of stopping criteria [Bishop, 1995a], and lack of information regarding the classification produced. The latter may be remedied via the combined application of ANN and an interpreting technique in a hybrid system (see below, p. 2.2.6).

Several texts provide detailed introductions to the ANN technique. Among the most informative are those by Hertz et al. [1991] and Bishop [1995a]. All include derivations of the sigmoid training rule and the backpropagation algorithm. Mitchell [1997] provides a clear derivation of the backpropagation training algorithm in relation to gradient descent. These sources also detail the history of ANNs, from early linear perceptrons to the present day, outlining some of the problems faced in reaching the current level of development. Early neural networks are covered well by Duda et al. [2000]. Manallack and Livingstone [1999] provide an informative review of the use of various ANN architectures for drug discovery and discuss both the problems encountered in the application of ANNs to drug discovery and their potential solutions. Kövesdi et al. [1999] provide a detailed application of neural networks to SAR analysis. This paper is domain-specific, but makes interesting points about the use of neural networks in a situation similar to the application considered by this work and is notable for a large number of useful citations. The introduction of ANN methods to SAR analysis is nicely documented by a collection of papers made available online by Igor Baskin (*http://org.chem.msu.su/people/baskin/neurchem.html*) and general acceptance of ANNs as a useful technique for drug discovery is suggested by [Schneider, 2000]. That it appears to take almost 20 years between initial applications of ANNs to structure-property relationship modelling and publications such as [Manallack and Livingstone, 1999] and [Schneider, 2000] serves to highlight the importance of successful interaction between the field of computer science and the domains to which its developments are applied.

**Radial basis function** (RBF) networks approximate the training data via the linear combination of basis functions in an architecture similar to that of ANNs. A typical such basis function describes a Gaussian distribution located in input space with centre $\mathbf{c} \in \mathbb{R}^m$ and width determined by a constant $\sigma$. Each function approximates a local area around its centre and its contribution to the approximation becomes weaker with distance. A basis function acts on a point $\mathbf{x} \in \mathbb{R}^m$ in the following manner

$$B(\mathbf{x}, \mathbf{c}, \sigma) = \exp\left(-\frac{||\mathbf{x} - \mathbf{c}||^2}{2\sigma^2}\right) \quad . \tag{2.30}$$

An RBF network consists of a weighted sum of the outputs of several basis functions located across the input space, which is directly analogous to the non-linear mapping described earlier in § 2.2.3. To classify an unlabelled example, the network decision boundary is provided by a threshold on the weighted sum of basis function contributions. For a point, $\mathbf{x} \in \mathbb{R}^m$, and a collection of $d$ basis functions with associated weights $w_i$, centres $\mathbf{c}_i$ and

widths $\sigma_i$ ($i = 1 \ldots d$), the RBF network output is

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{d} w_i B_i(\mathbf{x}, \mathbf{c}_i, \sigma_i) + w_0\right) . \tag{2.31}$$

The basis functions, $B_i(\mathbf{x}, \mathbf{c}_i, \sigma_i)$, are equivalent to $d$ individual mapping functions of the example vectors, $\mathbf{x}_i$. Therefore,

$$\phi_i(\mathbf{x}) \equiv B_i(\mathbf{x}, \mathbf{c}_i, \sigma_i)$$

and

$$\begin{aligned} f(\mathbf{x}) &= \text{sgn}\left(\sum_{i=1}^{d} w_i B_i(\mathbf{x}, \mathbf{c}_i, \sigma_i) + w_0\right) \\ &= \text{sgn}\left(\sum_{i=1}^{d} w_i \phi_i(\mathbf{x}) + w_0\right) = \text{sgn}\left(\mathbf{w}^T \Phi(\mathbf{x}) + w_0\right) \quad . \end{aligned}$$

The first stage of RBF network training determines the number, location and range of the approximating functions. Several methods are available for the initial placement of basis functions, from the location of basis functions around every training data example to the use of clustering algorithms (see p. 64 onwards) to place basis functions about groups of similar examples. The latter approach is more efficient than the former, especially on large data sets. The second stage of training optimises the approximation weights, which regulate basis function contributions to the decision function. This may be performed by gradient descent on the sum of squares loss function over the training data (cf. § 2.2.2 and ANNs above). The training structure of an RBF network may be viewed as a two-layer ANN, with Gaussians as hidden layer nodes connected to the output by weights. RBF network training is faster than training a two-layer ANN, however, as RBF network layers are trained separately. Another advantage is that the RBF 'hidden layer nodes' represent explicit mappings of input space, therefore, RBF networks lend themselves more favourably to probabilistic output than do ANNs, the hidden nodes of which tend to encode the input space rather than appear within it. Work by Walczak and Massart [1996] provides an example of the application of RBF networks to molecular structure-activity analysis and RBF network performance on QSAR and SPC analysis problems is assessed during the comparisons of Burbidge et al. [2001] and Trotter and Holden [2003].

Despite the popularity of neural network architectures, several other supervised machine learning algorithms have been applied widely to problems of drug discovery. **Genetic algorithms** (GAs) are a powerful stochastic technique for optimisation and regression [Goldberg, 1989]. Based on the principles of Darwinian evolution and genetics, GAs start with a population of *hypotheses* (a random subset of hypothesis space), which are allowed to 'breed' over several generations in order to evolve a solution. For example, the initial population could be a collection of linear classifiers with randomly drawn weight vectors. Hypothesis parameters are represented by *chromosomes*, to which genetic operators are ap-

plied to diversify the population. *Mutation* operators select a parent hypothesis at random and mutate it via a random change to the chromosomes. *Crossover* operators interchange selected chromosomes from two parent hypotheses, to produce offspring for the next generation. At each training iteration, the population are tested according to a *fitness* criterion, for example, a measure of empirical error across the training data. Those below a fitness threshold are rejected from the population under the assumption that the underlying causes of their weak performance should not be passed to the next generation. Those above the fitness threshold are acted upon by the genetic operators in order to produce the next generation.

GAs provide a particularly good random search optimisation technique and have produced good results when applied to drug discovery, especially when treating the traditional QSAR or 'drug-likeness' scenario during lead generation [Gillet et al., 1998; Shi et al., 1998]. Design and application of the technique is not without associated challenges that affect generalisation ability and the size of discrimination task involved. For example, the time taken to evolve an optimum solution may be much longer than for other learning techniques. The stochastic nature of GA population initialisation and genetic operators leaves output dependent upon choices made during implementation of the algorithm, which require careful consideration prior to training. GAs may converge prematurely due to the early appearance of a particularly strong hypothesis (and its offspring), which dominates the population and produces a situation analogous to neural network convergence to local minima.

Several modifications to the basic algorithm solve many impediments to their successful application. These range from different evolutionary strategies to the co-evolution of separate algorithms, which compete against each other. Comprehensive introductory texts on GAs and their development as a machine learning tool are provided by [Goldberg, 1989] and [Mitchell, 1996]. The practical application of GAs to structure-property data is the subject of a texts by Hibbert [1993] and Devillers [1996a] and a large body of work regarding the application of GAs to drug design is available online at *http://panizzi.shef.ac.uk/cisrg/links/ea_bib.html*.

**Decision trees** provide informative classification via the induction and combination of predictive rules for each data attribute. In the simplest case of discrete data and target attributes, training example attributes are examined and ranked according to their ability to partition the data examples according to the target attribute. The most common ranking measure employs the *entropy* of labelled data to measure the relative size of target attribute groupings (impurity) within the data. If an equal number of examples belong to each class, the entropy is maximal. If all examples belong to a single class, the entropy is zero. The *information gain* of a single data attribute is the reduction in entropy expected were the training data partitioned by a threshold on that attribute alone [Mitchell, 1997].

A popular method of decision tree construction employs *recursive partitioning*. Attributes are ranked as described above and the most informative is selected. Thresholds available on the most informative attribute are used to partition the training data. On each

partition of the training data, the selection process is repeated for all remaining attributes. The selection and partition steps are iterated until an attribute with zero entropy is reached (or there are no more attributes to consider). An attribute with zero entropy associates a single class label with all examples that reach it. To treat continuous attribute values, thresholds are selected in order to maximise the information gain of an attribute. A lower limit may be placed on the number of thresholds employed for each attribute, e.g. each attribute must possess at least two partitions. A detailed description of the above is provided by [Mitchell, 1997].

The attributes selected at each partitioning step may be represented as 'nodes' and the progression between nodes (along 'branches') represents the training data subset chosen by partition. Thus, the attributes and their partitions represent a tree structure in which the most informative attribute is chosen as the root node at the top of the tree, analogous to a tree trunk. The branches represent partitions of the training data according to thresholds on the preceding node. At the end of each branch a further node represents the next most informative attribute. The tree continues downwards and terminates at 'leaf' nodes, which label examples with values of the target attribute. The hypothesis space represented by a decision tree trained on discrete data is particularly expressive and is guaranteed to contain a number of classifiers in version space. The training data may be partitioned according to the target attribute by several possible tree sub-structures (hypotheses).

The decision tree, as described above, is designed so that its leaves replicate the target attribute as closely as possible. As described earlier, the pursuit of minimum empirical error leaves a learning algorithm open to overfitting the training data and, subsequently, poor generalisation performance. To counter this, the tree may be *pruned* via the systematic removal of attribute nodes and the branches below them until the empirical performance of the tree suffers as a result. Pruning a tree that fits the training data well without loss of empirical performance increases the likelihood that the pruned tree will generalise well on further examples drawn from input space (cf. the principle of Ockham's razor, described well in [Russell and Norvig, 2003]). Other methods to improve tree performance include various information gain measures and cost functions [Mitchell, 1997], which weight certain attributes so that they appear higher up the tree.

Quinlan's commercially available C5.0 tree algorithm is widely used for pharmaceutical classification and it is based on his earlier C4.5 and ID3 algorithms [Quinlan, 1986]. A benefit of decision trees is that they can be used to elaborate upon decisions made by more powerful and less informative techniques, such as ANNs (cf. hybrid techniques, p. 2.2.6). Decision trees have gained favour in pharmaceutical lead optimisation due to the user-friendly, rule-based nature of their predictions. Nevertheless, it is common that the rules produced by trees contain as much, if not more, information about which properties a molecule should not possess as they do about which properties it should possess in order to fulfill a given selection criterion. This is an open problem, the solution of which would be of great benefit to the molecular analysis community and which, although not touched upon further during this thesis, would make an interesting subject for future work. There exist

powerful decision tree methods capable of competing with both ANNs and GAs in terms of predictive accuracy, but they commonly require a great deal of tuning to do so [Murthy et al., 1998]. Simpler techniques are available and popular but can sometimes struggle to attain similar performance levels. This will become apparent during later stages of this work. A good example of the application of decision trees to molecular classification is provided by Hawkins et al. [1997] and the C5.0 algorithm is employed for the purposes of algorithmic comparison on QSAR and SPC data in [Burbidge et al., 2001; Trotter and Holden, 2003] and, similarly, Chapter 4 of this work.

*Lazy* learning algorithms only consult the training data upon introduction to an unknown example. Lazy algorithms are different to the *eager* algorithms described earlier, as they use the location and neighbourhood surrounding an unknown example as a basis for their classification, instead of discriminating over the training data before classifying a set of unlabelled data. For example, **Nearest-neighbour classifiers** ($k$-NNs) are one of the most popular, and least complex, supervised classification techniques [Cover and Hart, 1967]. The $k$-NN algorithm examines the class labels of a pre-specified number ($k$) of training examples that are nearest to an unknown example. The unknown example is assigned the class label associated with the majority of the $k$ nearest examples. The curse of dimensionality is partially avoided via the assessment of similarity between test and training examples, as for kernel transformations on example pairs (§ 2.2.3). The higher the dimension, however, the more calculations are necessary in order to find the nearest neighbours. Despite their relative simplicity, $k$-NN classifiers often provide class-leading predictive ability [Trotter and Holden, 2003] and are frequently used to provide a benchmark when evaluating new machine learning methods (cf. Chapter 4). Other lazy learning algorithms, such as non-linear density estimators, are increasingly employed to QSAR analysis for lead generation in large compound collections [Wilton et al., 2006] and are discussed further in Chapter 6.

**Hybrid techniques** represent the simultaneous use of more than one technique (a hybrid system). Hybrid systems are used in drug discovery to solve problems that range from data visualisation to classification and clustering. The most popular hybrid framework employed is the use of one technique to perform a principal function of another (*function-replacing hybrids* [Goonatilake and Khebbal, 1995]). This may be described as technique enhancement, whereby the weakness of one technique is replaced by the strength of another. For example, a GA may be employed to optimise the weights of a backpropagation ANN architecture. In doing so, performance is improved and the system converges in far fewer cycles, reducing the training time of the algorithm (hybrid GA / ANN architecture [Manallack and Livingstone, 1999]). In general, accurate techniques that are slow to train or difficult to optimise may be made faster to operate without consequent loss of accuracy [Bennett and Blue, 1997]. Hybrids have also been used to provide more information regarding the output of 'black box' techniques, e.g. ANNs. Goonatilake and Khebbal [1995] provide a comprehensive review of 'Intelligent Hybrid Systems' that includes examples of many different kinds of hybrid system, applied to a variety of problems. Some examples of their use for drug discovery may be found in [Gini et al., 1998; Li et al., 1999; Manallack and Livingstone,

1999] and [Langdon et al., 2001].

The supervised machine learning methods described above are arguably the most widely applied to the drug discovery process, but formulation of the relationship between molecular structure and biological properties of interest as a supervised machine learning problem has lead to the application of a much wider range of learning methods. For example, inductive-logic programming (ILP) [King et al., 1992; Hirst et al., 1994; Bryant et al., 1997], fuzzy logic [Russo et al., 1998], expert systems [Gini et al., 1998] and probabilistic approaches [Labute, 1998] have all been applied to various stages of the discovery process with various degrees of success.

A selection of unsupervised learning methods, used in order to identify trends in molecular similarity during combinatorial library design, are given brief descriptions in the same manner as for supervised learning methods above. *In silico* drug discovery produces a large amount of highly descriptive data, the complexity of which may require reduction to facilitate handling and visualisation. Accordingly, unsupervised learning techniques are employed widely throughout the discovery process primarily for such purposes. **Clustering techniques** are used to find patterns in sets of unlabelled data. Unlike supervised learning techniques, the data examples do not have associated target attribute values, thus prompting a search for groups of similar points in order to categorise the data. Clustering techniques are used to explore chemical space to identify regions of interest in large collections of unknown, or unlabelled, compounds. Information regarding the structure of chemical space can be used to narrow a search via the identification of suitable starting points and to design training sets for the supervised techniques listed above. The majority of clustering algorithms seek to minimise intra-group variance and maximise inter-group variance in order to find the clearest groups (clusters) in the set. Examples that do not fall easily into any cluster are called *outliers* or *singletons* [Butina, 1999].

Outlier detection techniques are used to remove combinatorial library members that possess similar binding affinity and / or biological property values, to avoid redundancy and allow a library to cover its chemical space more efficiently. The most diverse compounds in a library are weeded out and investigated further. If the structural features that cause outliers to be distinct from the other library members lead to unsuitable biological properties for the therapeutic aim, they are removed. If they possess suitable biological properties as well as possessing distinct structural characteristics, they are given special treatment, as they may describe a new area of chemical space that contains molecular combinations that can fulfill the therapeutic aim. Further combinatorial synthesis around an interesting outlier furthers the chance of discovering a novel therapeutic product [Butina, 1999].

*k-means* clustering is a simple partitioning technique that requires some knowledge of the data prior to its application. Cluster centroids are spread uniformly across the data, according to an initial estimate, $k$, (informed or otherwise) of the number of clusters within the data. Data examples are assigned to the closest cluster centroid (according to a chosen similarity metric) and centroids are moved to the mean point of their members. The process

is iterated until the centroids cease to move beyond a threshold limit. Thus, clusters are formed via the minimisation of intra-cluster variance and the maximisation of inter-cluster variance [Mitchell, 1997].

*Agglomerative* clustering is a hierarchical clustering technique. Initially, each data example present represents a cluster. Each cluster is iteratively expanded in a uniform manner. When two, or more, clusters make contact, they become a single, larger cluster containing all of the examples from the previous ones, and so on until all of the data examples are contained within one cluster. Each cluster has a *lifetime*, during which, the amount of points it contains does not change. When cluster lifetimes are examined, those that survive longest are deemed most stable. *Divisive* clustering works by the same principle, but starts with one cluster that contains the whole data set and shrinks it until every point represents a cluster. This method is well suited to the display of results, because the agglomeration (or division) process may be represented by a tree structure. The single cluster that encompasses the whole data set represents the trunk and, as that cluster is subsequently divided into smaller clusters, their appearance represents the tree separating into branches. The longer the branch, the more stable the cluster [Eisen et al., 1998]. An application of cluster trees to QSAR data is provided by [Santos Magalhães et al., 1999].

*Jarvis-Patrick* clustering [Jarvis and Patrick, 1973] is a non-parametric method, which is particularly suitable for examination of large, high-dimensional data sets. The method is aimed at solving clustering problems encountered when the examples are not grouped into easily separable 'globules'. Data examples are grouped according to the number of shared nearest neighbours, which requires selection of the number of nearest-neighbours to examine and the number that must be in common for two molecules to inhabit the same cluster. The revised method of Butina [1999] only requires one parameter selection. A primary cluster centroid is identified as the molecule with the largest number of neighbours within some similarity threshold. The centroid and its neighbours are removed from the data and the process iterated, selecting centroids and their neighbours, until the remaining molecules have no neighbours within the similarity threshold, i.e. they are singletons. Clustering is performed by placing an exclusion sphere of radius equal to the similarity threshold about each centroid, so as to contain the centroid neighbours and identify any new molecules that fall within a particular cluster. A brief review of several clustering methods, including hierarchical and non-parametric, employed in the design of combinatorial libraries is provided by [Drewry and Young, 1999].

Unsupervised learning is employed also for the visualisation of high-dimensional descriptions of molecular structure. Humans have formidable pattern recognition ability on data displayed in one, two or three dimensions. Data of higher dimension presents a problem, as we are unable to perceive it. Statistical transformations, such as principal component analysis, and unsupervised machine learning techniques, such as Kohonen neural networks, have been successfully employed in drug discovery to visualise multivariate data and to focus the information presented to supervised learning techniques via the extraction of information and reduction of redundancy in large data sets (cf. § 2.2.4) [Xu and Hagler,

2002].

**Neural networks** may be employed to visualise multi-variate data. *Kohonen networks* consist of a two-dimensional grid of nodes (e.g. the sigmoidal nodes described earlier), connected to each attribute of a data set. Upon encountering a new example, the node with weight vector most similar to the example has weights updated to approach the example. A neighbourhood function updates the weights of surrounding nodes in a manner inversely proportional to their distance from the most similar node. The range of the neighbourhood and the learning rate of the update procedure are reduced gradually during the learning process. Thus, the algorithm makes large initial changes to the weight vectors of grid nodes and proceeds to tune the changes made. Learning ceases when the learning rate approaches zero. Trained thus, the two-dimensional grid of nodes displays a 2D topology of the original data set, via similarities between node weight vectors. The Kohonen network is used widely for the elucidation of structure-property data and examples of its successful use are provided in a text by Devillers [1996b].

*Auto-associative neural networks* employ a feed-forward, multi-layer perceptron architecture to map input data to a lower-dimensional space. For example, a reduction of data with dimensionality $m$ to a lower dimensionality $d < m$, may be performed by a network with $m$ input layer nodes that feed forward into $d$ hidden layer nodes. The hidden layer nodes, in turn, feed forward to $m$ output layer nodes. The targets of the output layer nodes are the corresponding $m$ inputs. Thus, the network is trained to replicate the input. Once trained, the $d$ hidden layer outputs provide a $d$-dimensional representation of the original data. When presented with new input data, the hidden layer nodes map the new data to $d$-dimensional space, thereby reducing dimensionality [Bishop, 1995a; Schwenk, 1998].

**Principal Components Analysis** (PCA) [Jolliffe, 1986] maps attribute vectors to a set of orthogonal axes, formed of a weighted linear combination of the original attributes. Inter-attribute correlation and, hence, redundancy is removed from the new, orthogonal space to which the data are mapped. The original data are reduced by mapping to a smaller set of orthogonal axes, for example, to alleviate the negative effects of data redundancy and the curse of dimensionality when attempting to associate a target attribute with the data. The notion of mapping data to a lower dimensional space contrasts previous descriptions of high-dimensional mappings in order to facilitate the creation of non-linear relationships between data and a target attribute.

Dimensionality reduction requires the removal of a subset of the new axes, preferably involving as little information loss as possible. Work by Jolliffe [1986], described clearly in [Bishop, 1995a], shows that an orthogonal transformation of a body of data that accounts for the variance inherent within it, but not necessarily displayed by the original axes, permits the new axes to be ranked in order of the amount of variance that they account for. This is embodied by a convenient eigen-transformation of the original data, $X$, in the form of its covariance matrix,

$$\Sigma_X = \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T$$

where $\bar{\mathbf{x}}$ is the mean vector of the collection of attribute vectors, $X$. The eigen-solution,

$$\Sigma_X \mathbf{v}_i = \lambda_i \mathbf{v}_i$$

provides the orthogonal axes, in the form of the eigen-vectors $\mathbf{v}_i$, and the means to order them, in the eigen-values $\lambda_i$. The summed eigen-values represent the amount of variance accounted for by the transformation. Removal of eigen-vectors corresponding to the lowest eigen-values reduces dimensionality with least information loss. The heuristic methods of Kaiser [1960] and Catell [1966] provide guides to how many eigen-vectors to remove whilst retaining a sensible amount of the original variance. The scree test is employed later in this work in Chapter 4. An additional benefit of this approach is that the two (or three) eigen-vectors with highest eigenvalues may be employed to provide a low dimensional visualisation of the data. This approach is particularly popular when examining high-dimensional data, such as the pharmaceutical data encountered here, and is available in many commercial data analysis packages [Xu and Hagler, 2002], e.g. *DecisionSite* software [Spotfire Inc., 2005]. Relevant developments of the PCA transformation outlined above include its treatment by kernel methods, in order to provide a set of orthogonal axes that relate to the original data in a non-linear fashion [Schölkopf et al., 1999].

*Multi-dimensional Scaling* (MDS) [Kruskal, 1964] performs an eigen-transformation of an original data set, $X$, in a manner similar to PCA, but replaces the co-variance matrix with a matrix containing measures of inter-example dissimilarity. The dissimilarities may be described by Euclidean distance (or some other Minkowski distance) between examples, but may be described also by any valid measure of dissimilarity, e.g. the Pearson correlation coefficient subtracted from one yields a dissimilarity in the range [0,2]. A transformation performed on a measure of inter-example distance yields a new set of axes which account for the dissimilarities between examples in the original input space. Eigen-values associated with the transforming eigen-vectors may be used to order the eigen-vectors in order of how well they reflect the inter-example relationships in the original space, which allows the data to be reduced for visualisation or greater representational efficiency [Wang et al., 2004].

The *Mahalanobis Distance* is related to PCA data transformation. The distance of each example in a body of data from the estimated location (or centre) of that body is described by

$$D_M = (\mathbf{x}_i - \bar{\mathbf{x}}) \Sigma_X^{-1} (\mathbf{x}_i - \bar{\mathbf{x}})^T \ ,$$

where the mean vector of data set $X$ is employed to describe the location of the data and the covariance matrix of $X$ is employed to describe the shape of the data. The Mahalanobis distance is employed to identify outlying compounds that possess different structural properties to the other members of a particular library or collection. Outlier detection is facilitated by the knowledge that the Mahalanobis distance of points within a multivariate normal distribution correspond to an F-distribution with degrees of freedom determined by the dimensions of the set $X$ ($m$ and $n - m$). The location and shape parameters as described

above are not robust to the presence of outliers in the data, however. Accordingly, several robust Mahalanobis distance measures exist with which to identify outliers using a statistical threshold, e.g. [Franklin and Brodeur, 1997; Hardin and Rocke, 1999]. The method of Filzmoser et al. [2005], developed to identify outliers in geological data, provides an automated outlier detection threshold on a robust Mahalanobis distance measure.

The unsupervised techniques described above represent a selection of the more popular data analysis methods employed within the drug discovery process. Several sources provide more detailed descriptions, including Hand [1981]; Martens and Næs [1989]; Bishop [1995a]; Mitchell [1997]; Duda et al. [2000] and Russell and Norvig [2003]. Further work describing general applications of machine learning methods to the drug discovery process may be found in [van de Waterbeemd, 1995; Böhm and Schneider, 2000; Matter et al., 2001; Xu and Hagler, 2002; Schneider and Downs, 2003] and [Böcker et al., 2004].

### 2.2.7 Conclusion

As the discovery search is widened by *in silico* HTS and the combinatorial methods that accompany it, the amount of data that requires classification is growing many times faster than the amount of data that is available on which to build predictive models. Data that requires classification is filtered down from the top of the process, which may start with the virtual consideration of $10^{12}$ potential molecular combinations. Data available on which to build SPC models is taken from the opposite end of the process, where a lot is known about a small collection of optimised compounds. The ability of an algorithm to create predictors that cope with the classification of large data sets, both in terms of predictive generalisation and the time taken to make the predictions, is if increasing importance.

This chapter introduces SPC analysis as a vital part of contemporary drug discovery, which presents a range of familiar impediments to the generalisation required of classifiers created to perform this task *in silico*. Support vector machines are introduced as a relatively recent addition to the collection of supervised machine learning methods, from which SPC relationships are created, that have been designed and adapted to overcome many of the challenges provided by the creation of SPC relationships. Such challenges include:

- a paucity of available training examples, requiring that a model must be able to generalise well beyond the range of data from which it is created;
- the fact that there is no 'perfect' descriptor subset, with which to relate molecular structure successfully to biological properties under consideration;
- the representation of data examples by a large number of descriptive attributes. It is often the case that there are fewer examples in the available training data than there are dimensions of the chemical space that they inhabit;
- erroneous, or noisy, measurements within the data available from which to build predictors of a target property;

- the potential for complex, non-linear relationships between molecular structure and target properties; and
- there often existing a significant difference in the number of examples available to represent each class of data, due to the nature of target properties under examination.

It is promising that the SVM algorithm is robust to noisy, high-dimensional training data, small collections of which are required to generalise to large amounts of unlabelled data. That the SVM algorithm is adaptable to overcome domain-specific challenges, such as class imbalance and strong local effects within global example representations, via the incorporation of domain-relevant information shows even greater promise for the successful application of SVMs to predict biological properties from aspects of molecular structure.

The following chapter describes a collection of SPC ADMET data, provided by GlaxoSmithKline for this research, and outlines an experimental practice for the comparison of supervised machine learning algorithms upon such data and the assessment of their success or otherwise.

# Chapter 3

# ADMET Data & Experimental Practice

## 3.1 GlaxoSmithKline Data

Five ADMET data sets were made available by the ADMET Modelling Group at GlaxoSmithKline, Stevenage, UK, for this work. Each data set describes a binary separation problem, the nature of which is described below. The first two dimensions of a multi-dimensional scaling (MDS; § 2.2.6, p. 67) transformation of each problem are plotted alongside each description, to provide an impression of how the data are arranged in chemical space. One should note that the MDS transformation provides a low-dimensional representation of compound dissimilarity in chemical space, *not* a representation of their separability according to the target attribute. Review papers by Matter et al. [2001] and van de Waterbeemd [2003] provide generic information regarding the ADMET classification tasks described below.

### 3.1.1 Blood-Brain Barrier

Compounds designed to interact with target sites in the brain must bypass the protective membrane between blood and brain. To avoid side-effects, compounds designed to affect other parts of the body must be repelled. Models of blood-brain barrier (BBB) penetration are employed in consideration of the desired effects of novel compounds. A good example is provided by the production of sedating and non-sedating anti-histamines [Atkinson et al., 2002; Ecker and Noe, 2004]. In this data set, compounds that belong to the positive class are those that cross the BBB. Compounds that belong to the negative class do not.

The blood-brain barrier data set is the result of *in vivo* studies and contains 476 (337 +ve / 139 -ve) examples, represented by 72 VolSurf descriptors. The VolSurf molecular representation [Goodford, 1995; Cruciani et al., 2000] provides 3-D physico-chemical information in a 2-D string of real-valued descriptive attributes (cf. § 2.1.5). The MDS plot of figure 3.1, overleaf, displays the negative class interspersed throughout a larger, denser, positive class. A cluster of outlying positive data, towards the top of figure 3.1, may further complicate this separation problem. For the purposes of visualisation, 'positive' examples

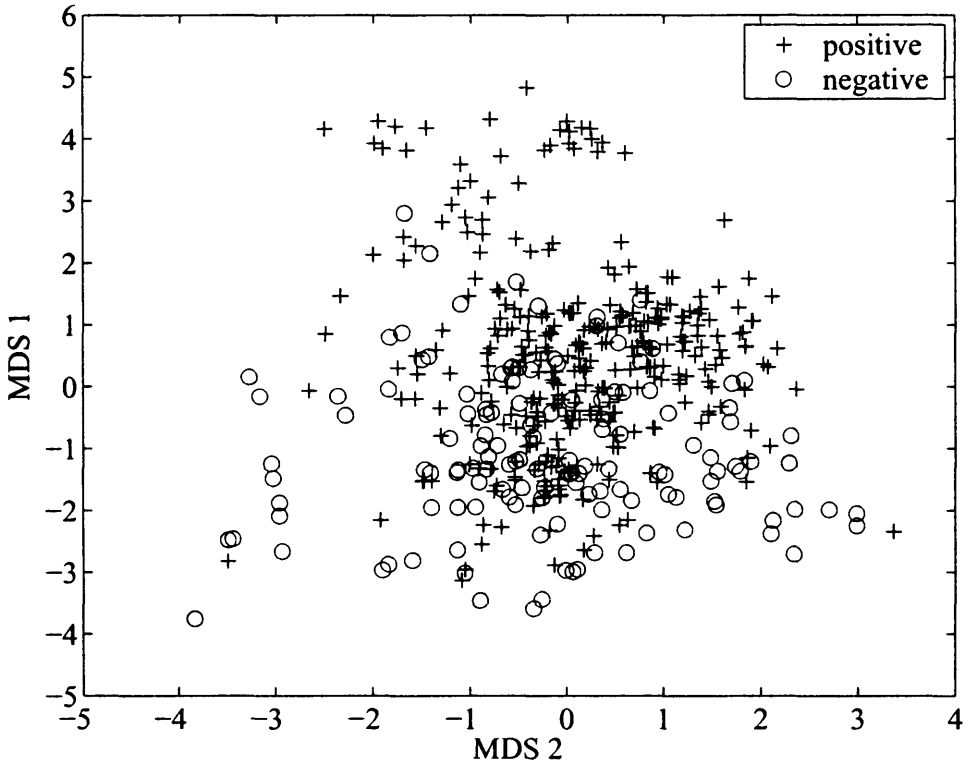


**Figure 3.1:** MDS Plot of BBB Data

represent compounds that exhibit high BBB penetration and 'negative' examples exhibit low penetration.

### 3.1.2 P-glycoprotein Substrate Binding

Membranes, such as the blood-brain barrier described above, contain P-glycoprotein (P-gp). Many compounds have a tendency to bind to P-gp (they are substrates), therefore they will be unable to cross the membrane if required to do so [Atkinson et al., 2002; Ecker and Noe, 2004]. A tendency to bind to P-gp may be seen as a benefit should a compound be required not to cross the membrane, but the P-gp in the membrane may become saturated if too much binding occurs, rendering it useless as a repellant thereafter. Models used to discriminate between P-gp substrates and non-substrates are built to reject substrates.

The P-gp data set contains 138 (59 +ve / 79 -ve) examples, represented by 5 Abraham molecular descriptors [Zhao et al., 2003], and is the smallest problem of the five detailed in this section. Examples described as 'positive' exhibit low propensity to bind to P-gp. Figure 3.2, , overleaf, shows the majority of the positive and negative data lying apart in the MDS transformed chemical space. Several outliers are visible at the extremities of the plot, which may lead robust global, rather than local, machine learning solutions to perform best on this data.

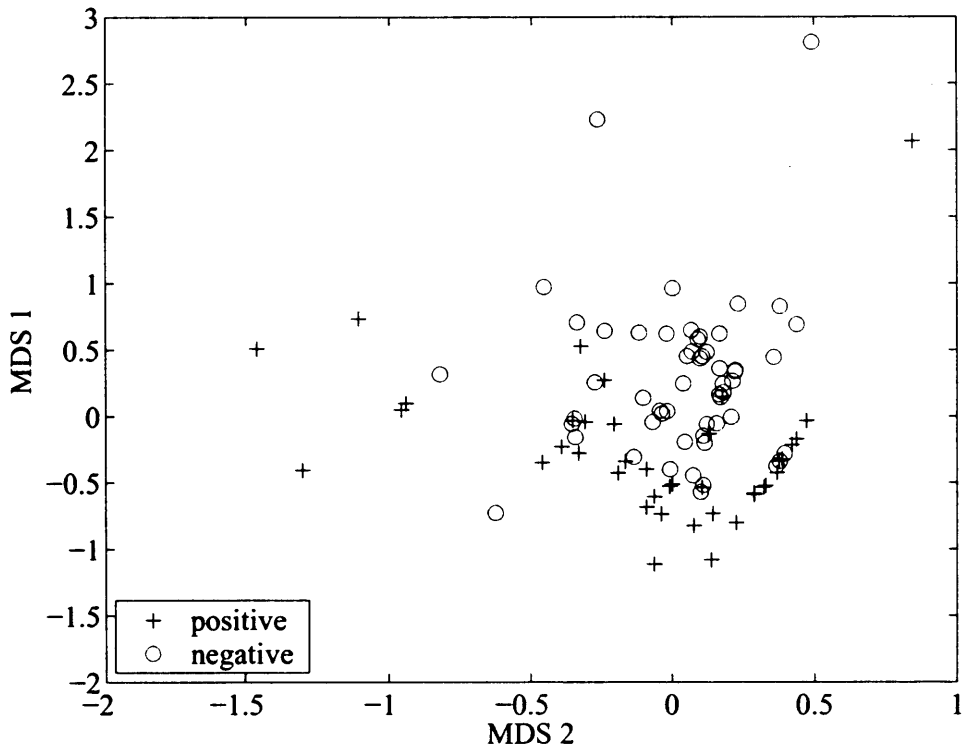


**Figure 3.2:** MDS Plot of P-gp Data

### 3.1.3 Acute Toxicity

Toxic compounds are inappropriate for therapeutic use. *In silico* models of toxicity are particularly useful, because compounds are rarely tested for toxicity *in vivo*. Training data for the computer modelling of discriminant toxicity relationships are likely to be provided by *in vitro* screening of well described, unlabelled libraries. Accordingly, the target attribute is more likely to contain noise than the structural attributes.

The Acute Toxicity data contains 1176 (226 +ve / 950 -ve) examples, represented by 72 VolSurf descriptors. Here, 'positive' examples exhibit high toxicity. Figure 3.3, overleaf, reveals that the positive class is greater in both number and density of examples compared to the minority, negative class. Such an imbalance may create difficulties when attempting to achieve good generalisation accuracy on the negative class, as the majority of supervised machine learning methods can be lead to overfit a dense majority class to the detriment of a sparser minority class (cf. § 2.2.4). In addition, a number of outliers belonging to both classes are visible towards the left of the plot.

### 3.1.4 Bioavailability

The aim of modelling bioavailability is to identify compounds that remain in the human system long enough and in sufficient amount to have a therapeutic effect. Compounds belonging to the positive class of this problem are those with bioavailability above an acceptable threshold level. Compounds belonging to the negative class are not sufficiently bioavailable

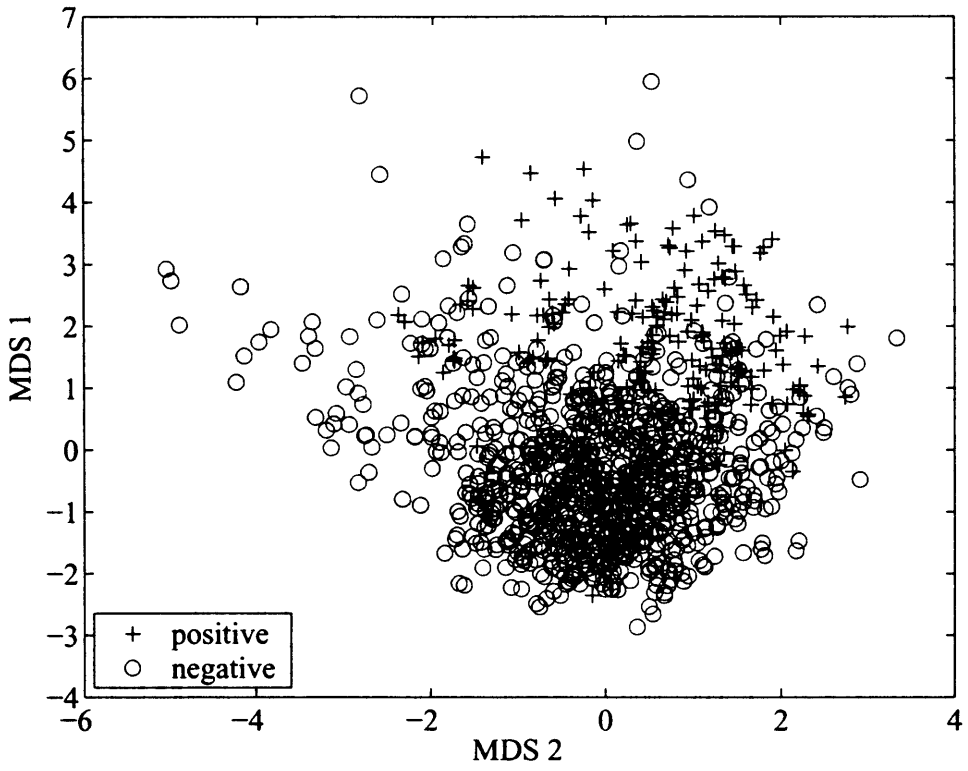


**Figure 3.3:** MDS Plot of Acute Toxicity Data

and would not be considered for further development. Bioavailability is affected by many factors, including solubility, permeability, and the residual concentration after a 'first pass' through the metabolism. Further details are available from pharmacokinetics texts, such as [Roland and Tozer, 1995]. The bioavailability data set is the result of *in vivo* studies and contains 481 (393 +ve / 88 -ve) examples, represented by 68 real-valued physico-chemical attributes calculated from those of the combined molecular fragments that comprise each compound [Matter et al., 2001] and a number of whole molecular properties. Positive examples represent compounds that display high bioavailability. This data set (and the protein binding data described below) is included with the specific intent of testing the supervised machine learning methods used during this work in challenging, real-world conditions. Figure 3.4, overleaf, suggests that the data lie in two clusters, both containing a mixture of data from both classes. The clusters are distinct, but are not well defined, and a number of outlying points are visible across the transformed chemical space. It is difficult to predict which type of machine learning solution (for example linear / non-linear or global / local) will perform best on what appears to be a non-uniform sample.

### 3.1.5 Protein Binding

Protein binding can negatively affect pharmaceutical efficacy. If a compound is prone to bind with proteins in the blood, it is likely that it will not be effective upon arrival at the target. In this case, binding is particularly undesirable and models of protein binding are used

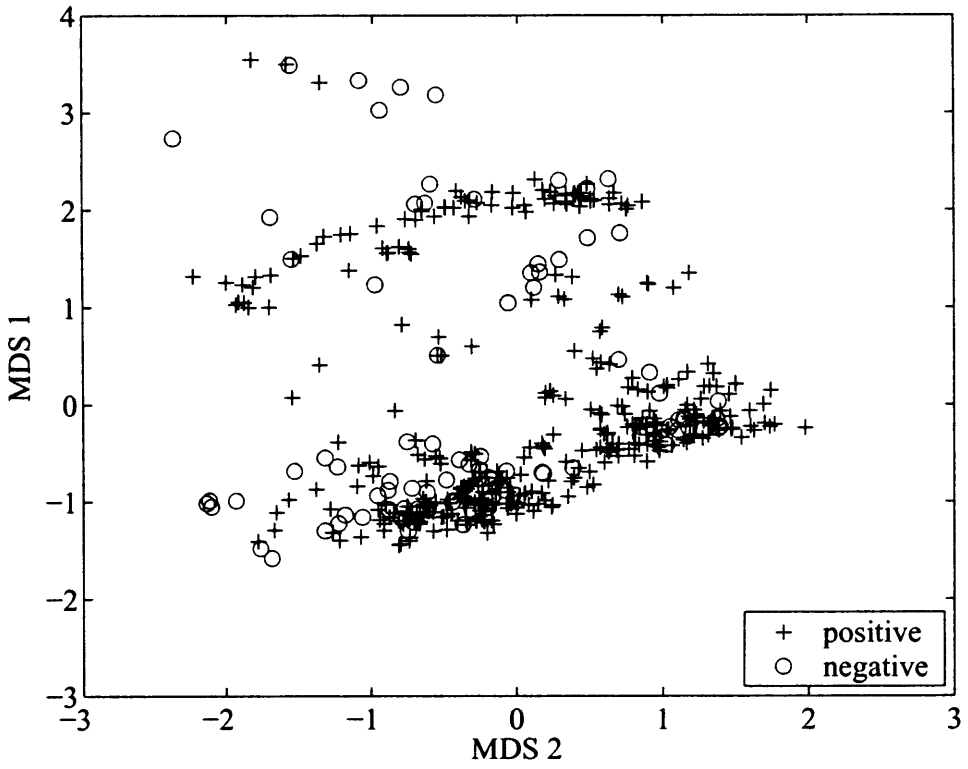


**Figure 3.4:** MDS Plot of Bioavailability Data

to reject compounds that have a protein binding tendency above a predetermined threshold (negative class). The protein binding data set contains 459 (297 +ve / 162 -ve) examples, represented by 16 real-valued physico-chemical attributes calculated from those of the combined sub-molecular fragments that comprise each compound [Matter et al., 2001]. Despite representing a different separation problem with different descriptive attributes, the protein binding data appear similarly distributed in figure 3.5, overleaf, to the bioavailability data described above and in figure 3.4, above. The data appear in two distinct clusters in the MDS-transformed chemical space, but appear to be less mixed than the bioavailability data. There also appear to be fewer inter-cluster outliers.

## 3.2 Experimental Practice

It is reasonable to assess the potential worth of a new technique to an existing application by comparing it to other techniques that represent a state-of-the-art, possibly alongside a traditional benchmark. It is challenging to assess the ability of a new technique for drug discovery in a standard manner. The ability of a classifier, or modelling technique, must be measured against the specific task within the process that it will be used for, whether it performs that task satisfactorily and, moreover, whether it outperforms the techniques currently employed to perform the task. The following practice is suggested in order to provide a principled comparison of several supervised machine learning algorithms on the ADMET data described above.

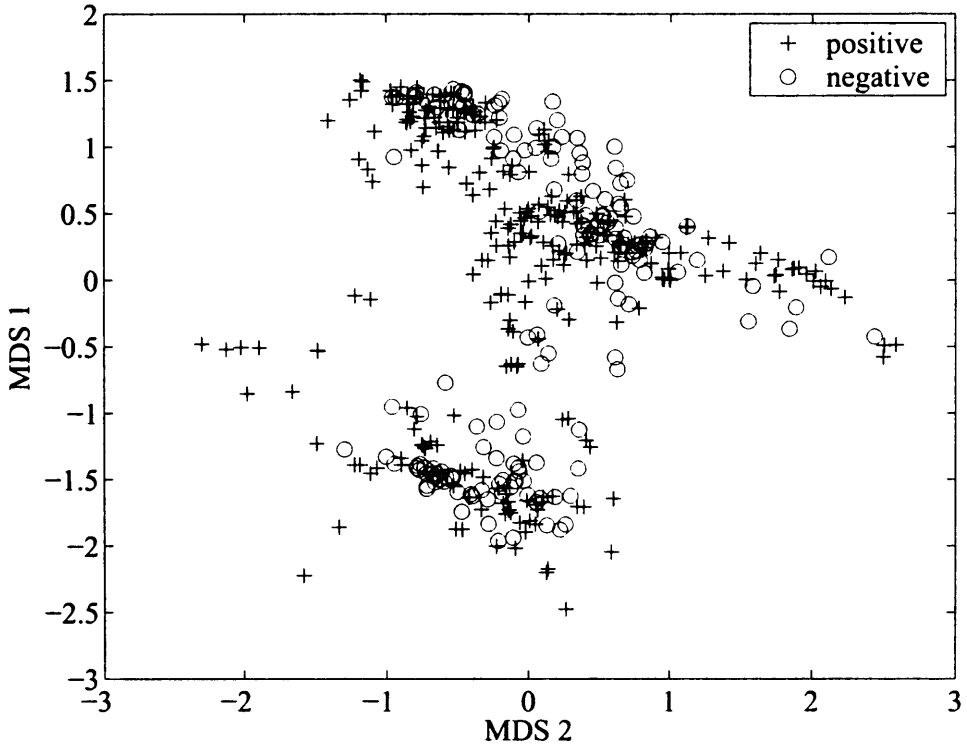


**Figure 3.5:** MDS Plot of Protein Binding Data

### 3.2.1 Data Partitioning

Cross-validation [Kohavi, 1995; Molinaro et al., 2005] (cf. § 2.2.4) is commonly employed to provide a comparative measure of algorithmic performance on a set of labelled data, especially when a limited amount of labelled data is available. If sufficient labelled data exist, however, a single partitioning of the data may be used to provide an impression of how a classifier generalises to a significant amount of unseen data, as would be the case in a real-world scenario. A set of labelled data are partitioned into two subsets (preferably of equal size). All training, parameter setting and performance estimation is performed on one subset (the training set) and a final performance measure is gained from classifying the second set (the test set) with a classifier trained on the first. It is common to partition the data in a random manner for this purpose, but other principled methods exist.

The Kennard and Stone (K&S) partitioning algorithm [Kennard and Stone, 1969] produces two separate subsets of the available data that reflect its distribution in approximately the same manner. The data are split into two groups, with each group designed to contain a similar distribution of examples both to each other and to the original data set that they comprise. This method is useful when the internal structure of the data may not be uniform, or distributed in a standard manner. For example, if the data are not spread uniformly across the input space, but rather they are grouped in a number of discrete clusters, it is conceivable that a randomly sampled subset of the data will not fully represent each cluster - although this depends upon the amount of data and size of the sampled subset. K&S partitioning avoids this situation by employing a selection strategy that aims to sample data

evenly across the 'inhabited' area of input space. Thus, if the data lie in a number of discrete clusters, each cluster should be proportionally represented in both the training and test partitions. An additional advantage of this is that, in the majority of cases, examples of the same class tend to occupy similar regions of chemical space (if there was no relationship between data class and location in input space, no useful model could be drawn from the data). Because K&S partitioning samples data evenly from all localities of the original input space, it is likely that class proportions will be preserved, or at least approximated, in the presence of class imbalance (cf. § 2.2.4). Table 4.1 in Chapter 4 demonstrates how this occurs in practice.

### 3.2.2 Accuracy Measures for Imbalanced Class Populations

The data sets used in this comparison, along with the method employed to partition them, are described in section 3.1 and the preceding subsection respectively. It is clear that all of the sets contain unequal class populations, with the proportion of positive and negative data equal to 71% / 29% (BBB), 43% / 57% (P-gp), 19% / 81% (Tox), 82% / 18% (Bio), and 65% / 35% (PrB). For example, if one were to evaluate algorithmic performance on the bioavailability data solely by comparing predicted class labels to true class labels, a solution that classifies every example in the set as positive would record 82% (0.82) overall accuracy. Such a solution would not perform the task required of machine learning in this circumstance because, as described in Chapter 2, a successful solution must reject examples that are not sufficiently bioavailable as well as retain those that are.

There exist performance measures that account for class imbalance when providing an assessment of algorithmic performance on binary classification tasks. Sensitivity, specificity and receiver operating characteristic (ROC) curves [Scott et al., 1998] provide such measures. A measure that combines the descriptive attributes of both sensitivity and specificity in a single performance estimate would be a good method with which to evaluate algorithmic performance in the comparison required by this work.

A single figure that provides a measure of accuracy that reflects performance on both classes may be obtained by weighting the classifications of both classes so that both contribute 50% towards a balanced accuracy. For example, if Class $A$ contains 80 examples and Class $B$ contains 20, then predictions of Class $A$ members should contribute (50/80 = 0.625) when summed to provide a performance measure. Predictions of Class $B$ members should contribute (50/20 = 2.4) when summed to provide a performance measure. Thus, if all 100 examples are classified as Class $A$, the balanced performance records 0.5 (or 50%). The same figure is achieved if all 100 examples are classified as Class $B$. If all examples are classified correctly, the balanced performance records 1.0 (or 100%) as expected. A balanced measure thus calculated is equivalent to averaging the sensitivity and specificity of the prediction and ensures that the situation described above, wherein a classifier concentrates solely on the majority class, is not rewarded with a seemingly acceptable measure of performance. Similar balanced accuracy measures are employed in [Matthews, 1975; Lodhi et al., 2000] and [Lee et al., 2001].

### 3.2.3 Parameter Selection

The majority of machine learning techniques require the selection of one or more free parameters to set an inductive bias [Kohavi, 1995] that controls the structure of the solution that they create from the training data. Free parameters are selected so as to maximise an estimate of the generalisation error of the classifier created. If free parameters are set solely according to performance on the training data, it is likely that the parameter values chosen would lead to a model which overfits any noise, or anomalies, contained within the training data and, subsequently, fail to generalise well beyond the training data. It is advisable to use a method that allows an estimate of generalisation performance to be obtained from the application of an algorithm to the training data.

A widely used method of parameter selection is to train a classifier on a representative sample of training data using a number of different parameter settings. The generalisation performance of each parameter set can subsequently be estimated by testing the resulting classifiers on another, independent, sample of the training data. The training set may be partitioned, in order to provide a separate validation set for generalisation estimation during parameter setting [Zickus et al., 2002], or partitioned into multiple disjoint subsets for cross-validation [Kohavi, 1995; Molinaro et al., 2005].

The potentially troublesome circumstance of class population imbalance is mentioned above (Chapter 2, § 2.1.7) and must be taken into account during parameter selection. Using the example described above, in which 80% of data in a set belong to Class A and the remaining 20% belong to Class B, if cross-validation were used to partition the data 10 times into 10 independent training (90%) and test (10%) folds, it is conceivable that some folds may contain very few examples of the minority class. It might be better, and more representative of the class distribution, to maintain the original class distribution in each of the ten folds. Otherwise, a performance estimate obtained from the unbalanced cross-validation may suggest a parameter set that defines a classifier which does not generalise well to further examples from the same distribution. To maintain class proportions in each fold of a cross-validation is referred to as *stratified* cross-validation [Kohavi, 1995].

A performance measure intended for use in the presence of class population imbalance is described above and such a measure should be considered when setting free parameters. If overall accuracy is used as a measure of performance during cross-validation over a range of parameters, the winning parameter set will be the one that delivers the greatest overall accuracy averaged over the folds of the cross-validation. This may not be desirable, as the chosen parameters may favour the majority class heavily and achieve the best overall accuracy largely by leading a classifier to ignore the minority class. It appears sensible, therefore, to employ a class weighted accuracy measure, as described above, to estimate the generalisation performance of algorithms and parameter sets that may be required to produce high accuracy when classifying future examples of both classes. In all trials undertaken during this work, 5-fold stratified cross-validation on the training partition is used to set algorithmic parameters. The folds of the cross-validation are selected in a random manner (both positive and negative classes are randomly split into five independent training

and test folds and then combined to maintain the original class proportions in each fold). The cross-validation is carried out ten times for each algorithm and the results averaged to provide a single performance estimate.

### 3.2.4 Performance Comparison

As described in § 3.2.3, algorithmic parameters are determined via performance estimation on partitions of a labelled training set. These parameter values are used subsequently to construct classifiers from all available training data, in order to predict the class labels of examples from the, hitherto unseen, test partition. Thus, the relative generalisation performance of several machine learning algorithms, when applied to a specific ADMET classification task, may be estimated and compared.

It is beneficial to place any observed performance difference between classifiers into context. Context may be provided by attaching a measure of *significance* to a comparison. Here, significance suggests the likelihood that a difference observed on the subset of data used to assess the algorithms would be reflected if the same practice were applied to more data drawn from the same data distribution. There exist parametric methods, which make no assumption as to the distribution of class labels or the errors made when predicting them, with which to assess the significance of agreement (or, conversely, disagreement) between binary predictors (trained two-class classifiers). For example, the McNemar statistic [McNemar, 1947] assesses agreement between binary predictions in the following manner.

The agreement between the predictions of two binary classifiers may be represented in a $2 \times 2$ matrix. In the matrix shown below, $t_1$ represents the correct classifications made by 'classifier 1', $t_2$ the correct classifications made by 'classifier 2' and $f_1$ & $f_2$ the false classifications made by the respective classifiers, with

$$(|t_1| + |f_1|) = (|t_2| + |f_2|) = N$$

where $N$ is the number of classifications made by each classifier.

| | True | False |
|---|---|---|
| True | $t_1 \cap t_2$ | $t_1 \cap f_2$ |
| False | $f_1 \cap t_2$ | $f_1 \cap f_2$ |

The McNemar statistic describes the imbalance observed on the disagreements between the two classifiers, i.e. between the upper-right and lower-left elements of the matrix. The statistic is shown in equation 3.1

$$X^2 = \frac{[(t_1 \cap f_2) - (f_1 \cap t_2)]^2}{(t_1 \cap f_2) + (f_1 \cap t_2)} \tag{3.1}$$

Equation 3.1 may be corrected for discontinuity and becomes

$$X^2 = \frac{[|(t_1 \cap f_2) - (f_1 \cap t_2)| - 1]^2}{(t_1 \cap f_2) + (f_1 \cap t_2)} \quad . \tag{3.2}$$

The integer elements of the agreement matrix may be converted to the respective proportions of the data that they represent [McNemar, 1947]. At small, integer values of $[(t_1 \cap f_2) + (f_1 \cap t_2)]$, $X^2$ follows a binomial distribution. At greater values, the distribution of $X^2$ may be approximated by the $\chi^2$ distribution. The corresponding probability distributions of the McNemar statistic allow one to assess the statistical significance of the classification difference observed between two binary raters.

The McNemar statistic treats all predictions made by two classifiers as having equal weight. As described in §3.2.2, however, a class-weighted accuracy measure is used to assess classifier performance here. A McNemar statistic has been employed to assess significant difference between balanced classifications previously by Guyon et al. [2005]. In the presence of balanced performance assessment, the matrix contributions shown above are themselves weighted and the test statistic applied as usual. In the performance assessments of Chapters 4–6, the McNemar statistic is applied to balanced classifier performance over both classes and, in certain cases, classifier performance on each class in turn. This allows the cause of any observed difference in balanced classification performance to be examined in greater detail. For example, it allows one to determine whether a perceived increase in balanced classifier accuracy is caused by a significant increase in accuracy on both data classes, or whether an increase on one class is responsible, potentially to the detriment of the other class.

## 3.3 Model Building

An adequate range of parameter values is assessed on the training data for each algorithm and care is taken to ensure comparative fairness, via the assessment of similar ranges of possible parameter values for each algorithm. The best performing parameter set for each algorithm, as measured by a stratified, cross-validated performance estimate over the training data, is selected. The parameter set selected for each algorithm is subsequently trained on the complete training set. The solutions obtained are used to classify the test data and a measure of algorithmic performance is obtained by comparing the classifications made by each solution to the true test data labels.

The algorithms used in the comparison are a support vector machine (SVM), a feed-forward artificial neural network (ANN), a radial basis function (RBF) network, a C5.0 decision tree, and a Euclidean distance nearest-neighbour classifier. The C5.0 decision tree, ANN, and RBF network were implemented by the *Clementine* data mining package [SPSS, 2002]. The SVM and nearest-neighbour algorithms were implemented using the mathematical programming package *Matlab* [Mathworks, 2002]. All trials were performed on a

standard PC workstation. A range of parameters were evaluated for each technique, with the parameter set that performed best during 5-fold stratified cross-validation (as described above in § 3.2.3) on the training data selected for evaluation on the test set.

### 3.3.1 SVM Parameters

The standard SVM, used for comparisons and as a development platform over the course of this work, was coded in Matlab primarily by colleague R. Burbidge [Burbidge, 2004] whilst at the Department of Computer Science, UCL. The algorithm minimises the 2-norm of the weight vector and the 1-norm of the slack variables. The decomposition method of Osuna et al. [1997] is incorporated and employs the working-set selection method of Hsu and Lin [2002]. A heuristic early-stopping criterion halts optimisation when a model-based error bound remains unchanged for 1000 iterations.

An SVM requires selection of a kernel function and the regularisation parameter, $C$ (§ 2.2.5, p.49). Linear and quadratic kernels were evaluated with $C = \{1,10,100\}$ (a small set, but found to provide an effective range of regularisation). In addition, RBF kernels were evaluated with the RBF width, $\sigma$, set using five different heuristics (see [Jaakkola et al., 1999] for further details):

| | |
|---|---|
| Hinton: | square root of (no. of data dimensions / 2) |
| Median: | median closest separation distance of training data; |
| Mean: | mean closest separation distance of training data; |
| Jaakkola: | median separation of positive training data to nearest negative; and |
| Jaakkola-Mean: | mean separation of positive training data to nearest negative. |

It is important to note that the Hinton heuristic is used on the assumption of data scaled to be non-dimensional, e.g. attributes lie in [-1,+1]. When results are reported in Chapters 4–6, the above heuristics are abbreviated to 'H', 'Md', 'Mn', 'J' & 'J-M' respectively.

### 3.3.2 ANN & RBF Parameters

The generalisation performance of a single-layer ANN is largely affected by the number of 'hidden' units it employs to map the data to the class attribute [Bishop, 1995a] (cf. § 2.2.6). Too few hidden units and the data may be modelled in insufficient detail. Too many hidden units and the model can overfit noise in the data. The data sets used in this comparison are relatively free of noise, but are complex and class imbalanced, thus a relatively high number of hidden layer units may be expected. Accordingly, a single layer, back propagation ANN was trained using {5,10,15,20,25,35,45,55,65} hidden layer nodes on both the BBB, Tox & Bio data sets, {2,3,4,5,6,8,10,12,14,16} hidden layer nodes on the PrB data set and {1,2,3,4,5} hidden layer nodes on the P-gp data set. Learning rate and momentum were both set to default values of 0.3 (exponentially decaying) and 0.9 respectively for all data sets.

The number of cluster centroids employed by an RBF network to smooth the data before mapping their output to the class attribute is largely analogous to the number of hidden layer

nodes employed by a single layer ANN in this context. Accordingly, the RBF network was trained with the same number of cluster centroids as ANN hidden layer nodes for each data set. *k-means* clustering [Duda et al., 2000] was used to place the cluster centroids before gradient descent maps their outputs to the class attribute.

### 3.3.3 C5.0 Parameters

The balance between the empirical and generalisation performance of a C5.0 decision tree is controlled by *pruning* the tree during training [Quinlan, 1986; Mitchell, 1997] (cf. 2.2.6). The pruning parameter, indicative of the amount pruned from the tree, was set to percentage values from the set {0,10,20,30,40,50,60,70,80,90,100}.

### 3.3.4 Nearest-Neighbour Parameters

A nearest-neighbour classifier [Cover and Hart, 1967; Mitchell, 1997] was used as a conventional benchmark during the comparison. The classifier used Euclidean distance as a measure of similarity between examples. Significant interactions were expected on a local level, therefore the number of neighbours employed ranged over the set {1,3,5,7,9,11,13,15} for all data sets.

## 3.4 Conclusion

A wealth of options are available to the machine learning practitioner for the assessment of algorithmic performance and a wealth of considerations confront the SPC analyst when applying machine learning to ADMET data. The experimental practice described in sections 3.2 & 3.3 does not aim to be the only, or optimum, practice with which to apply machine learning to problems of ADMET classification. Rather, it aims to provide a balanced approach to the fair comparison of machine learning techniques, which acknowledges both the challenging nature of real ADMET data and existing industrial practice. An attempt is made to marry machine learning practice, e.g. the use of stratified cross-validation to select algorithmic free parameters, with facets of drug design practice, such as [Kennard and Stone, 1969] partitioning of labelled data into training and test sets. As such, the data and experimental practice described in this chapter, alongside the background knowledge of Chapter 2, provide a platform for investigation of the research hypotheses stated in Chapter 1. The hypotheses state that SPC analysis will benefit from the successful application of SVMs to ADMET classification and that both technique and application may be adapted to improve performance further. The experimental practice is employed to test the hypotheses by investigation, the nature and results of which are presented in Chapters 4–6.

# Chapter 4

# Support Vector Machines for ADMET Property Classification

The case for further integration of supervised machine learning in order to improve present SPC analysis practice is made in Chapter 2. An experimental practice for the comparison of machine learning techniques on pharmaceutical data is described in Chapter 3. This chapter describes the practical application of the SVM algorithm and a selection of other, widely used, supervised machine learning methods to five sets of ADMET structure-property data provided by GlaxoSmithKline (GSK). The comparison provides an empirical assessment of the arguments made in Chapter 2 and also provides context against which to assess domain-specific adaptations of the SVM algorithm later in the work. With this in mind, two separate comparisons are reported in sections 4.1 & 4.2 of this chapter. The first comparison employs the algorithms involved in an 'off-the-shelf' manner. That is, they are applied to the data with no further consideration of the domain than that outlined in the experimental practice of Chapter 3. The subsequent section employs the same method of comparison, but the algorithms are influenced by two generic methods designed to overcome the effects of having mismatched training data class sizes, i.e. one class of training data is represented in greater number than the other.

## 4.1 Machine Learning Comparison - SVM vs. State-of-the-Art

Table 4.1 displays the class populations of each of the five GSK data sets both before and after being partitioned into training and test sets with the method of Kennard and Stone [1969]. From left to right, the first column displays the data set, the second column displays, from top to bottom, the data set partition (entire set, training partition, or test partition) and the third column displays the number of examples in each partition. The fourth and fifth columns display the number of examples of majority and minority data classes respectively contained within the corresponding partition. The percentage of data belonging to each class is shown in brackets.

All training data attributes were scaled in the range [-1,+1] prior to analysis [Bishop,

| Data Set | Partition | Examples | Majority Examples | Minority Examples |
|---|---|---|---|---|
| BBB | Overall | 476 | 337 (71%) | 139 (29%) |
| | Training | 238 | 153 (64%) | 85 (36%) |
| | Test | 238 | 184 (77%) | 54 (23%) |
| | | | | |
| P-gp | Overall | 138 | 79 (57%) | 59 (43%) |
| | Training | 69 | 42 (61%) | 27 (39%) |
| | Test | 69 | 37 (54%) | 32 (46%) |
| | | | | |
| Tox | Overall | 1176 | 950 (81%) | 226 (19%) |
| | Training | 588 | 452 (77%) | 136 (23%) |
| | Test | 588 | 498 (85%) | 90 (15%) |
| | | | | |
| Bio | Overall | 481 | 393 (82%) | 88 (18%) |
| | Training | 240 | 185 (77%) | 55 (23%) |
| | Test | 241 | 208 (86%) | 33 (14%) |
| | | | | |
| PrB | Overall | 459 | 297 (65%) | 162 (35%) |
| | Training | 230 | 146 (63%) | 84 (37%) |
| | Test | 229 | 151 (66%) | 78 (34%) |

**Table 4.1:** GSK Data: Class Distribution in Training and Test Partitions

1995a] and test data attributes were rescaled accordingly. Figures 4.1–4.5 show multi-dimensional scaling (MDS) plots [Kruskal, 1964] of the training and test partitions of each data set. In each figure, the labelling convention of Chapter 3 is maintained and the two data classes are denoted positive and negative according to the descriptions given in Chapter 3 (section 3.1). In the BBB, Bio & PrB data sets, the positive class is the majority class. In the P-gp & Tox data sets, the negative class is in the majority.

### 4.1.1 Results

Table 4.2 displays algorithmic performance on the test partitions of the five GSK data sets respectively. The table contains seven columns. From left to right, the columns display

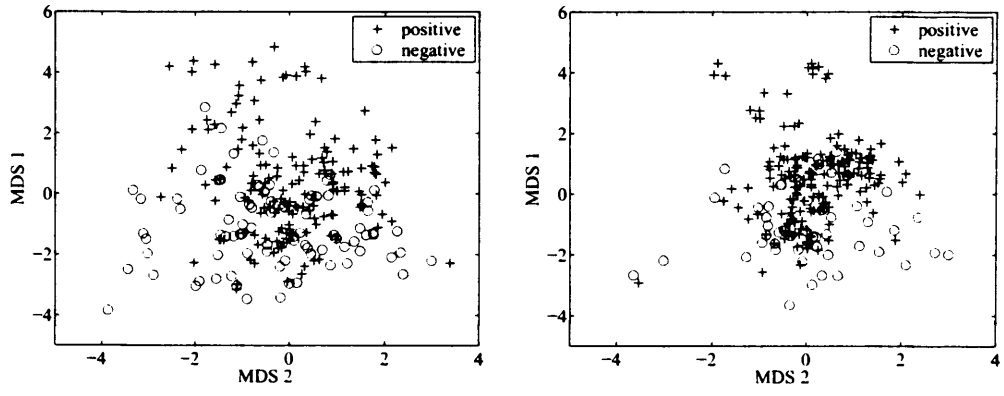


**Figure 4.1:** MDS Plots of BBB Training (left) and Test (right) Partitions

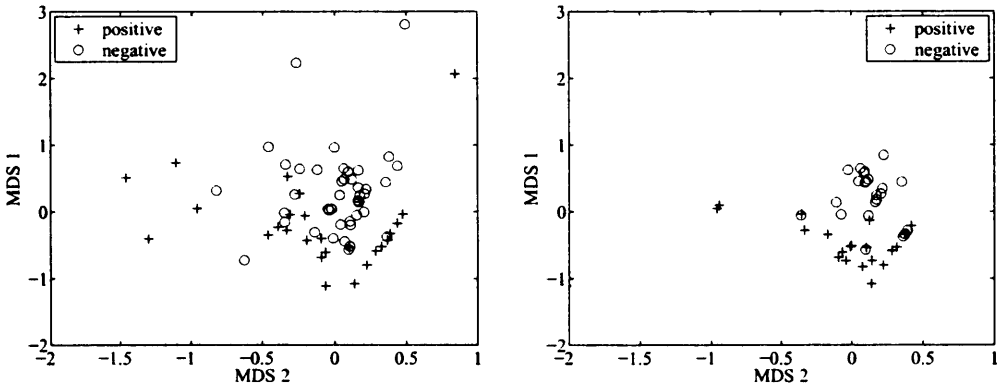


**Figure 4.2:** MDS Plots of P-gp Training (left) and Test (right) Partitions

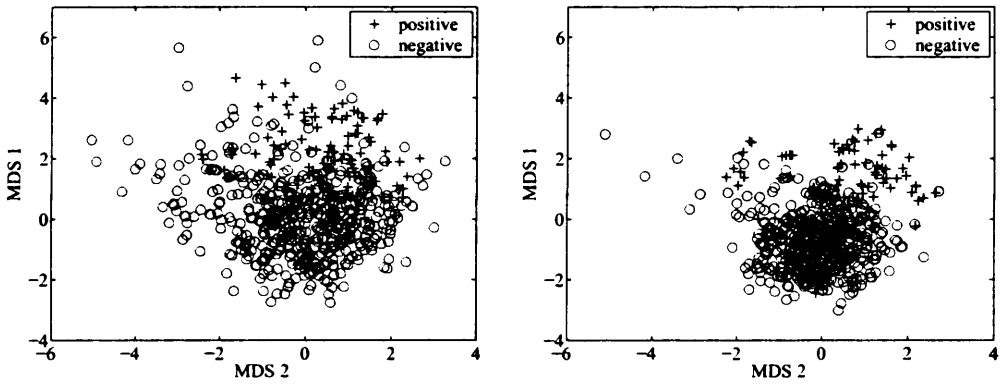


**Figure 4.3:** MDS Plots of Acute Toxicity Training (left) and Test (right) Partitions

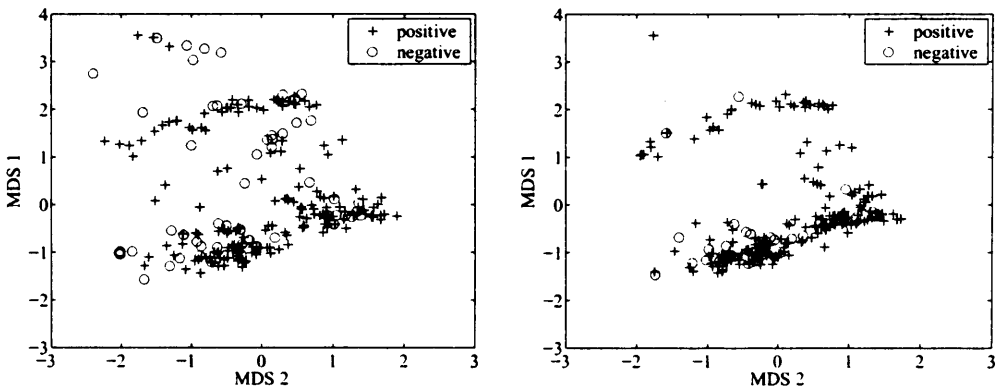


**Figure 4.4:** MDS Plots of Bioavailability Training (left) and Test (right) Partitions

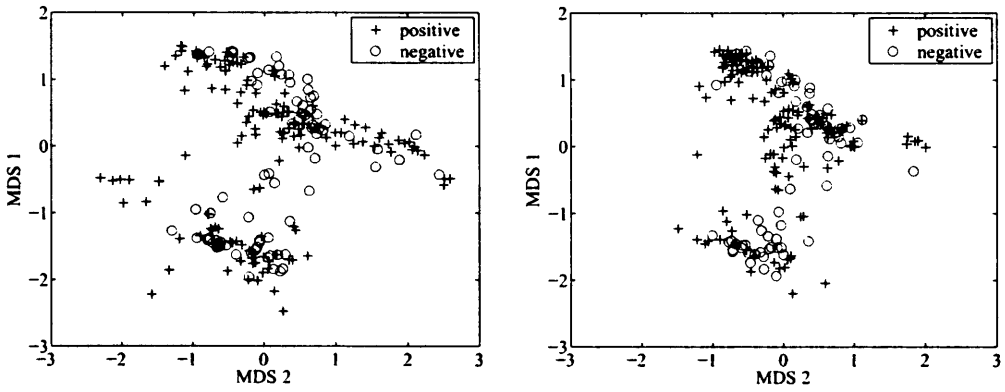


**Figure 4.5:** MDS Plots of Protein Binding Training (left) and Test (right) Partitions

the data set, the algorithm used, the algorithmic parameters as selected by stratified cross validation, the overall accuracy and balanced accuracy of the resulting classifier on the test partition, and classifier accuracy on the majority and minority class test examples respectively. For each data set, the classifier displaying highest balanced accuracy is displayed in bold type.

As discussed in Chapter 3 (§ 3.2.4), a weighted McNemar test of marginal homogeneity may be employed to compare and assess the balanced accuracy measures displayed. A one-tailed weighted McNemar test was applied to classifier output responsible for the contents of table 4.2 in order to produce the results summary below. A difference in performance is deemed *significant* at the 95% level ($p < 0.05$), *marginally significant* if between 90-95% levels ($0.05 \leq p < 0.10$) and not significant otherwise ($p \geq 0.10$).

BBB: The classifier with highest balanced accuracy is an SVM with RBF kernel ($\sigma$ set by the Jaakkola-mean heuristic and $C = 10$). The leading classifier performs significantly better than both linear and quadratic SVMs, better than the RBF network with marginal significance ($p = 0.058$), but not significantly better than the other classifiers.

P-gp: The classifier with highest balanced accuracy is a quadratic SVM ($C = 1$). The leading classifier performs significantly better than its RBF-SVM counterpart, the ANN and the C5.0 decision tree, better than a linear SVM and the RBF network with marginal significance, but not significantly better than the $k$-NN classifier.

Tox: The classifier with highest balanced accuracy is the C5.0 decision tree with pruning parameter set to 90%. The leading classifier performs significantly better than SVMs with quadratic and RBF kernels, better than the RBF network classifier with marginal significance, but not significantly better than the linear SVM, ANN or $k$-NN classifiers.

Bio: The classifier with highest balanced accuracy is the ANN (20 hidden layer nodes). The leading classifier performs significantly better than the linear SVM, better than

| Data | Algorithm | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB | Lin. SVM | $C = 1$ | 0.849 | 0.732 | 0.946 | 0.519 |
| | Quad. SVM | $C = 10$ | 0.832 | 0.761 | 0.891 | 0.630 |
| | **RBF SVM** | $\sigma =$ **'J-M'**, $C = 10$ | **0.870** | **0.805** | **0.924** | **0.685** |
| | ANN | Hid. nodes = 25 | 0.832 | 0.780 | 0.875 | 0.685 |
| | RBF | Centers = 15 | 0.857 | 0.751 | 0.946 | 0.556 |
| | C5.0 | Pruning = 100% | 0.832 | 0.774 | 0.880 | 0.667 |
| | k-NN | $k = 1$ | 0.857 | 0.796 | 0.908 | 0.685 |
| | | | | | | |
| P-gp | Lin. SVM | $C = 1$ | 0.768 | 0.761 | 0.865 | 0.656 |
| | **Quad. SVM** | $C = 1$ | **0.870** | **0.870** | **0.865** | **0.875** |
| | RBF SVM | $\sigma =$ 'J', $C = 10$ | 0.739 | 0.738 | 0.757 | 0.719 |
| | ANN | Hid. nodes = 2 | 0.739 | 0.734 | 0.811 | 0.656 |
| | RBF | Centers = 5 | 0.812 | 0.805 | 0.892 | 0.719 |
| | C5.0 | Pruning = 0% | 0.754 | 0.756 | 0.730 | 0.781 |
| | k-NN | k = 15 | 0.826 | 0.817 | 0.946 | 0.688 |
| | | | | | | |
| Tox | Lin. SVM | $C = 10$ | 0.939 | 0.814 | 0.994 | 0.633 |
| | Quad. SVM | $C = 1$ | 0.930 | 0.800 | 0.988 | 0.611 |
| | RBF SVM | $\sigma =$ 'H', $C = 100$ | 0.932 | 0.801 | 0.990 | 0.611 |
| | ANN | Hid. nodes = 20 | 0.939 | 0.814 | 0.994 | 0.633 |
| | RBF | Centers = 45 | 0.934 | 0.811 | 0.988 | 0.633 |
| | **C5.0** | **Pruning = 90%** | **0.930** | **0.827** | **0.976** | **0.678** |
| | k-NN | k = 3 | 0.920 | 0.812 | 0.968 | 0.656 |
| | | | | | | |
| Bio | Lin. SVM | $C = 10$ | 0.822 | 0.603 | 0.904 | 0.303 |
| | Quad. SVM | $C = 1$ | 0.834 | 0.649 | 0.904 | 0.394 |
| | RBF SVM | $\sigma =$ 'J', $C = 10$ | 0.867 | 0.655 | 0.947 | 0.364 |
| | **ANN** | **Hid. nodes = 20** | **0.851** | **0.671** | **0.918** | **0.424** |
| | RBF | Centers = 55 | 0.863 | 0.627 | 0.952 | 0.303 |
| | C5.0 | Pruning = 20% | 0.822 | 0.654 | 0.885 | 0.424 |
| | k-NN | k = 1 | 0.830 | 0.634 | 0.904 | 0.364 |
| | | | | | | |
| PrB | Lin. SVM | $C = 100$ | 0.734 | 0.659 | 0.894 | 0.423 |
| | **Quad. SVM** | $C = 1$ | **0.756** | **0.715** | **0.841** | **0.590** |
| | RBF SVM | $\sigma =$ 'H', $C = 10$ | 0.716 | 0.667 | 0.821 | 0.513 |
| | ANN | Hid. nodes = 8 | 0.716 | 0.689 | 0.775 | 0.603 |
| | RBF | Centers = 16 | 0.747 | 0.690 | 0.868 | 0.513 |
| | C5.0 | Pruning = 80% | 0.707 | 0.639 | 0.854 | 0.423 |
| | k-NN | k = 3 | 0.694 | 0.654 | 0.782 | 0.526 |

**Table 4.2:** Algorithmic Performance on GSK Test Data

RBF and $k$-NN classifiers with marginal significance, but not significantly better than the other classifiers.

PrB: The classifier with highest balanced accuracy is the quadratic SVM ($C = 1$). The leading classifier performs significantly better than other SVM kernels, the C5.0 decision tree and the $k$-NN classifier, but not significantly better than either ANN or RBF networks.

### 4.1.2 Discussion

Three major trends are apparent from table 4.2 and the results summary.

1. There is no clear 'winner' on any of the five data sets. That is, no classifier records a balanced accuracy that is significantly higher than all others according to the outcome of a one-tailed weighted McNemar test of marginal homogeneity.

2. Despite this, the SVM algorithm appears competitive against the other techniques present. An SVM classifier records highest balanced accuracy on three of the five data sets and is significantly better than at least two of the other classifiers on two of those three sets. On the two sets upon which other classifiers provide higher balanced accuracy, the winning performance is not significantly better than at least one of the SVM classifiers.

3. All methods struggle to classify the minority data class with acceptable accuracy ($\sim 80\%$) on all data sets. Conversely, nearly all classifiers perform acceptably on the majority class of all data sets.

The data supplied by GlaxoSmithKline and the classification scenario that they represent provide a severe examination of all supervised machine learning methods investigated during the comparison. Every method encounters difficulty in classifying unseen examples from the minority class on all data sets. The quadratic SVM on the P-gp data is the only method to achieve an 'industry acceptable' balanced performance of $> 80\%$ accuracy on both classes. In light of this, and the trends outlined above, it may be said only that the SVM algorithm has demonstrated *potential* for successful application to this area of the drug discovery process. As the strongest trend, however, the generalisation imbalance observed between majority and minority data classes warrants further investigation before proceeding.

The five GSK data sets are under the varying influences of molecular representation and sample quality. Four different molecular representations are employed to describe the compound collections encountered in the comparison. The P-gp data set is the only one represented by a small subset of molecular descriptors known, through prior knowledge, to relate to the distribution of the target attribute across chemical space. The other four data sets are described by generic representations, related to aspects of their structural composition and relevant whole-molecular attributes, e.g. hydrophobicity and molecular weight. At

first glance, it may appear surprising that some of the highest balanced classification accuracies are achieved on the data set (P-gp) that comprises the fewest training examples. The P-gp data set is also represented by the fewest and most relevant molecular descriptors and, therefore, covers its input space as well as, or better, than the larger data sets. MDS visualisation of the GSK data sets (Chapter 3) reveals a relatively clear distinction between the mutual similarities of examples that bind to P-gp and those that do not. Despite some outlying examples of both classes, the data appear to cover a single region of chemical space. The Bioavailability and Protein Binding data, however, display mutual similarity between many examples of opposing classes and the data sample appears to lie in two distinct regions of the space defined by inter-example similarities.

In the majority of cases, classification accuracy on the minority class of each problem appears weak in comparison to that on the majority class. It should be noted, however, that a *blind* classifier, which bases predictions on the training class populations alone, would perform much worse. For example, the BBB training partition has a class population split of 64% / 36% (cf. Table 4.1). The winning classifier, RBF-SVM, records approximate class accuracies of 92% / 69% (cf. table 4.2). The minority class accuracy, based on observation of the structural attributes, is nearly twice that which could be achieved without learning from the structural information. Similarly, on the Acute Toxicity data, the class population split is 77% / 23%, but the winning classifier, C5.0, records approximate class accuracies of 98% / 68% and all classifiers record approximate class accuracies $> 95\%$ / $> 60\%$. It is apparent that, despite an imbalance in predictive accuracy that results from the difference in training data class sizes, learning does occur on the structural information presented to the training algorithms. This effect is less clear on the Bioavailability data, for which the class population split is also 77% / 23%, but the winning classifier, ANN, records approximate class accuracies 92% / 42% and minority class accuracies of other classifiers are as low as 30%.

The BBB and Toxicity data are both represented by 72 Volsurf descriptors [Cruciani et al., 2000]. The minority class of the Toxicity training data (136 examples) is of similar size to the majority class of the BBB training data (153 compounds), but algorithmic performance exhibits similar levels of imbalance on the corresponding test partitions (see previous paragraph). An SVM with quadratic kernel function records complete training performance in both cases, therefore, the majority class of training data appears to possess greater influence over orientation of the separating hyperplane created in feature space. In both cases, the QSVM decision boundary is supported by more support vector points of the majority class than of the minority class, but the relative proportion of majority class SV points is lower than that of minority class SV points. Therefore, the hyperplane appears oriented in a manner that will lead to better generalisation on the majority class. These results suggest that the presence of class size imbalance, rather than minority class size, is responsible for performance imbalance. Generalisation accuracy on majority class examples in the BBB test partition is $> 80\%$ for all classifiers, whereas generalisation accuracy on the minority class examples of the Toxicity test partition is $< 70\%$ for all classifiers despite training on

similar amounts of data in each case.

The assumption that the majority class dominates hyperplane orientation due its relative size against the minority class assumes that sample quality of both data classes is equal. The labelled data is extracted from a process designed to retain examples of the class to be retained. Data of the class to be rejected is sampled from any compounds that fail the labelling screen of a compound sample extracted from a combinatorial library and, therefore, may not provide as focused a body of data as the class to be retained. In all but the P-gp data, the majority class is that which is most desirable to retain for further participation in the drug discovery process. The P-gp data is the only set in which the minority class is that to be retained and also the only set upon which an adequate performance balance is obtained. It would be natural to consider, if examples of the majority class are to be retained, whether performance imbalance represents a particular impediment to successful identification of promising development leads. In fact, low generalisation performance on a data class that should be rejected from the process is rather a large impediment. The relationships created from small collections of labelled compounds are likely to be applied subsequently in order to classify many more unlabelled compounds, it is assumed the majority of which will not satisfy all criteria for eventual development into a commercial product. Failure to correctly reject unsuitable compounds may lead to large numbers entering the combinatorial design process and result in the severe disruption of the goal-based search through chemical space that it represents. This training scenario, wherein the desired class appears in relatively high proportion compared to the class to be rejected, is different to that encountered when analysing structure-activity relationships on the output of HTS during earlier stages of the discovery process. For example, the class of data to be retained is in the vast minority (lead identification) and its correct identification paramount.

Deeper examination of SVM classifier performance on the BBB data reveals further potential sources of performance imbalance. Class population imbalance in the training data is moderate (64% / 36%), but has a clear effect upon all classifiers. As for all classification tasks presented, the BBB training data are not linearly separable according to their class labels. The quadratic SVM makes no training errors (it invokes a feature space with ${}^{(72+2)}C_2$ dimensions, of which the 238 training examples occupy a manifold) but generalisation performance is imbalanced. Similarly, the RBF-SVM makes one training error only on the minority class training data and, although it records the best balanced generalisation accuracy, its performance is imbalanced also. Attempts to force regularisation upon these (and any other) classifiers that appear to over-perform on the training data do not improve *balanced* performance. Such attempts to allow training errors in an attempt to avoid overfitting simply lead to training errors on the minority class. This should be expected, because the algorithmic specifications employed for the comparison include no option to treat each class in a distinct manner. For example, from § 2.2.5, a regularised SVM classifier minimises the objective function $||\mathbf{w}||^2 + C\left(\sum_{i=1}^{n} \xi_i^k\right)$, where the right-hand term treats margin transgressions equally (via the single constant multiplier, $C$) regardless of the data class involved. With no instruction to balance the losses made on each class, it is likely that

a relatively small proportion of training errors on the majority class will be accompanied by a relatively large proportion of training errors on the minority class. The result of further regularisation leads the minority class to become indistinguishable from the majority class. This is displayed by the performance of the regularised ($C = 1$) linear SVM classifier, for which the majority of enforced training errors are made on the minority class (training accuracy is 92% / 61% on the majority and minority classes respectively) and minority class generalisation performance is poor accordingly.

As described in Chapter 2 (§ 2.2.4), classifiers with high capacity (in this case, an ANN with many hidden nodes, or SVM with high-dimensional kernel function) are capable of modelling complex relationships between data examples and the target attribute. Classifiers with lower capacity are less capable of modelling complex relationships, but, as a result, have the potential to avoid overfitting the training data and provide better generalisation performance. Here, non-linear SVM classifiers model separation of the training data in a highly accurate manner but fail to perform similarly when presented with new examples of the minority data class. It appears that the training data is being overfitted, but attempts to induce regularisation do not improve results. High capacity is required for the minority class to be distinguished from the majority class in each case. When capacity is reduced, the size and location of the minority class in relation to the majority class leads it to be underfitted. In some respects, classifiers of high capacity appear to act as one-class classifiers [Tax and Duin, 1999] on the majority class. The circumstance of greater information provision for one data class over another, which is not taken into account by unweighted regularisation procedures, presents itself as a primary cause of the performance imbalance observed in table 4.2. A method of reducing the influence of the majority class only in order to treat the minority class in a generalisable manner is required (cf. section 4.2).

Predictive imbalance is much greater on the Bioavailability data, which has the same population imbalance as does the Acute Toxicity data. There appear two differences between these data sets. First, the Bioavailability data has far fewer examples than the Toxicity data. Second, MDS visualisation of the data shows the Bioavailability data to cover its input space in an irregular, or patchy, manner. Examples of the minority class appear scattered across the space occupied by the majority class and many members of the minority class appear more similar to members of the majority class than to members of their own class. Principal component analysis (PCA) is described earlier (Chapter 2, § 2.2.6) and is suggested as a potential solution to poor classification of minority examples in ADME prediction tasks by Trotter and Holden [2003]. One reason for such weakness, aside from the information bias and sample quality issues proposed above, is that the minority class training examples fail to cover adequately a chemical space described by the molecular description employed. For example, the minority class of the Bioavailability training data lies on a manifold of input space. A reduction in the cardinality of chemical space, coupled to the removal of data redundancy (cf. § 2.2.4, p. 40), may focus the previously sparse representation of the minority class afforded by the available training data so as to provide sufficient information with which to improve generalisation. In addition, § 2.2.5 of

Chapter 2 describes how reduction of data dimension reduces the VC dimension of linear classifiers and, hence, may improve generalisation performance.

Table 4.4 displays algorithmic performance after PCA data reduction of both training and test partitions. The training partitions of the original comparison were PCA transformed and the resulting orthogonal feature sets reduced so as to retain 95% of the available information (a limit suggested by the scree test heuristic [Catell, 1966]). Algorithmic free parameters were selected as before [1]. Test partitions were transformed and reduced according to the transformation of their respective training partitions. Rows of table 4.4 in bold type signify those classifiers that exhibit a significant performance increase after PCA data reduction. Table 4.3 displays the dimensionality reduction made to each data set by PCA transformation and reduction.

| Data Set | BBB | P-gp | Tox | Bio | PrB |
|---|---|---|---|---|---|
| Pre-PCA Dimension | 72 | 5 | 72 | 64 | 16 |
| Post-PCA Dimension | 16 | 3 | 18 | 34 | 8 |

**Table 4.3:** Data Dimension Pre- and Post- PCA Transformation

Of 35 applications (7 classifiers to 5 data sets), only 6 classifiers display increased balanced performance (of at least marginal significance) upon PCA data reduction. The classifiers in question are a quadratic SVM on the BBB data, SVMs with linear and RBF kernels plus an ANN on the P-gp data and quadratic and RBF SVMs on the Acute Toxicity data. Most (16) classifiers exhibit no significant increase or decrease in performance, but 13 classifiers exhibit significantly reduced performance upon PCA data reduction.

In several cases, and especially on the Bioavailability data that prompted discussion of sparse input space coverage, PCA reduction appears to magnify conclusions drawn from the original comparison. Few classifiers perform significantly better on the PCA-reduced data sets, whereas several perform significantly worse. This pattern persists when information retained by PCA reduction is increased to 99%, lowered to 90% and also when univariate feature selection, of the sort used regularly on gene expression microarray studies [Shipp et al., 2002], is employed to select a subset of the original features (results unreported). It appears, therefore, that focusing the description of chemical space, by mapping to orthogonal, non-redundant axes, may magnify the effects of class imbalanced and undersampled training data.

It remains to consider how well the framework for algorithmic comparison, described in Chapter 3, performed its task when producing the results described above. All algorithmic parameters displayed in tables 4.2 & 4.4 were selected by stratified cross-validation on a class-balanced accuracy measure. Empirical trials, not reported above in the interests of space and concision, reinforce the judgment that this is a good way to select the best model. Parameters selected using the overall accuracy, rather than balanced accuracy, dur-

[1] It is noted that the Hinton RBF kernel width heuristic (cf. p. 80) should strictly be applied to data scaled to be non-dimensional, e.g. [-1,+1]. The PCA reduced data here is not, but the reduced data attribute ranges are similar - PCA was applied to scaled data - and within an order of magnitude.

| Data | Algorithm | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB | LSVM | $C = 10$ | 0.832 | 0.689 | 0.951 | 0.426 |
| | **QSVM** | $C = 1$ | **0.866** | **0.795** | **0.924** | **0.667** |
| | RBF SVM | $\sigma$ = 'H', $C = 1$ | 0.840 | 0.687 | 0.967 | 0.407 |
| | ANN | Hid. nodes = 10 | 0.836 | 0.783 | 0.880 | 0.685 |
| | RBF | Centers = 45 | 0.824 | 0.690 | 0.935 | 0.444 |
| | C5.0 | Pruning = 100% | 0.807 | 0.659 | 0.929 | 0.389 |
| | $k$-NN | $k = 5$ | 0.832 | 0.721 | 0.924 | 0.519 |
| | | | | | | |
| P-gp | **LSVM** | $C = 100$ | **0.826** | **0.823** | **0.865** | **0.781** |
| | QSVM | $C = 100$ | 0.826 | 0.823 | 0.865 | 0.781 |
| | **RBF SVM** | $\sigma$ = **'H'**, $C = 10$ | **0.826** | **0.823** | **0.865** | **0.781** |
| | **ANN** | **Hid. nodes = 2** | **0.812** | **0.805** | **0.892** | **0.719** |
| | RBF | Centers = 5 | 0.797 | 0.792 | 0.845 | 0.719 |
| | C5.0 | Pruning = 0% | 0.768 | 0.782 | 0.595 | 0.969 |
| | $k$-NN | $k = 3$ | 0.841 | 0.837 | 0.892 | 0.781 |
| | | | | | | |
| Tox | LSVM | C = 100 | 0.896 | 0.679 | 0.992 | 0.367 |
| | **QSVM** | $C = 1$ | **0.918** | **0.820** | **0.962** | **0.678** |
| | **RBF SVM** | $\sigma$ = **'H'**, $C = 100$ | **0.937** | **0.831** | **0.984** | **0.678** |
| | ANN | Hid. nodes = 15 | 0.920 | 0.798 | 0.974 | 0.622 |
| | RBF | Centers = 55 | 0.925 | 0.792 | 0.984 | 0.600 |
| | C5.0 | Pruning = 90% | 0.917 | 0.787 | 0.974 | 0.600 |
| | $k$-NN | $k = 1$ | 0.901 | 0.814 | 0.940 | 0.689 |
| | | | | | | |
| Bio | LSVM | $C = 100$ | 0.739 | 0.632 | 0.779 | 0.485 |
| | QSVM | $C = 1$ | 0.826 | 0.580 | 0.918 | 0.242 |
| | RBF SVM | $\sigma$ = 'H', $C = 100$ | 0.859 | 0.651 | 0.938 | 0.364 |
| | ANN | Hid. nodes = 10 | 0.851 | 0.633 | 0.933 | 0.333 |
| | RBF | Centers = 65 | 0.863 | 0.577 | 0.971 | 0.182 |
| | C5.0 | Pruning = 70% | 0.797 | 0.551 | 0.889 | 0.212 |
| | $k$-NN | $k = 1$ | 0.834 | 0.636 | 0.909 | 0.364 |
| | | | | | | |
| PrB | LSVM | $C = 1$ | 0.721 | 0.639 | 0.894 | 0.385 |
| | QSVM | $C = 1$ | 0.707 | 0.682 | 0.762 | 0.603 |
| | RBF SVM | $\sigma$ = 'H', $C = 10$ | 0.725 | 0.683 | 0.815 | 0.551 |
| | ANN | Hid. nodes = 14 | 0.703 | 0.666 | 0.782 | 0.551 |
| | RBF | Centers = 16 | 0.690 | 0.629 | 0.821 | 0.436 |
| | C5.0 | Pruning = 90% | 0.659 | 0.649 | 0.682 | 0.615 |
| | $k$-NN | $k = 9$ | 0.699 | 0.669 | 0.762 | 0.577 |

**Table 4.4:** Algorithmic Performance on PCA-Reduced GSK Test Data

ing cross-validation favour the majority class more than is shown in table 4.2. In a large majority of cases, the algorithmic parameters selected by five-fold stratified cross validation on the training data provide balanced generalisation performance better than, or at least not significantly worse than, that provided by other parameter selections across the ranges assessed. The only circumstance in which this proves not to be the case is in the selec-

tion of algorithmic parameters for certain algorithms on the P-gp data. The parameter sets selected for linear and quadratic SVMs, ANN and C5.0 are all better, or not significantly worse, than the best performing parameter set within the ranges assessed. Parameters for RBF-SVM, RBF network and $k$-NN algorithms, which all assess inter-example similarity and/or approximate the training data, do not generalise as well as most other parameter sets available. All three classifiers are, in fact, capable of matching or bettering the performance of a quadratic SVM, described above as very good, but the chosen parameter sets overfit the data considerably by comparison. One relatively sensible interpretation of this situation is that the partitions created by random sampling such a small amount of data during stratified cross validation do not provide a realistic appraisal of the generalisation performance of such methods from the entire training set. Thus, the results reported for RBF-SVM, RBF network and $k$-NN classifiers on the P-gp set are potentially pessimistic.

An alternative to the use of cross-validation is to perform two applications of Kennard & Stone partitioning of the original data sets, the first in order to provide training data and the second to split remaining data into hold-out and validation sets. The hold-out set would provide performance assessment for the selection of algorithmic parameters, prior to the estimation of generalisation performance on the test partition as above. Another alternative practice would be to repeat the comparison having reversed the training and test partitions employed above (the original data are split 50/50), in order to investigate persistence of the trends observed. This method is a common approach in drug design, as is the repetition of performance assessment on classifiers trained upon scrambled class labels [Wold and Eriksson, 1995].

It is appreciated that wider free parameter ranges could be made available to some of the algorithms assessed during the comparison. The decision was taken to provide a limited but similarly-sized range of free parameter values to all algorithms during the comparison. Although wider ranges would have the potential to provide different results, the ranges employed are sufficient to prompt exhibition of the trends discussed above and provide a fair comparison of the techniques involved. One recommended change to this approach is to evaluate multiples of a single RBF width heuristic, rather than to employ the five heuristics of the comparison, because the non-Hinton heuristics employed produce similar values on occasion.

The comparison of supervised machine learning methods on five real SPC analysis problems reveals much regarding the nature of the domain and the impediments that it presents to successful analysis. The comparison employs a detailed framework of experimental practice, but no action is taken to alter data or algorithms within that framework in order to overcome the challenges presented. The SVM algorithm is seen to perform competitively alongside machine learning techniques already well-used for SPC analysis. Given the straightforward nature of application to such problematic tasks, it is unlikely that one supervised learning algorithm would perform much better than all others in all circumstances (unless custom designed with the task in mind, cf. Chapter 6, or embodying a particularly different approach to the task of learning). Rather, it is in how an existing learning algo-

rithm may be adapted to perform adequately in such circumstances that may provide a more relevant measure of success. The remainder of this chapter investigates generic remedies to performance imbalance. Chapters 5 and 6 introduce SPC analysis approaches specific to SVMs.

## 4.2 Balancing Generalisation Performance

The situation wherein one data class is classified preferentially over another is not desirable (cf. Chapter 2; § 2.1.7). When screening in the early lead optimisation stage of the drug discovery process, it is almost as important to reject non-drugs as it is to retain potential new drugs. This section investigates approaches designed to remedy such performance imbalance. The first approach weights the influence of the minority training data class as higher than that of the majority class, so as to balance the treatment of both data classes during training. The second approach concerns a strategic reduction of the majority data class so as to balance the training data and, thus, the level of information provided to a learning algorithm during classifier creation. The reduction strategy entails the removal of majority class examples in a manner designed to retain the information most typical of the distribution that the class represents.

The application of variable misclassification costs to the SVM algorithm is discussed earlier in Chapter 2 (§ 2.2.5). The C5.0 and $k$-NN algorithms may be weighted also in order to coerce them into treating the accurate classification of one class with higher priority than that of another. Several methods exist by which to incorporate variable misclassification costs into the SVM algorithm. The method employed here is similar to that suggested by Osuna et al. [1997] and used by (among others) Joachims [1998a]; Brown et al. [2000] and Shin and Cho [2003]. The regularisation parameter (and thus the upper bound on classifier weights) is set to different levels for each data class, which has the effect of varying tolerance of misclassification according to the class of example misclassified. The higher of the two parameters results in lower tolerance of misclassification for the corresponding class of data.

The question remains as to how much to raise and / or lower the regularisation parameter for each class from the equilibrium point at which they would be trained without weighting. The relative importance of each class to the other is unknown. The only principled weighting that may be applied without introducing another free parameter into the training procedure is that which is inverse to the ratio of training class population sizes. A normalised weighting of $n/2n_+$ for positive examples and $n/2n_-$ for negative examples provides a similar weighting scheme to that employed thus far to calculate a balanced classification accuracy over the test examples. As described in Chapter 3 (§ 3.3.1), the SVM used during this comparison optimises over the 1-norm of the slack variables, which results in the Lagrange multiplier of each example being upper bound by the regularisation constant $C$ in the SVM constrained optimisation. The above weighting scheme may be applied directly to the SVM upper bound. Thus, multipliers of positive class examples are upper bound by $C_+ = n/2n_+$ and the multipliers of negative class examples are bound by $C_- = n/2n_-$.

The same weighting structure may be applied simply to both C5.0 decision trees and $k$-NN classifiers. Decision trees partition the data attribute-by-attribute (recursively) until the training data is separated according to the class labels. At each iteration, the most informative of the remaining attributes is chosen to form the next node in the tree structure. The ranking measure employed commonly observes the information regarding training data classification contained within each attribute. Variable misclassification costs may be applied at this point, allowing the attribute values of minority class examples greater influence during information gain calculations, with the intention of creating a classifier more likely to classify minority class examples correctly. The *Clementine v7.1* [SPSS, 2002] implementation of C5.0, employed here, permits the association of different misclassification costs with each data class. The results displayed below are the result of using the weighting described above to associate different misclassification costs with the two classes present in each problem. It is also simple to weight a $k$-NN classifier with variable misclassification costs. When presented with an unlabelled example, the $k$-NN algorithm classifies it via a vote across class labels of the $k$ nearest training examples. The unit vote attributed to each of the $k$ class labels is replaced by a positive constant that depends upon the corresponding class label. For example, in the scenario described above, each vote for the majority class would be worth $n/2n_+$ and each vote for the minority class would be worth $n/2n_-$ when summed in order to classify an unseen example.

Whereas variable misclassification costs implemented in the above manner affect the C5.0 and $k$-NN algorithms regardless of circumstance, the weighted SVM implementation is restricted to certain cases. The reason for non-universality of the weighted SVM, implemented as described above, is that variable misclassification costs are attached to the regularisation parameter, which only acts when training errors are made. An SVM solution that is stable and performs strongly over the training data is likely to remain unaffected by the relatively subtle re-weighting of the regularisation parameter described above. Such occurrences are noted during analysis of experimental results in subsequent sub-sections. Strong SVM classifiers may be balanced by scaling the reweighted regularisation so that the majority parameter is very small compared to the minority parameter, in order to enforce regularisation upon the majority class. Imposing regularisation upon such situations in order to balance performance involves more intensive parameter selection, usually involving cross-validation or some other assessment of generalisation performance over a grid of separate regularisation parameter values for the two classes (asymmetric regularisation). In the situation wherein training data present no impediment to separation after transformation to a high-dimensional feature space, enforced regularisation lessens the effect of majority class boundary determination by ignoring majority class examples nearest the boundary. This is discussed further in § 4.2.2.

As an alternative to algorithmic weighting in order to overcome performance imbalance, one may also consider revision of the data presented to a standard algorithm. Previous attempts to overcome the negative effects of class population imbalance have included the reduction of majority class data, via random sampling, and expansion of minority class

data, via the addition of Gaussian noise about existing minority class examples [Bishop, 1995b]. In agreement with Shin and Cho [2003], it is not expected that a random sample of the majority class data will provide best results, because it may lead to the loss of significant patterns, or internal structure, from the data (cf. Chapter 3, § 3.2.1). Furthermore, several random reductions would be required to provide significant experimental results, which negatively impacts upon the practical use of such a method and may also impede the comparison of results obtained from its use.

Strategic sampling from an abundance of data is not a new concept [Provost et al., 1999; Lee et al., 2001] and methods to do so include the identification and use of only those examples with nearest-neighbours of mixed class, in order to extract the region around any potential decision boundary from a large body of labelled data [Shin and Cho, 2003]. The introductory chapter of this thesis describes how potential improvements to present machine learning practice, and SVMs in particular, are sought in conjunction with approaches that facilitate incorporation into present drug discovery practices. Upon consideration of how best to reduce a majority class in order to balance the training data, a method widely used in combinatorial library design presents itself as a potentially useful reduction strategy.

In order to reduce the population of the majority class to a size similar to that of the minority class, it appears sensible to select examples that are 'typical' of the bulk of the majority class. The Mahalanobis distance (§ 2.2.6, p. 67) is widely used in drug discovery as a method of outlier assessment [Dominik, 2000]. The retention of majority class examples with lowest Mahalanobis distance to the entire body of majority class data may typify the majority class in the manner desired. Accordingly, majority class examples are assessed against the shape and location of the majority class. If the minority class contains $n_-$ examples, the $|n_-|$ majority class examples with lowest Mahalanobis distance are selected for use in a balanced training set and the rest excluded from training.

The removal of training examples from a data-poor scenario is opposed to the ideal situation of having more labelled data upon which to train. Nevertheless, and as seen in the comparison of section 4.1, the undesirable effects of performance imbalance are most likely to arise in the presence of imbalanced training data class populations, regardless of their size. Enforcing equal training data class sizes in a relevant manner may eliminate this source of performance imbalance without damaging majority class performance to the extent that its reduction outweighs the expected increase in minority class performance. For SVMs, an additional benefit of this approach is that it should effect the training algorithm regardless of the strength of original training performance. The results displayed in § 4.2.1 represent a first look at the use of an existing pharmaceutical analysis procedure for the strategic balancing of training data class as an alternative to extensively tuned asymmetric regularisation. Existing methods of algorithmic weighting is also assessed in order to provide context. Subsequent analysis of the results assesses the effect of both approaches on ADMET SPC analysis and includes suggestions for further development.

### 4.2.1 Results

The same experimental practice as employed in section 4.1 is employed here to assess both balancing methods described above. Results are displayed in tables 4.5 & 4.6 using the same tabular style as in section 4.1. To compare class-weighted algorithms, an SVM weighted as described above and with linear, quadratic, and RBF kernels was applied to the five GSK data sets. Similarly weighted versions of the C5.0 and $k$-NN algorithms were applied as benchmarks. ANN and RBF networks are not present in this comparison because no facility to weight them was available in the *Clementine v7.1* implementation employed [SPSS, 2002]. It is important to note, however, that both techniques may be weighted directly [Schwenk and Bengio, 1998].

Figures 4.6–4.10 display MDS plots of each data set training partition before (left) and after (right) Mahalanobis reduction. Results of the Mahalanobis reduction trial are displayed in table 4.6. All algorithms compared previously in section 4.1 are trained on Mahalanobis-reduced training sets and generalisation performance assessed on the usual test partitions. In order to replicate the eventual application of majority class reduction to the entire training set, Mahalanobis reduction is performed on each training fold of the stratified cross-validation when setting free-parameters.

As before, a one-tailed weighted McNemar test was applied to assess the significance of observations made from tables 4.5 & 4.6. Any increase in balanced accuracy of at least marginal significance ($p < 0.10$) when observed against the results of table 4.2 is signified by bold type. A similarly significant increase in minority class accuracy against the original application is marked with an asterisk.

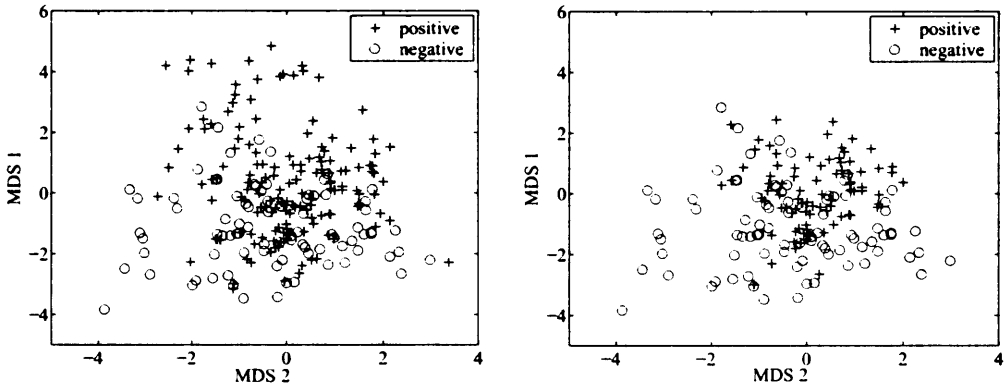


**Figure 4.6:** Original (left) and Mahal-Reduced (right) BBB Training Data

### 4.2.2 Discussion

The results of tables 4.5 & 4.6 suggest that balanced accuracy as a result of algorithmic weighting and strategic majority class reduction is moderately increased when compared to the standard application of section 4.1. No classifier performance is significantly reduced by balancing.

| Data | Algorithm | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB | **Wei. LSVM** | $C = 1$ | **0.832** | **0.806** | **0.853** | **0.759*** |
| | Wei. QSVM | $C = 10$ | 0.828 | 0.758 | 0.886 | 0.630 |
| | Wei. RBF SVM | $\sigma =$ 'J-M', $C = 10$ | 0.861 | 0.799 | 0.913 | 0.685 |
| | Wei. C5.0 | Pruning = 100% | 0.836 | 0.776 | 0.886 | 0.667 |
| | Wei. $k$-NN | $k = 11$ | 0.811 | 0.780 | 0.837 | 0.722 |
| | | | | | | |
| P-gp | **Wei. LSVM** | $C = 1$ | **0.855** | **0.861** | **0.784** | **0.938*** |
| | Wei. QSVM | $C = 1$ | 0.841 | 0.847 | 0.757 | 0.938 |
| | **Wei. RBF SVM** | $\sigma =$ **'J'**, $C = 1$ | **0.841** | **0.847** | **0.757** | **0.938*** |
| | Wei. C5.0 | Pruning = 100% | 0.754 | 0.734 | 1.000 | 0.469 |
| | Wei. $k$-NN | $k = 5$ | 0.841 | 0.847 | 0.757 | 0.938* |
| | | | | | | |
| Tox | **Wei. LSVM** | $C = 1$ | **0.939** | **0.868** | **0.970** | **0.767*** |
| | **Wei. QSVM** | $C = 1$ | **0.932** | **0.819** | **0.982** | **0.656*** |
| | **Wei. RBF SVM** | $\sigma =$ **'H'**, $C = 10$ | **0.942** | **0.870** | **0.974** | **0.767*** |
| | Wei. C5.0 | Pruning = 100% | 0.918 | 0.820 | 0.962 | 0.678 |
| | **Wei. $k$-NN** | $k = 9$ | **0.893** | **0.837** | **0.918** | **0.756*** |
| | | | | | | |
| Bio | **Wei. LSVM** | $C = 1$ | **0.822** | **0.718** | **0.861** | **0.576*** |
| | Wei. QSVM | $C = 1$ | 0.817 | 0.639 | 0.885 | 0.394 |
| | Wei. RBF SVM | $\sigma =$ 'J', $C = 10$ | 0.859 | 0.676 | 0.928 | 0.424 |
| | Wei. C5.0 | Pruning = 20% | 0.805 | 0.658 | 0.861 | 0.455 |
| | **Wei. $k$-NN** | $k = 3$ | **0.788** | **0.724** | **0.813** | **0.636*** |
| | | | | | | |
| PrB | Wei. LSVM | $C = 10$ | 0.686 | 0.703 | 0.649 | 0.756* |
| | Wei. QSVM | $C = 1$ | 0.725 | 0.733 | 0.709 | 0.756* |
| | **Wei. RBF SVM** | $\sigma =$ **'H'**, $C = 10$ | **0.712** | **0.732** | **0.669** | **0.795*** |
| | **Wei. C5.0** | **Pruning = 80%** | **0.703** | **0.691** | **0.729** | **0.654*** |
| | **Wei. $k$-NN** | $k = 11$ | **0.694** | **0.712** | **0.656** | **0.769*** |

**Table 4.5:** Weighted Algorithm Performance on GSK Data Test Data

| Data | Algorithm | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB | **LSVM** | $C = 1$ | **0.836** | **0.790** | **0.875** | **0.704*** |
| | **QSVM** | $C = 1$ | **0.845** | **0.808** | **0.875** | **0.741*** |
| | RBF SVM | $\sigma$ = 'H', $C = 1$ | 0.811 | 0.767 | 0.848 | 0.685 |
| | ANN | Hid. nodes = 25 | 0.828 | 0.790 | 0.859 | 0.722 |
| | RBF | Centers = 10 | 0.815 | 0.782 | 0.842 | 0.722* |
| | C5.0 | Pruning = 100% | 0.840 | 0.799 | 0.875 | 0.722 |
| | $k$-NN | $k = 9$ | 0.870 | 0.778 | 0.946 | 0.611 |
| | | | | | | |
| P-gp | **LSVM** | $C = 1$ | **0.855** | **0.854** | **0.865** | **0.844*** |
| | QSVM | $C = 10$ | 0.826 | 0.832 | 0.757 | 0.906 |
| | RBF SVM | $\sigma$ = 'J', $C = 1$ | 0.797 | 0.807 | 0.676 | 0.938* |
| | ANN | Hid. nodes = 4 | 0.783 | 0.781 | 0.811 | 0.750 |
| | RBF | Centers = 3 | 0.855 | 0.853 | 0.892 | 0.813 |
| | C5.0 | Pruning = 0% | 0.754 | 0.734 | 1.000 | 0.469 |
| | $k$-NN | $k = 9$ | 0.841 | 0.847 | 0.757 | 0.938* |
| | | | | | | |
| Tox | **LSVM** | $C = 1$ | **0.927** | **0.852** | **0.960** | **0.744*** |
| | **QSVM** | $C = 1$ | **0.881** | **0.825** | **0.906** | **0.744*** |
| | **RBF SVM** | $\sigma$ = **'H'**, $C = 1$ | **0.927** | **0.843** | **0.964** | **0.722*** |
| | **ANN** | **Hid. nodes = 35** | **0.925** | **0.828** | **0.968** | **0.689*** |
| | **RBF** | **Centers = 20** | **0.920** | **0.853** | **0.950** | **0.756*** |
| | C5.0 | Pruning = 100% | 0.893 | 0.828 | 0.922 | 0.733 |
| | **$k$-NN** | $k = 15$ | **0.924** | **0.841** | **0.960** | **0.722*** |
| | | | | | | |
| Bio | **LSVM** | $C = 1$ | **0.739** | **0.696** | **0.755** | **0.636*** |
| | QSVM | $C = 100$ | 0.701 | 0.687 | 0.707 | 0.667* |
| | RBF SVM | $\sigma$ = 'H', $C = 10$ | 0.681 | 0.636 | 0.697 | 0.576* |
| | ANN | Hid. nodes = 35 | 0.743 | 0.685 | 0.764 | 0.606* |
| | RBF | Centers = 55 | 0.660 | 0.675 | 0.654 | 0.697* |
| | **C5.0** | **Pruning = 40%** | **0.755** | **0.731** | **0.764** | **0.697*** |
| | **$k$-NN** | $k = 1$ | **0.764** | **0.672** | **0.798** | **0.546*** |
| | | | | | | |
| PrB | LSVM | $C = 100$ | 0.642 | 0.673 | 0.576 | 0.769* |
| | QSVM | $C = 1$ | 0.686 | 0.687 | 0.682 | 0.692* |
| | RBF SVM | $\sigma$ = 'H', $C = 1$ | 0.738 | 0.677 | 0.868 | 0.487 |
| | ANN | Hid. nodes = 10 | 0.677 | 0.678 | 0.676 | 0.680* |
| | RBF | Centers = 16 | 0.686 | 0.681 | 0.695 | 0.667* |
| | C5.0 | Pruning = 90% | 0.646 | 0.661 | 0.616 | 0.705* |
| | **$k$-NN** | $k = 11$ | **0.712** | **0.698** | **0.742** | **0.654*** |

**Table 4.6:** Mahalanobis-Reduction: Algorithm Performance on GSK Test Data

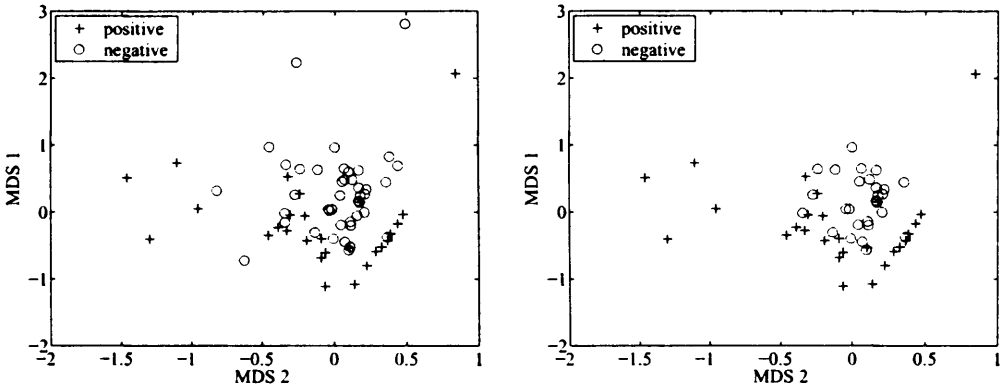


**Figure 4.7:** Original (left) and Mahal-Reduced (right) P-gp Training Data

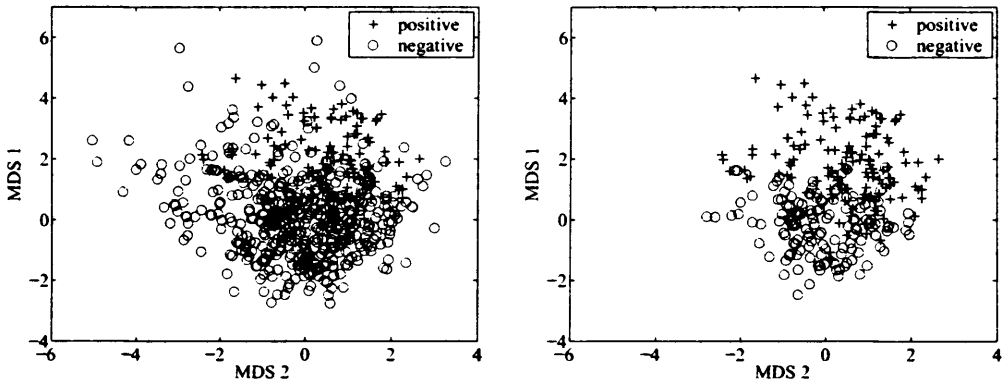


**Figure 4.8:** Original (left) and Mahal-Reduced (right) Acute Toxicity Training Data

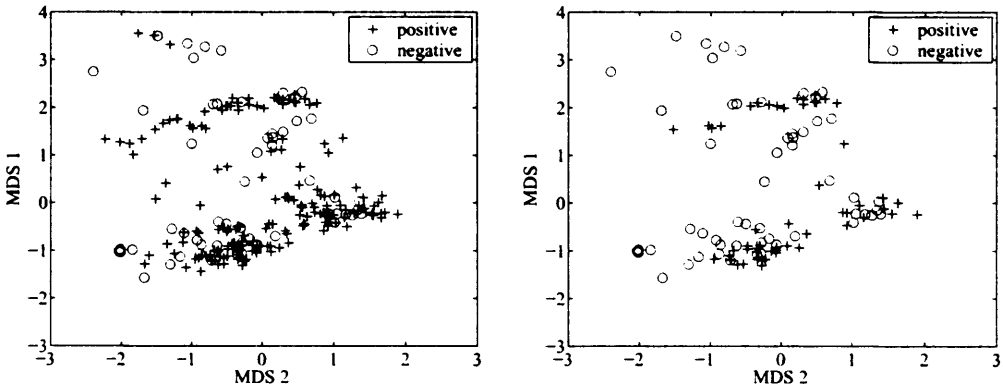


**Figure 4.9:** Original (left) and Mahal-Reduced (right) Bioavailability Training Data

The weighted linear SVM displays significant increases in balanced accuracy against its unweighted performance in 4 of 5 applications. As suggested earlier, unscaled weighting does not appear to affect non-linear SVMs in situations where they perform strongly on the training data (the difference in class-specific regularisation parameters is not great enough to enforce regularisation upon the majority class), i.e. on the BBB and Bioavailability data. The only contradictions to this are a slight perturbation of quadratic SVM performance bal-

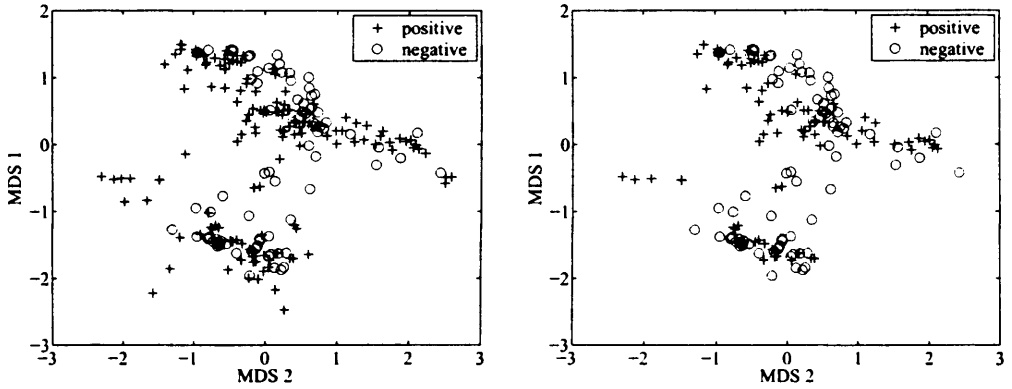


**Figure 4.10:** Original (left) and Mahal-Reduced (right) Protein Binding Training Data

ance on the P-gp data, upon which unbalanced classifiers displayed generally good levels of performance balance, and an increase in quadratic SVM performance on the Acute Toxicity data. A marginally significant increase in balanced performance is exhibited, but the effect is lower than for linear and RBF SVMs, which make training errors.

The weighted $k$-NN classifier exhibits significantly higher balanced accuracy than when unweighted in three of five applications. Furthermore, minority class accuracy is increased on the P-gp data, but does not produce a significant increase in balanced accuracy. A decrease in majority class accuracy on the BBB data considerably outweighs a small increase in minority class accuracy. By comparison, the weighted C5.0 algorithm is not affected successfully, except on the Protein Binding data. Poor performance balance on the P-gp data, the opposite effect on performance balance to that desired, results from the creation of a particularly small tree that is unable to successfully distinguish the minority class from the majority class. Examination of the tree structure suggests that it lacks the capacity to successfully recognise both classes of data because weighting the recursive partitioning process on such a small collection of training data produces a tree with very few nodes. This situation also exposes weakness in a sole reliance upon balanced accuracy to compare algorithmic performance. It is stated above that no classifier displays significantly reduced balanced accuracy as a result of the measures employed. This is true, but C5.0 performance is clearly undesirable.

The application of an unweighted linear SVM to Mahalanobis-balanced training data exhibits increased performance in the same four circumstances as for algorithmic weighting. In addition, quadratic SVM performance increases significantly on the BBB data (upon which it made no training errors previously) and, likewise, quadratic and RBF SVMs on the Acute Toxicity data. The ability of majority class reduction to affect the original performance of high capacity classifiers on the Toxicity data is further confirmed by significant increases in ANN and RBF network performance after training on the balanced data.

The $k$-NN classifier displays increased minority class accuracy in the same three applications as observed for algorithmic weighting. $k$-NN performance on the BBB data is reduced by majority reduction. C5.0 performance only increases on the majority reduced

Bioavailability data and the same problem as observed for algorithmic weighting is also displayed by C5.0 on the majority reduced P-gp data.

Significant increases in minority class accuracy are exhibited by nearly all classifiers on both Bioavailability and Protein Binding problems, but comes at the cost of an even larger decrease in majority class accuracy in most cases. Such an effect does not represent any particular success, as it is not desirable to reduce majority class performance to unacceptable levels in order to obtain an improvement in minority class performance - especially as one would wish to retain the majority class in both of these problems. It appears that, on the Bioavailability and Protein Binding data, any attempt to balance performance merely 'pivots' performance on majority and minority classes about a balanced accuracy that remains roughly the same. One feasible conclusion is that the value about which balanced generalisation appears to pivot is the limit of classifier generalisation ability on these data sets, i.e. the structural information provided can be mapped to the target attribute in no more successful manner. Other conclusions are drawn below.

The above observations raise questions regarding the nature and effect of attempting to balance algorithmic treatment of the two data classes involved in the ADMET classification tasks. First, one may consider why majority class accuracy remains higher than minority class accuracy in the majority of balanced applications on the BBB, Toxicity and Bioavailability training data and, second, why majority class accuracy is lower than minority class accuracy in several balanced applications on the P-gp and Protein Binding data.

The former case may be related to the discussion of sample quality bias in § 4.1.2. The three problems in which majority class accuracy remains greater than minority class accuracy after balancing, albeit to a lesser extent than without balancing, are those in which the majority class should be retained by the discovery process. Any potential quality bias towards the retained class might explain why retained class performance of classifiers balanced during training remains higher than rejected class performance. Even if there exists no quality bias toward the retained class, the Mahalanobis reduction method is likely to induce one, by removing outliers from the reduced class but not providing the same favour to the other class. Similarly, algorithmic weighting will provide greater credence to all members of the minority class regardless of their quality. A potential remedy might be to apply algorithmic weighting with individual weights for each example that assess both class membership and some measure of typicality, such as the Mahalanobis distance.

The latter case is apparent in the results for P-gp and Protein Binding data. The P-gp data arguably requires little balancing, as long as a classifier of sufficiently high capacity is employed. The Protein Binding data, however, does require balancing measures and, upon their application, occasionally reverses the imbalance observed originally. In this circumstance, balancing may lead to under description of the majority class. Especially when using reduction, the previous majority class is 'shrunk' about its location (regularisation ignores examples closest to the eventual decision boundary) and, therefore, the reduced class may no longer reflect the wider distribution of that class across chemical space. The class to be retained (the majority in the Protein Binding data) is likely to occupy a more distinct region

of chemical space than the class to be rejected. Again, even when weighting and reduction result in well-balanced generalisation performance for both classes, it may be considered that the level obtained is the generalisation limit of the structure-property information provided.

In cases wherein one wishes the majority class accuracy to remain at levels similar to those of minority class accuracy after performing some form of balancing during training, a scaling parameter may be employed and assessed by cross-validation over a range of values in order to control the effects of balancing (as for enforced asymmetric SVM regularisation [Brown et al., 2000]). Alternately, reduction strategies may be stopped prior to class population balance according to minimum difference between cross-validated performance measures on each class. Likewise, on balanced data, algorithmic parameters could be chosen according to minimum difference between cross-validated performance measures on each class. An example of this approach is provided in Chapter 6.

Of the two balancing measures implemented above, it is anticipated that algorithmic weighting would 'scale-up' best in order to treat problems of vast imbalance, e.g. the 2001 KDD Cup Thrombin binding QSAR data, which has a minority / minority class size ratio of approximately 98% / 2%. In such a circumstance, in which 42 of 1909 compounds form the minority class, it may be difficult to provide a representative sub-sample of 42 majority class examples. Fortunately, this situation is unlikely to occur when creating SPC relationships for lead optimisation.

Table 4.6 demonstrates that the strategic reduction of majority class examples with the Mahalanobis distance has the potential to reduce performance imbalance to an extent similar to that of algorithmic weighting or balanced regularisation. Upon further consideration of how the majority class is reduced in order to balance the training data, further options for such a reduction become apparent. First, the Mahalanobis distance itself is, like PCA, not particularly robust to outlying examples in a body of data. That is, outlying examples may have undue influence upon the calculated shape and location of a body of data, which are used subsequently to assess the extent of the outlying examples themselves. Robust variants of the Mahalanobis distance have been introduced and used to good effect for a variety of outlier detection tasks [Franklin and Brodeur, 1997; Hardin and Rocke, 1999] and it is envisaged that the application of robust Mahalanobis methods may improve further upon the results introduced here. More simply, the present method could be made slightly more robust by re-calculating all remaining majority class distances upon the removal of each example.

A further limitation of the Mahalanobis distance is that it is only calculable when a body of data presents a shape within its input space, i.e. it has more examples than descriptive attributes. As discussed in section 4.1, this may not be the case in the example poor, attribute rich scenario of *in silico* drug discovery. Accordingly, one must consider methods, preferably those already employed for drug discovery, by which to achieve a similar typification of the majority class. The Kennard and Stone [1969] (K&S) partitioning method, employed here to partition the data into training and test sets, may also be employed to

typify a majority class. A potential weakness of strategic majority class reduction via Mahalanobis distance rejection is that the area of input space covered by the majority class is likely to 'shrink' as it is reduced, i.e. the shape of the majority class will change as examples are removed. The K&S method samples evenly across occupied chemical space and could be employed here to remove majority class examples evenly across the space that they occupy. By doing so, remaining majority class examples would be more likely to represent the original majority class distribution. Table 4.7 displays algorithmic performance upon data reduced by K&S reduction in the same manner as described above for Mahalanobis reduction (only SVM results shown in the interests of concision). As previously, rows of table 4.7 shown in bold font refer to an increase in balanced accuracy of at least marginal significance against the original results of table 4.2. Minority class accuracies accompanied by an asterisk are also increased significantly.

| Data | Kernel | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB | **LSVM** | $C = 1$ | **0.811** | **0.786** | **0.832** | **0.741*** |
| | **QSVM** | $C = 10$ | **0.836** | **0.822** | **0.848** | **0.796*** |
| | RBF SVM | $\sigma =$ 'H', $C = 1$ | 0.815 | 0.763 | 0.859 | 0.667 |
| | | | | | | |
| P-gp | **LSVM** | $C = 1$ | **0.855** | **0.854** | **0.865** | **0.844*** |
| | QSVM | $C = 10$ | 0.841 | 0.847 | 0.757 | 0.938 |
| | **RBF SVM** | $\sigma =$ **'J'**, $C = 1$ | **0.841** | **0.847** | **0.757** | **0.938*** |
| | | | | | | |
| Tox | **LSVM** | $C = 1$ | **0.937** | **0.867** | **0.968** | **0.767*** |
| | **QSVM** | $C = 10$ | **0.872** | **0.834** | **0.890** | **0.778*** |
| | **RBF SVM** | $\sigma =$ **'H'**, $C = 1$ | **0.929** | **0.849** | **0.964** | **0.733*** |
| | | | | | | |
| Bio | **LSVM** | $C = 1$ | **0.768** | **0.713** | **0.789** | **0.636*** |
| | QSVM | $C = 10$ | 0.689 | 0.641 | 0.707 | 0.576* |
| | RBF SVM | $\sigma =$ 'J-M', $C = 10$ | 0.730 | 0.691 | 0.745 | 0.636* |
| | | | | | | |
| PrB | LSVM | $C = 10$ | 0.686 | 0.697 | 0.662 | 0.731* |
| | QSVM | $C = 1$ | 0.707 | 0.729 | 0.662 | 0.795* |
| | **RBF SVM** | $\sigma =$ **'H'**, $C = 10$ | **0.716** | **0.714** | **0.722** | **0.705*** |

**Table 4.7:** K&S-Reduction: SVM Performance on GSK Test Data

The results of table 4.7 display a slightly, but not significantly, stronger balanced accuracy than when the training data is Mahalanobis reduced. Mahalanobis reduction samples non-outlying examples from the majority class, whereas K&S reduction samples evenly across the majority class. The former is more likely to alter the shape and location of the majority class in relation to the minority class when providing a reduced but improved sample, whereas the latter thins the majority class under the assumption that its original shape and location are representative of the separation problem. The K&S reduction approach is also demonstrated in section 6.3 of Chapter 6.

Support vector domain description (SVDD) [Tax and Duin, 1999] employs the same

quadratic optimisation framework of SVM learning in order to enclose a body of data with a hypersphere and locate its centre. SVDD may also be performed in feature space, via an appropriate kernel expansion, in order to provide non-spherical enclosures. Regularisation may be applied and controlled in a manner similar to that of the SVM method in order to make the enclosure robust in the manner of robust Mahalanobis assessment, described above, with the drawback that, unlike supervised learning on labelled data, there is no natural stopping point for such regularisation. SVDD could be employed to typify a majority data class, but would involve considerations of free-parameter selection that are not present when using Mahalanobis distance or K&S sampling. Although not investigated during the course of this work, the application of SVDD to wider aspects of combinatorial library design would make an interesting subject for future work.

A balancing approach that is not pursued here is the concept of 'up-sampling' the minority training class to the same size as the majority class. This may involve the addition of similar examples from a body of unlabelled data, or the addition of virtual examples interpolated from existing minority class data. Both methods depend upon information provided by the minority class data regarding its shape and location. A one-class classifier, trained on the minority class, could be employed to conscript unlabelled examples to balance the data. The SVDD method may be a suitable technique with which to build such a one-class classifier.

Finally, the flexible formulation of SVMs may offer a more elegant and integrated performance balancing method than that described above. Two developments of the SVM algorithm present interesting considerations in the context of data typification. First, the work of Lee and Mangasarian [2001] describes a reduced support vector machine formulation (*RSVM*) that a) is formulated as a linear, rather than quadratic, programming problem and b) is thus able to optimise over a rectangular kernel matrix composed of kernel inner products between all training data and a randomly sampled subset of the training data. Trials in the literature demonstrate that the original performance, i.e. that of an SVM trained conventionally using all training data, is maintained when using as little as 1% of the training data in the rectangular kernel matrix. An interesting piece of future work would be to investigate whether the majority class reduction strategies, proposed here, may be incorporated into the RSVM framework in order to achieve performance balance without the outright removal of majority class examples. Another approach that may not employ all available training data is the use of an SVM within a query learning framework [Campbell et al., 2000; Warmuth et al., 2002]. As described in Chapter 2, § 2.2.6, query learning involves the use of a supervised machine learning algorithm that, after training on a small subset of labelled data, requests class labels of unlabelled examples close to the decision boundary and, once labelled, adds them to the training data in order to improve generalisation performance on unlabelled data from the same distribution in future iterations of the process. SVMs within a query learning framework have been applied to lead generation data previously [Mathieson, 2001; Warmuth et al., 2002] and their application to these imbalanced SPC problems may invoke a situation in which the

classifier is not dominated by a majority class. For example, an SVM could be trained initially on a small, balanced sub-sample of the available training data and request labels from the remaining data in order to improve generalisation on both classes of data.

It has been demonstrated that the strategic removal of majority class training examples, in order to provide a class-balanced training set, produces results similar to those of conventional algorithmic weighting and regularisation when applied to five ADMET SPC data sets. Several avenues for further improvement are apparent and form a body of future work that appears worthwhile to pursue. Regardless of the method employed, methods to counter the effects of imbalanced class populations within the training data should be implemented when using supervised machine learning to create *in silico* SPC relationships for lead optimisation. Two suggested methods of strategic majority class reduction, Mahalanobis and K&S sampling, typify the majority data class via the removal of outliers and an even sampling strategy respectively. An ideal situation would combine the two strategies, in order to obtain a sample representative of the original distribution of majority class examples that is of higher quality than the original sample. In addition, a milder treatment of the minority class, to remove outliers in lesser number than from the majority class may also improve matters. Chapter 5 considers an adaptation to the SVM algorithm towards this purpose.

This chapter contributes to the research hypotheses stated in Chapter 1 in the following form. The initial comparison (section 4.1) reveals challenges posed by learning complex relationships between molecular structure and ADMET properties from small collections of labelled data. Further to arguments presented during Chapter 2, the results of the comparison suggest that the application would indeed benefit from the introduction of any technique able to improve performance by overcoming the challenges presented. As to whether the SVM algorithm represents such a technique, the results of both comparisons suggest that, although it does not offer an immediate and significant increase in predictive accuracy over extant techniques when employed in a form that is not adapted for the application, it does perform competitively against them. In agreement with a growing body of published work on this subject, SVMs display potential for the successful treatment of this application. Further to the comparisons, it appears that the success of a newly introduced technique may rely on how well it retains its natural predictive ability when adapted to overcome the challenges of the domain. The remaining chapters of this work investigate two such adaptations.

# Chapter 5

# Neighbourhood Influence on Support Vector Machine Classification

The challenge of structure-property classification is described in Chapter 2 and demonstrated in Chapter 4. Earlier comparisons have displayed both the strengths and limitations of several machine learning techniques, particularly SVMs, when applied to data drawn from the lead optimisation stage of the drug discovery process. The predictive success displayed overall is tempered by the effort required to distinguish a minority, or under-represented, class of data from a larger class in a manner that maintains good balanced generalisation performance.

Section 4.2 of Chapter 4 describes approaches with which to balance performance on class imbalanced training data via weighted training algorithms and a strategy for the removal of majority class examples respectively. It would be convenient if the SVM algorithm could behave similarly regardless of regularisation and without the outright removal of training examples. Accordingly, this chapter assesses a method of kernel matrix construction that may focus the algorithm upon relevant regions of the training data and, thereby, may invite balanced generalisation performance. Subsequent discussion regards the potential of this approach, and suggested domain-relevant adaptations to it, for SPC analysis.

It is impractical to weight kernel matrix contributions according to training data class labels, because the class of new examples is unknown and, therefore, weighted contribution to the kernel function is not available upon classification. One may, however, focus the kernel matrix upon particular areas of the training set, under the assumption that these areas will best define the general separation of the data according to the target attribute. Concentration upon relatively well-populated areas of input space, which will possess subsequently a greater influence upon classifier structure during training, is another form of data typification (cf. section 4.2 of Chapter 4).

## 5.1 Neighbourhood Weighting the SVM Kernel Matrix

Several groups have introduced methods of kernel combination in order to treat contributions of attribute subsets in a distinct manner prior to their combination in order to represent inter-example similarity. One of the earliest examples is that of Schölkopf et al. [1998], which incorporates the effects of local feature correlation into an image classification task. Similarity between pixellated image vectors is assessed via a function on local image similarities, rather than a function on all possible combinations of pixel values as would a standard polynomial kernel ($K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^p$, cf. § 2.2.3). Each image vector pair is combined to form a pixel-wise product of the two images, e.g. a third image in which pixel values result from the product of corresponding pixels in the original pair. A local area around each pixel provides attribute (pixel) values that contribute to a dot-product between each pixel pair. The local dot-products for each pixel pair are raised to a power $p1$ and represent local image correlations. The local correlation values for each pixel are subsequently summed and raised to a power $p2$ in order to provide a global image correlation. Thus, the final kernel is of order $p1 \cdot p2$ but does not contain all possible pixel-pair contributions - only those relevant to the comparison of two images. A sizeable reduction in test error is observed in comparison to a standard polynomial kernel on a hand-written character recognition task. Several other works have combined partial kernel contributions in order to involve prior-knowledge regarding local attribute contributions, including [Brailovsky et al., 1999; Lodhi et al., 2000; Zien et al., 2000; Vert, 2002] and [Rätsch et al., 2006], some of which are discussed further in Chapter 6.

Alongside similar work on combined attribute kernel functions, Brailovsky et al. [1999] introduce a method of SVM kernel combination that permits the influence of local patterns in input space upon transformation to feature space. The method attempts to introduce the benefits of 'lazy' classifiers (such as $k$-NN), which examine a small subset of training data local to each test example when predicting its class, with a global classification boundary that may not require a search across all training examples upon each classification. Thus, the method differs from previous work on combined kernels by considering the combination of example, rather than attribute, subsets.

Regardless of the function chosen to provide a 'base' kernel, the output for each training set pair is weighted according to a measure of their locality in input space. Locality, in this scenario, should be distinguished from similarity. Similarity assesses the likeness of an example pair, e.g. the similarity between two molecular structures. Locality assesses the number of examples located in the immediate surroundings of the examples between which similarity is assessed. Thus, the kernel function mapping of an example pair is weighted according to the context in which their similarity is assessed. From [Brailovsky et al., 1999], consider a hard window located in input space with centre $\mathbf{w}_0$ and an associated binary function that attributes zero weight to input space examples that lie outside a threshold distance (or dissimilarity) $\theta_0$ from $\mathbf{w}_0$ and unit weight to examples inside the threshold

distance

$$h(|\mathbf{x} - \mathbf{w}_0|) = \begin{cases} 1 & \text{if } |\mathbf{x} - \mathbf{w}_0| \leq \theta_0 \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

A kernel function, applied to points in the windowed input space, will provide non-zero output only for examples that inhabit the window

$$K_{\mathbf{w}_0}(\mathbf{x}, \mathbf{z}) = K(\mathbf{x}, \mathbf{z})\, h(|\mathbf{x} - \mathbf{w}_0|)\, h(|\mathbf{z} - \mathbf{w}_0|) \tag{5.2}$$

Centres $\mathbf{w}_1, \ldots, \mathbf{w}_k$, distributed across the training data so that functions $h(|\mathbf{x} - \mathbf{w}_i|)$ cover the domain, provide partial kernel contributions, $K_{\mathbf{w}_i}(\mathbf{x}, \mathbf{z})$, that may be summed

$$K^*(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{k} K_{\mathbf{w}_i}(\mathbf{x}, \mathbf{z}) = K(\mathbf{x}, \mathbf{z})\, n_{\mathbf{x},\mathbf{z}} \tag{5.3}$$

where $n_{\mathbf{x},\mathbf{z}}$ is the number of windows that include both $\mathbf{x}$ and $\mathbf{z}$.

Data thus weighted ensures that well-sampled regions have greater influence than poorly sampled regions of the data. There remains the possibility, however, that some test examples may not inhabit any of the windows determined by the training data and, hence, will provide no relevant evidence for classification in the locally-weighted feature space. On one hand, it is rather sensible that such examples are not classified on the basis that they are not reflected by the training data from which the classifier is constructed. On the other, it would be preferable to classify such examples and present their classification alongside a warning flag. In order to do so, an extra window with infinite distance threshold, $\theta_{k+1} = \infty$, is introduced and the corresponding partial kernel, $K_{\mathbf{w}_{k+1}}(\mathbf{x}, \mathbf{z}) = K(\mathbf{x}, \mathbf{z})$, is added to the sum of partial kernels for an example pair $\mathbf{x}$ and $\mathbf{z}$.

The original work [Brailovsky et al., 1999] suggests several strategies for the placement of influence windows across the training data, including hard (i.e. non-overlapping) and soft (overlapping) windows, arranged in a grid, located on the centres provided by $k$-means data clustering, or placed at each training example so as to enclose a certain number of nearest-neighbours. The competitive performance of a $k$-NN classifier in Chapter 4 suggests the soft window, nearest-neighbourhood method. In brief, each training example provides the centre of a spherical window, $\mathbf{w}_i = \mathbf{x}_i$, of radius large enough to contain the $k$ nearest examples (neighbours) to the centre, i.e. $\theta_i = |\mathbf{x}_i^{(k)} - \mathbf{x}_i|$, where $\mathbf{x}_i^{(k)}$ is the $k^{\text{th}}$ nearest example to $\mathbf{x}_i$.

The combination of partial base kernel applications, in order to reflect relevant patterns within the training data, appears to fulfill the aim outlined at the end of the previous chapter, i.e. to typify the training data without the outright removal of examples. There exist situations, however, in which this method may not perform the task as required. For example, the technique is applied under the assumption that there exists sufficient, sufficiently well-sampled, labelled data *of both classes*. This appears likely for the BBB and Toxicity data sets, but unlikely for the Bioavailability and PrB data sets. When minority class examples

are scattered amongst a larger majority class, concentrating the training algorithm upon the densest areas of the data is unlikely to assist recognition of the minority class and may, in fact, lead to further performance imbalance by strengthening the majority class further. Another consideration is that, from the sum of partial kernels in equation 5.3, the neighbourhood method represents a form of classifier combination. A large amount of work on classifier combination [Breiman, 1994; Freund, 1995; Evgeniou, 2000; Skurichina, 2001] suggests that weak classifiers, i.e. those that make errors on the training data, are those best affected by combination over different subsets of the training data. To investigate this, both linear and quadratic SVM kernels are used as base kernels when assessing the neighbourhood method below. RBF kernels are not assessed, for reasons including their strength on the training data, the consideration that separate RBF widths may be required for each partial kernel matrix and their use of an extra free parameter during training.

## 5.2 Results

The neighbourhood kernel method is applied to the GSK data as before (Chapter 4) and the results are reported in the same tabular format. Table 5.1 displays the performance of regular and neighbourhood SVM classifiers, with both linear and quadratic base kernels, when used to classify the GSK data test partitions. Linear and quadratic kernels are denoted 'LSVM' and 'QSVM' respectively and neighbourhood kernels are prefixed with 'NN-'. The neighbourhood of $k$-NN windowing is assessed over the range $k = \{3, 5, 7, 9, 11, 13, 15\}$.

A one-tailed, weighted McNemar test of marginal homogeneity was applied to compare the performance (balanced accuracy) of neighbourhood weighted SVMs against the performance of the original algorithm (Chapter 4, section 4.1). An unweighted one-tailed McNemar test was applied to compare classifier performance on majority and minority classes in isolation. As before, a difference in performance is deemed *significant* at the 95% level ($p < 0.05$), *marginally significant* if between 90-95% levels ($0.05 \leq p < 0.10$) and not significant otherwise ($p \geq 0.10$). Results are summarised below.

BBB: The neighbourhood-weighted linear SVM (NN-LSVM) displays increased balanced accuracy against the unweighted LSVM, but the increase is not significant. Majority class accuracy is significantly reduced ($p = 0.001$) but remains $> 80\%$, whereas minority class accuracy is significantly increased ($p = 0.032$). Conversely, the neighbourhood-weighted quadratic SVM (NN-QSVM) displays a significant decrease in balanced accuracy against its unweighted counterpart.

P-gp: The neighbourhood-weighted linear SVM (NN-LSVM) displays significantly higher balanced accuracy than the unweighted LSVM, courtesy of a significant increase in minority class accuracy and an insignificant increase in majority class accuracy. The neighbourhood-weighted quadratic SVM (NN-QSVM) also displays higher balanced accuracy than its counterpart, but, because of the already strong quadratic SVM performance on this data, the increase is not significant.

| Data | Algorithm | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB | LSVM | $C = 1$ | 0.849 | 0.732 | 0.946 | 0.519 |
| | NN-LSVM | $C = 1, n = 3$ | 0.798 | 0.758 | 0.832 | 0.685 |
| | QSVM | $C = 10$ | 0.832 | 0.761 | 0.891 | 0.630 |
| | NN-QSVM | $C = 1, n = 5$ | 0.790 | 0.714 | 0.853 | 0.574 |
| | | | | | | |
| P-gp | LSVM | $C = 1$ | 0.768 | 0.761 | 0.865 | 0.656 |
| | NN-LSVM | $C = 100, n = 5$ | 0.899 | 0.897 | 0.919 | 0.875 |
| | QSVM | $C = 1$ | 0.870 | 0.870 | 0.865 | 0.875 |
| | NN-QSVM | $C = 100, n = 13$ | 0.899 | 0.897 | 0.919 | 0.875 |
| | | | | | | |
| Tox | LSVM | $C = 10$ | 0.939 | 0.814 | 0.994 | 0.633 |
| | NN-LSVM | $C = 100, n = 11$ | 0.925 | 0.828 | 0.968 | 0.689 |
| | QSVM | $C = 1$ | 0.930 | 0.800 | 0.988 | 0.611 |
| | NN-QSVM | $C = 100, n = 11$ | 0.930 | 0.831 | 0.974 | 0.689 |
| | | | | | | |
| Bio | LSVM | $C = 10$ | 0.822 | 0.603 | 0.904 | 0.303 |
| | NN-LSVM | $C = 100, n = 9$ | 0.801 | 0.655 | 0.856 | 0.455 |
| | QSVM | $C = 1$ | 0.834 | 0.649 | 0.904 | 0.394 |
| | NN-QSVM | $C = 100, n = 13$ | 0.809 | 0.635 | 0.875 | 0.394 |
| | | | | | | |
| PrB | LSVM | $C = 100$ | 0.734 | 0.659 | 0.894 | 0.423 |
| | NN-LSVM | $C = 1, n = 5$ | 0.716 | 0.673 | 0.808 | 0.539 |
| | QSVM | $C = 1$ | 0.756 | 0.715 | 0.841 | 0.590 |
| | NN-QSVM | $C = 100, n = 13$ | 0.699 | 0.682 | 0.735 | 0.628 |

**Table 5.1:** $k$NN-SVM vs. SVM Performance on GSK Test Data

Tox: The neighbourhood-weighted linear SVM (NN-LSVM) shows no significant performance increase against balanced accuracy of an unweighted LSVM. The minority class accuracy is increased slightly, but not significantly, at the expense of a significant decrease in majority class accuracy. The neighbourhood-weighted quadratic SVM (NN-QSVM) displays significantly higher balanced accuracy against an unweighted QSVM. A significant increase in minority class accuracy is accompanied by a decrease in majority class accuracy of marginal significance.

Bio: The neighbourhood-weighted linear SVM (NN-LSVM) displays increased balanced accuracy against unweighted LSVM performance with marginal significance ($p = 0.057$), courtesy of a marginally significant increase in minority class performance. The neighbourhood-weighted quadratic SVM (NN-QSVM) displays an insignificant decrease in balanced accuracy against its counterpart.

PrB: The neighbourhood-weighted linear SVM (NN-LSVM) displays an insignificant increase in balanced accuracy against an unweighted LSVM. A significant increase in minority class accuracy is accompanied by a significant decrease in majority class accuracy, although majority class accuracy remains $> 80\%$. The neighbourhood-weighted quadratic SVM (NN-QSVM) displays an insignificant decrease in balanced accuracy against its counterpart.

## 5.3 Discussion

Any positive effect of weighting linear and quadratic SVM kernels with the method of Brailovsky et al. [1999] appears dependent upon the nature of the data to which the algorithm is applied and the strength of the original algorithm itself. For example, a linear base kernel displays increased balanced accuracy when neighbourhood weighted on all five data sets (twice with at least marginal significance) and improves minority class accuracy with at least marginal significance while retaining majority class accuracy $> 80\%$ in four of those cases. Conversely, a quadratic base kernel significantly increases balanced accuracy once (Toxicity) and significantly decreases balanced accuracy once also (BBB), although good performance on the P-gp data does not represent a significant increase on the already strong performance of an unweighted quadratic SVM. This appears contrary to results of the original work, in which a three-degree polynomial kernel improved performance on an image recognition task with $k = 5$. The task involved was larger than the tasks treated here, however, an it may have been the case that an unweighted polynomial SVM did not perform strongly on the training data, as it does here.

In general, the balancing effect observed is smaller in magnitude than that achieved by the explicit balancing techniques of section 4.2, excepting impressive performance on the P-gp data. Performance on the Bioavailability data in particular reflects the caution expressed at the end of section 5.1, regarding neighbourhood weighting on a sample of insufficient size or quality. Both BBB and Bioavailability further reflect potentially negative

results when combining high capacity classifiers. On the BBB data, for example, the linear SVM obtains a margin of separation on the training data after neighbourhood weighting, in much the same manner as do weak classifiers under conventional classifier combination strategies [Skurichina, 2001]. The quadratic SVM already has a margin of separation on the training data and the combination of partial kernel matrices appears to induce further overfitting on both classes of the training data. A potential reason for this on the BBB data is the locality threshold selected by cross-validation ($k = 5$ nearest-neighbours). At such a level, locality tends towards the reinforcement of similarity between training example pairs. The linear SVM ($k = 3$) appears to benefit from this influence, but the quadratic SVM with higher capacity does not. The same effect is less evident on the other data sets, upon which $k$ is set to 11–13 neighbours for the quadratic SVM. In many respects, concentration on multiple small localities of the training data has an effect similar to adding extra information around the examples with greatest window membership to achieve an effect similar to that of ensemble creation methods [Evgeniou, 2000; Skurichina, 2001].

The neighbourhood influence method of Brailovsky et al. [1999] displays some promise for strengthening weak SVM classifier performance on the small sets of data presented. It is in the consideration of how one may improve the method, in terms both of performance and domain-relevance, that its potential becomes apparent. The first potential improvement to the neighbourhood technique, in the presence of small, class imbalanced training sets, is the separate treatment of majority and minority data classes when determining the number, location and reach of windows employed to create partial kernel matrices. It is difficult to consider this in the context of the $k$-NN windowing employed above, because it is not clear how different neighbour ranges would affect the representation of majority and minority classes respectively. The use of clustering procedures to attribute windows to positive and negative classes separately may be a more interpretable method (see below). Alternatively, neighbourhood weighting could be applied to data balanced as suggested in section 4.2 of Chapter 4. Such application would have the potential to represent relevant data patterns in the presence of equally represented classes and, thereby, to focus on locality without an overwhelming majority class presence in overlapping regions. Thus, the application of neighbourhood weighting could typify both classes and correct a potential flaw of the majority reduction method, in which only the majority class is typified.

A weakness of the $k$-NN windowing is that windows are centred about all training data examples. The result of this is that the resulting classifier requires the input of all training data when transforming unlabelled data into feature space for classification. The use of a potentially sparse training data subset to support the decision boundary is a primary benefit of SVM classifiers, especially in circumstances wherein they may be employed to classify a large amount of unlabelled data. The $k$-NN windowing removes this advantage. As an alternative, the original work on neighbourhood kernels [Brailovsky et al., 1999] suggests local windowing using $k$-means clustering alongside the nearest-neighbour approach employed here. In some respects, one might expect such a windowing to replicate the performance of

an RBF network, especially when applied to linear kernel functions (see § 2.2.6, p. 59). A previous comparison of RBF networks against RBF-kernel SVMs [Schölkopf et al., 1997] suggests that the performance advantage of the RBF-SVM over its RBF network counterpart is due to the placement of RBF network centers by $k$-means clustering and, thus, would appear to cast doubt on the usefulness of such an approach. Nevertheless, the use of other clustering methods to window input space may provide interesting avenues of further investigation, especially because only cluster centres, rather than the entire data set, are consulted upon the transformation of unlabelled data for classification, thus retaining the SVM advantage of a sparse solution. The nearest-neighbour method of Jarvis and Patrick [1973], which is used widely as a method of clustering pharmaceutical data [Butina, 1999], may provide a suitable windowing with the additional benefit of doing so via a method that is well-known to practitioners of drug design.

Furthermore, the method of Butina [1999] would offer a more recent technique, designed by a drug design practitioner in order to cluster pharmaceutical data. As described in Chapter 2, § 2.2.6, not only is this method applicable to both real and binary representations of structural and whole-molecular attributes, the excluding spheres employed to define clusters about each centre may be adapted to provide overlapping windows (or left unchanged to provide hard windows) and examples identified as outliers by the clustering may be removed from classifier creation or, in the case of test examples, flagged as examples classified in a region of insufficient training information. The limiting width of excluding spheres, the only free parameter of the algorithm, could be set using heuristics similar to those employed here for setting RBF kernel width (real-valued structural attributes), or set using prior domain knowledge on a suitable representation of whole molecule similarity (cf. Chapter 6). The use of partial kernel construction to involve domain-relevance in SVM classification is suggested as future work by Brailovsky et al. [1999] and this appears one such example.

The BBB data is seen in table 5.1 to respond to neighbourhood weighting to an extent that suggests there is sufficient data available from which to extract typical information regarding both classes. It is also observed that a quadratic base kernel responds to $k$-NN weighting by over-fitting the training data. Accordingly, the BBB data is employed here in a brief trial of two of the above suggestions for improvement. Table 5.2 displays SVM performance on the BBB data when training is influenced by Butina clustering in the manner suggested above. The clustering algorithm is adapted slightly, to allow the excluding spheres of the original formulation to overlap. Kernel contributions are weighted according to the number of spheres shared by an example pair and an infi-window is added in order to treat outlying data. The similarity threshold of the clustering algorithm is set using the 'Jaakkola' width heuristic, described in Chapter 3, § 3.3.1, although a full comparison should set any heuristically determined threshold by cross-validation over a range of multiples (cf. § 4.1.2). Linear and quadratic base kernels are weighted as described and an SVM trained and assessed according to the experimental practice of Chapter 3. Table 5.2 displays generalisation performance in the same format as employed earlier. The unweighted linear and quadratic SVMs are referred to as 'LSVM' and 'QSVM' respectively. Their weighted

variants are referred to as 'BC-LSVM' and 'BC-QSVM' respectively.

| Data | Algorithm | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB | LSVM | $C = 1$ | 0.849 | 0.732 | 0.946 | 0.519 |
| | BC-LSVM | $C = 10$ | 0.807 | 0.777 | 0.832 | 0.722 |
| | QSVM | $C = 10$ | 0.832 | 0.761 | 0.891 | 0.630 |
| | BC-QSVM | $C = 100$ | 0.819 | 0.785 | 0.848 | 0.722 |

**Table 5.2:** BC-SVM vs. SVM Performance on BBB Test Data

The Butina-weighted linear SVM (BC-LSVM) displays a marginally significant ($p = 0.075$) increase in balanced accuracy against the unweighted LSVM. Majority class accuracy is significantly reduced, but remains $> 80\%$, whereas minority class accuracy is significantly increased ($p = 0.002$). The Butina-weighted quadratic SVM (BC-QSVM) displays an insignificant increase in balanced accuracy against the unweighted QSVM. Majority class accuracy is reduced with marginal significance, but remains $> 80\%$, whereas minority class accuracy is increased with marginal significance ($p = 0.090$). This result contrasts sharply with the corresponding NN-QSVM performance in table 5.1. A potential reason for this is that the windows provided by clustering are wider in scope and overlap less than those produced by the $k$-NN method's immediate view of the region surrounding each example. The partial matrices produced by clustering describe patterns directly rather than via accumulation, to the apparent benefit of data typification.

The effect of weighting the SVM kernel matrix according to cluster membership appears to balance performance on the majority and minority data classes to a similar extent as that observed for training data balanced by Mahalanobis reduction (Chapter 4, § 4.2). Table 5.3 displays the effects of combining these two methods, as suggested above. The Butina neighbourhood weighting is applied and assessed using the experimental practice employed for the assessment of Mahalanobis reduction in section 4.2 of Chapter 4.

| Data | Algorithm | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB | LSVM | $C = 1$ | 0.849 | 0.732 | 0.946 | 0.519 |
| | MR-LSVM | $C = 1$ | 0.845 | 0.789 | 0.875 | 0.704 |
| | BC-LSVM | $C = 10$ | 0.807 | 0.777 | 0.832 | 0.722 |
| | **MRBC-LSVM** | $C = 1$ | **0.832** | **0.826** | **0.837** | **0.815** |
| | QSVM | $C = 10$ | 0.832 | 0.761 | 0.891 | 0.630 |
| | MR-QSVM | $C = 1$ | 0.845 | 0.808 | 0.875 | 0.741 |
| | BC-QSVM | $C = 100$ | 0.819 | 0.785 | 0.848 | 0.722 |
| | MRBC-QSVM | $C = 100$ | 0.807 | 0.797 | 0.815 | 0.778 |

**Table 5.3:** SVM and BC-SVM Performance on Mahalanobis-Reduced BBB Data

The effect of combining Mahalanobis data reduction and neighbourhood-weighted SVM kernels displays a promising effect on performance over separate applications of both techniques using both linear and quadratic SVM base kernels. The combined method with linear base kernel displays a significant increase in balanced accuracy over the unweighted

LSVM and records > 80% accuracy on both classes of the BBB data. The combined method with quadratic base kernel displays a significant ($p = 0.029$) increase in minority class accuracy against an unweighted quadratic kernel, but this increase is offset by a significant decrease in majority class accuracy and, thus, the displayed increase in balanced accuracy is not significant. Nevertheless, the class accuracies displayed by the combined method with quadratic base kernel approach those of the combined method with linear base kernel and are more even than those attained by the constituent methods when applied separately.

Locality weighting the SVM kernel matrix displays potential to a) incorporate domain relevance into the SVM framework, and b) reinforce the balancing measures suggested in Chapter 4. Focus upon data patterns during training tends to improve performance balance and judicious choice of clustering algorithm would facilitate integration into current industrial drug discovery practices. One key drawback of such an approach is the introduction of an extra free-parameter during SVM training. Greater attention than is provided by the initial trials above is required in order to optimise, for example, the similarity threshold employed by the Butina clustering algorithm. For example, as suggested for tuning RBF width in § 4.1.2 of Chapter 4, the threshold could be tuned over multiples of a single, heuristically-chosen value. The use of pharmaceutical clustering methods to weight SVM kernel contributions on ADMET property prediction is the subject of work ongoing, which will examine methods of threshold setting more fully.

Further consideration of clustered kernel construction yields several options for further development. For example, in the original formulation of Brailovsky et al. [1999], partial kernels contribute equally to the combined similarity matrix. Weighted combinations of partial kernels, which incorporate measures of partial kernel contribution, may improve performance further. For example, partial kernels could be weighted according to the number of example pairs that they affect, some measure of the expected performance of their contribution, or class heterogeneity of the associated examples in the case of class imbalanced training data. Kernel similarity matrices constructed of partial kernels on attribute subsets have been weighted similarly to good effect in the past. A more sophisticated approach may be provided by optimising a set of weights across partial kernel contributions, as performed recently by Rätsch et al. [2006] who solve an optimised weighting over combined local attribute kernel contributions during SVM training.

The extraction of relevant information from the available labelled data appears a promising approach in circumstances involving small amounts of relatively well-sampled data, but displays weakness in the presence of insufficient labelled data of sufficient sample quality. The idea of using the structure of an associated body of unlabelled data in order to improve classifier inference on a set of labelled data is proposed in several works [Vapnik, 1998; Bennett and Demiriz, 1998; Joachims, 1999; Jaakkola et al., 2000; Campbell et al., 2000] and has an unexplored relevance to this framework of kernel construction. For example, in the simplest case, one may consider clustering on a relevant body of unlabelled data, instead of the training data, and using the windowing extracted to produce partial kernel functions on the training data. The data sets employed thus far do not provide a sufficient

body of relevant unlabelled data for each problem. Nevertheless, the idea of involving data beyond the labelled data available is an inviting one, especially in cases of inadequate sampling. Action to reinforce classifier construction and generalisation should be pursued further in attempts to improve the quality of *in silico* ADMET screens when few examples are available from which to generalise. To date, such work has been pursued in the context of drug discovery by [Mathieson, 2001; Warmuth et al., 2002] and [Weston et al., 2003].

This chapter has introduced the neighbourhood kernel method of Brailovsky et al. [1999] and assessed its application to several small, class imbalanced ADMET classification problems. The original method, which describes training data locality in a similar manner to the $k$-NN algorithm (cf. § 2.2.6, p. 63), displays some good effect in strengthening the unbalanced performance of linear SVMs in particular. It is in the incorporation of a domain-relevant windowing strategy, however, that more promise is shown particularly in combination with a data balancing procedure introduced in the previous chapter. The work presented here is intended to prompt wider research into the application of domain-relevant kernel weighting for problems such as SPC analysis. Much work has been pursued towards the combination of local attribute subsets to represent locality within example vectors via partial kernel combinations, but the neighbourhood weighting of [Brailovsky et al., 1999] has received relatively little attention as a method of performing the same task across a distribution of examples. This may be because powerful methods of classifier combination on sub-samples of training data exist already, in the form of classifier combination algorithms such as Bagging [Breiman, 1994] and Boosting [Freund, 1995], and have been worked successfully into the SVM framework [Rätsch et al., 2000; Evgeniou, 2000]. This application of partial kernel construction was motivated by a desire to extract information related to the generalisable separation of imbalanced training data, rather than the creation an ensemble classifier. Because many of the clusters employed are class homogeneous, it is unclear as to whether this method may be directly related to traditional approaches to classifier combination. Nevertheless, the positive effect of combining partial kernel matrices, particularly upon weak SVM classifiers, suggests that study of the method from the perspective of ensemble creation [Kittler et al., 1998] may yield more specific information regarding how best to refine the technique.

The incorporation of domain relevance into an SVM kernel framework via the selection of suitable windowing methods warrants further investigation along with adaptations that are specifically related to classifier combination, such as the optimisation of partial kernel weights. The use of windowing strategies to guide a transductive approach to SPC relationship analysis represents a separate strand of research, albeit one that also warrants consideration when attempting to generalise well beyond small collections of labelled ADMET data.

# Chapter 6

# Tanimoto Kernels for Support Vector Machine Classification

The methods described in this chapter were developed independently of the contemporary and recently published works of Chen et al. [2006] and Fröhlich et al. [2005, 2006]. This unpublished work compliments these citations and, in conjunction with them, makes a powerful statement regarding the future role of machine learning within the drug discovery process.

As discussed during Chapter 2, not all *in silico* molecular representations available to the SPC analyst are real-valued. Sparse, binary representations of molecular structure have found great favour in combinatorial library construction, because measures of similarity between binary strings are rapidly calculated and have been demonstrated particularly effective in the assessment of structural dissimilarity [Drewry and Young, 1999].

Daylight fingerprints are introduced in § 2.1.5 of Chapter 2 as 1024-bit binary strings that provide an abstract representation of molecular structure via the combination of 4 / 5 bit encodings of the molecular sub-structures that comprise a single compound. Chemical space, thus described, may be considered a 1024-dimensional hypercube with compounds located on its vertices [Kondor and Lafferty, 2002]. An advantage of such a representation is to provide chemical space with a set of discrete co-ordinates, at which compounds are located uniquely according to their sub-molecular constituents.

The high cardinality and sparse nature of Daylight strings (typically 1024 bits, only 20% of which are switched on) have, until recently, rendered them a choice less obvious than explicit, real-valued descriptors for treatment by supervised machine learning. Rather, fingerprints were used to assess the similarity of unlabelled compounds in large collections (such as those found after HTS or in the large combinatorial libraries of lead generation) to a reference structure (or structures) of known activity against a molecular target. Clustering and outlier detection methods work well on binary data, because similarity between examples is rapidly calculated, and their treatment of Daylight data is used widely to increase diversity in large combinatorial libraries [Holliday et al., 2002; Daylight, 2006].

The measurement of similarity between Daylight strings requires consideration of the

mapping that they provide between molecular structure and binary representation. As described in § 2.1.5, the Daylight map of structure to chemical space is abstract. That is, sub-molecular fragments are encoded prior to their combination to form a single string. The encoding does not represent sub-structural fragments explicitly. If the encoding of a particular fragment is present within a fingerprint, the fragment itself is highly likely to form part of the larger molecule that the fingerprint describes. It is of note that the information encoded is 2D. Fragment presence within the molecule is represented, but fragment geometry and location are not.

Direct assessment of similarity between Daylight strings serves to approximate the explicit comparison of their constituent fragment encodings. The more bits mutually on in two fingerprints, the greater the likelihood that the compounds represented share sub-structural fragments and, therefore, are structurally similar. The abstract nature of the encoding limits the relevance of contiguous sub-strings, except as a combinatorial approximation to the overall similarity between strings. In addition, the attributes (the elements of each string) have no explicit meaning, i.e. they are not explicitly descriptive attributes, except as contributions to the assessment of similarity between strings.

For example, the Euclidean distance records bits mutually off in two strings as contributing towards their similarity. Daylight strings typically have 20% of their bits switched on. Two strings, each with 20% bits on but none overlapping would appear 60% similar according to the Euclidean distance despite having no structural similarities. Measures designed to obviate such a situation are available and widely used. The Tanimoto similarity [Butina, 1999; Holliday et al., 2002] normalises the number of bits mutually on, $c$, in two strings, $i$ and $j$, by the number of bits on in both strings, $(a + b)$, less the double-counting of bits mutually on,

$$T(i,j) = \frac{c}{a+b-c} \quad .$$

As discussed in Chapter 2, the lack of a universal molecular representation suitable for machine learning is a barrier to successful *in silico* drug discovery. The machine learning treatment of structure-activity relationships on molecular encodings is growing in popularity for lead generation [Hert et al., 2006]. Were it practical to employ supervised machine learning to learn smaller and more complex ADMET SPC relationships on some rapidly calculable standardised representation, it is conceivable that the *in silico* discovery process could be performed automatically and in a single molecular representation from combinatorial chemistry through to the latter stages of lead optimisation (the 2D stages). Standardised, used in this context, represents a molecular description as having relevant standard form rather than being universally descriptive.

SVMs have been applied to sparse binary fingerprint representations of compound structure, in order to mine HTS output and large combinatorial libraries at the lead generation stage of the drug design process [Warmuth et al., 2002; Weston et al., 2003; Wilton et al., 2006]. An SVM with high-dimensional polynomial kernel (degree = 5) is applied to a variety of fingerprint representations by [Wilton et al., 2006], to create structure-activity

relationships from a large collection of agrochemical compounds arising from pesticide discovery. The SVM is compared to another method, binary kernel discrimination (BKD), which employs a smoothed similarity function to assess the similarity of unlabelled compounds to known compounds in a form of 'lazy' classification (cf. p. 63) and proves more able than BKD when employed to predict compound activity from a subset of labelled compounds. This observation is tempered, however, by the nature of the problem treated - a one-against-all classification scenario involving five activity classes (and, hence, a 20% / 80% class imbalance), and results in which the proportion of true positives in each class prediction is 30–60% (performance is best on extreme categories). A recent advance in the BKD method [Chen et al., 2006] relates to the work of this chapter and consideration of the technique provides useful background information.

From [Wilton et al., 2006], the standard BKD method employs a similarity function,

$$k_\lambda(i,j) = \left[\lambda^{m-d_{ij}}(1-\lambda)^{d_{ij}}\right]^{\left(\frac{p}{m}\right)} \tag{6.1}$$

where $\lambda$ is a smoothing free-parameter in the range $(0.5, 1.0)$, $d_{ij}$ is the squared Euclidean distance between two compounds, $i$ and $j$, $m$ is the string length and $p$ is a user-defined scaling parameter. The similarity function is employed in kernel density estimators [Bishop, 1995a] to estimate the likelihood that an unclassified compound, $j$, is active against the same target as a reference compound, $i$. One should note the use of $k$ to reference the density function instead of $K$, which is used throughout this thesis to refer to a valid SVM inner-product kernel. In the presence of a labelled set of active compounds, $A$, and another of inactive compounds, $B$, a scoring function may be employed to rank the activity or otherwise of an unlabelled compound $j$,

$$S_{BKD}(j) = \frac{\sum_{i\in A} k_\lambda(i,j)}{\sum_{i\in B} k_\lambda(i,j)}$$

which, for binary prediction, may be represented as the decision function

$$f(j) = sgn(S_{BKD} - 1) \quad \text{where} \quad \text{sgn}(\mu) = \begin{cases} +1 & \mu \geq 0 \\ -1 & \text{otherwise,} \end{cases}$$

or

$$f(j) = sgn\left(\sum_{i\in\{A,B\}} y_i k_\lambda(i,j)\right) \quad \text{where} \quad y_i \in \{+1,-1\} \text{ denotes activity.}$$

By contrast, the SVM decision function

$$f(\mathbf{z}) = sgn\left(\sum_{i=1}^{nsv} \alpha_i y_i K(\mathbf{x}_i, \mathbf{z}) + w_0\right)$$

weights kernel contributions to the decision function by the optimised weights, $\alpha_i$, and only

involves similarities between an unlabelled example and training examples with non-zero weights (the SV points) that support an optimum separating hyperplane. As described in [Wilton et al., 2006], the BKD kernel similarity function is dominated by labelled compounds most similar to the unlabelled compound as $\lambda$ increases. BKD is a 'lazy' learning classifier and, in essence, an appropriate choice of $\lambda$ leads BKD to employ a relevant subset of nearest neighbours upon which to classify each unlabelled compound.

Chen et al. [2006] replace the Euclidean dissimilarity measure above with the complement of several, popular measures for the assessment of similarity between bit-strings. The Tanimoto dissimilarity (the complement of the Tanimoto similarity described above) is found to provide significantly better performance in the prediction and retention of active compounds from a large collection and represents the incorporation of domain-relevance. Thus far, SVMs and BKD have been applied to scenarios wherein the efficient discovery of active lead compounds is paramount. The ADMET classification tasks presented during early lead optimisation place a different emphasis on the correct classification of both target property classes.

SVMs have been applied successfully to treat sparse, discrete representations in other domains. Joachims [1998b] applied SVMs to a sparse representation of document text and demonstrated effective kernel transformation of sparse data to a feature space in which examples are represented by inner-product pairs (i.e. a function of their similarity). The challenges posed to supervised machine learning by Daylight fingerprints (sparsity, abstract representation, no contiguous sequence structure, etc.) are similar in nature to those described in [Joachims, 1998b] and successfully overcome via the application of SVMs. In light of this, it is informative to consider the field of research that has arisen around the design of appropriate kernel transformations of discrete structures and to describe a contemporary example.

Further to the application of standard Euclidean kernel functions to discrete data, the independently developed techniques of Watkins [1999] and Haussler [1999] first introduced kernel transformations specific to the treatment of discrete sequences. The representation of a joint probability distribution as valid kernel transformation to feature space enables the treatment of examples described by different amounts of discrete information (different example vector lengths) in a continuous feature space. Such kernel functions are able to assess the similarities between examples of discrete data, such as text strings or biological sequences, via the construction of kernel transformations that represent example similarity in feature space without the explicit construction of an input space. The string kernels suggested in [Watkins, 1999] are applied to a problem of text document classification by Lodhi et al. [2000], in which similarities between documents, represented as sequences of characters from a discrete alphabet, are recursively calculated to identify the mutual presence of non-contiguous sub-strings, weighted by the length of their occurrence within the original sequence.

A second example, and arguably the most elegant practical application of SVMs to discrete data, is provided in the work of Vert [2002], in which the string kernel methodology of

[Watkins, 1999] is applied to a binary representation of evolutionary pathways in phylogenetic trees in order to classify conserved gene function. This work is notable for its custom approach to both the domain information and the kernel function applied to it. Further work by Kondor and Lafferty [2002] introduces a family of 'diffusion' kernels for the generic kernel transformation of discrete, or graph, structures directly to feature space. Of interest is the definition of a diffusion kernel that treats the hypercube representation of binary data,

$$K(\mathbf{x}, \mathbf{z}) = [\tanh(\lambda)]^{d_{\mathbf{x},\mathbf{z}}}$$

where $\lambda$ is a free-parameter and $d_{\mathbf{x},\mathbf{z}}$ is the Hamming distance between two binary vectors. This function appears similar in nature to the BKD kernel function (cf. equation 6.1 above). As suggested by their name, diffusion kernels approximate a continuous similarity function that diffuses from a point source in a discrete input space according to the exponential heat equation of classical physics. The standard Euclidean RBF kernel is shown to represent a diffusion kernel in the limit as a discrete space becomes continuous. The work of [Jaakkola et al., 1999] demonstrates further that, if a domain-relevant similarity measure may be represented by dot-products between examples, it provides a valid kernel with which to learn in a domain-relevant feature space.

Whilst many early applications of kernels that map directly from graph or sequence to feature space were applied to document classification and biological sequence data, work by Fröhlich et al. [2005, 2006] introduces a similar transformation of graph molecular structure to feature space in order to treat classification problems of SPC analysis. The approach centres on the calculation of an optimal assignment between two molecular structures, which includes specific structural and chemical properties of the atoms and their immediate neighbourhoods that comprise each structure. In order to obviate the classic QSAR analysis problem of having no universal best set of molecular descriptors that work well for all problems (cf. § 2.1.5) and the extensive feature selection that results, a kernel function is defined between graph molecular representations instead of vectors of explicit descriptive attributes. Each molecule is represented as a graph, with atoms at the nodes and bonds represented by edges between nodes. Nodes and edges are labelled with information relevant to the problems, e.g. atomic properties and structural memberships. Thus, the graph of a molecule provides detailed information regarding its topology without consideration of the whole-molecular relevance of the descriptors employed. The intuition behind the use of this representation is that similarity between molecules depends upon matching constituent sub-structures. This is especially relevant to problems of compound binding, but one must be aware of the case in which molecules with different sub-structural arrangements offer the same, or similar, values of a less specific target property, e.g. bioavailability.

The optimal assignment between two molecular structures is defined as the sum of edge weights in a weighted bipartite graph, optimised so as to link each atom in one structure to the most similar atom in another. Each atom in one structure can only be linked to one in the other structure, with inter-atomic similarities providing edge weights. The maximised bipartite weighting problem may be solved in $\left(O(n^3)\right)$ time where $n$ is the size (number

of atoms) of the largest structure under consideration. The similarity between atoms, i.e. the edge weights of the bipartite graph, is calculated according to the following combined kernel function,

$$k_{\text{nei}}(a, a') := k_{\text{atom}}(a, a') + R_0(a, a') + \sum_{l+1}^{L} \gamma(l) R_l(a, a') \quad .$$

Similarity between atoms, $a$ and $a'$, is composed of similarities between the atoms themselves, their direct nearest-neighbours and their indirect nearest neighbours up to a predetermined locality limit ($L$). Standard RBF kernel functions, $k_{\text{atom}}(a, a')$ and $k_{\text{bond}}(a, a')$ are applied to assess similarity between the numeric atom and bond descriptors employed to label the nodes and edges of two molecular graphs under consideration. The atomic kernel function $k_{\text{atom}}(a, a')$ is applied to assess similarity between atoms $a$ and $a'$. In order to assess similarity between direct nearest-neighbours of $a$ and $a'$, the product of $k_{\text{atom}}$ and $k_{\text{bond}}$ is calculated for all combinations of atoms directly linked (by bonds) to $a$ and $a'$ and used to label the edges of a further weighted bipartite graph between the two sets of atomic neighbours. The sum of optimised bipartite graph weights provides the similarity, $R_0(a, a')$, between the direct neighbours of atoms $a$ and $a'$. Finally, a third term calculates the locality-weighted mean of $R_0(a_i, a'_j)$ for all direct and indirect neighbours of $a$ and $a'$ in order to capture similarities across a wider topological distance.

This representation may be reduced by replacing atomic information at the nodes of a graph molecular structure with similar information regarding sub-structural fragments. The similarity between two sub-structural fragments in separate molecular structures is found via optimisation of a weighted bipartite graph between them as before. This summary of the optimal assignment kernel method describes the representation of inter-molecular similarity via a valid kernel transformation between graph molecular representation directly into a relevant feature space, within which inner product pairs may be acted upon by kernel-based learning methods. It is highly recommended that the reader consults the original work for a more detailed and reasoned description of the method, the complexity of which is clear.

It is true that, by assessing molecular similarity directly, explicit feature selection across many whole-molecular structural descriptors is obviated. Nevertheless, classical feature selection is replaced by a raft of algorithmic free parameters and remaining feature selection for node and edge labeling. For example, widths must be set for both atom and bond RBF kernels, $k_{\text{atom}}(a, a')$ and $k_{\text{bond}}(a, a')$, as must the locality path length limit, $L$, and the locality weighting, $\gamma(l)$. Furthermore, the numeric information employed to describe atoms and bonds at the nodes and edges of a molecular graph must also be chosen (or selected) from the wide range of available atomic information [Kier, 1995]. Another consideration is of the time taken to calculate the kernel transformation of a large amount of unlabelled data prior to classification. Despite this, the method offers a comprehensive assessment of sub-structural similarity that is well worked into the kernel framework. The optimal assignment kernel is demonstrated to perform better than a standard RBF kernel when applied to a selection of ADME classification and regression tasks (BBB, Human Intestinal Absorption

and Bioavailability), albeit that the standard kernel is applied to a full atomic description (DESC) that involves thousands of structural descriptors. A wrapper-based feature selection [Fröhlich et al., 2004] is employed to improve standard kernel performance, but is not particularly successful. Interestingly, the reduced graph representation performs similarly to the full optimal assignment kernel in all cases and both perform similarly to a small selection of whole-molecular descriptors of known relevance to the HIA and BBB problems respectively. The combination of optimal assignment kernel output on graph molecular structures and standard RBF kernel output on problem-specific molecular descriptors shows potential to improve performance further in much the same manner as the data fusion methods described in [Hert et al., 2006].

This contemporary work merits citation and the above description because it represents one of the first uses of kernel design to provide a relevant feature space within which SPC relationships may be constructed by SVMs. As described earlier in this section, such approaches have been prevalent in applications of text processing or the comparison of biological sequences and structures. The concept that present methods of SPC analysis may be improved by departing the use of explicit molecular descriptors towards the creation of feature spaces that represent relevant molecular similarities is wholeheartedly subscribed to by this thesis, for reasons to become apparent.

The work presented in this chapter introduces the formulation of the Tanimoto similarity coefficient as a combination of dot-products between binary vectors in order to enable its use for SVM learning. The ADMET SPC data sets of previous chapters are converted to Daylight binary representation and a performance comparison assesses SVM performance with standard and Tanimoto kernels on real-valued and Daylight representations. The use of Tanimoto similarities for SVM classification was developed independently and prior to publication of both the optimal assignment method and recent developments involving the use of Tanimoto similarity for BKD density estimation, but may be observed to compliment both.

For example, under the Daylight representational schema, the reduced graph optimal assignment kernel function may be seen to perform a similar function to normalised pattern matching between the substructural elements that comprise a complete Daylight molecular fingerprint. Rather than combine similarities between sub-structural elements, the Daylight encoding represents the presence of sub-molecular fragments within a particular structure and the Tanimoto similarity assesses similarity after combination of sub-structural elements. Thereby, reduced-graph optimal assignment kernels are approximated by a domain-relevant similarity measure applied to a widely-used encoding of molecular structure.

Formulation of the Tanimoto similarity co-efficient as a combination of dot-products between binary strings enables its use for SVM learning. Furthermore, the recent introduction of similar domain-relevant procedures at the lead generation stage of the discovery process suggests that successful application to the less specific ADMET SPC properties of lead optimisation may invite a unified treatment of several stages of the drug discovery process. The remainder of this chapter outlines the representation of Tanimoto similarity as a com-

bination of Mercer kernels and subsequently applies the resulting domain-relevant SVMs to Daylight fingerprint representations of the five ADMET classification tasks described in Chapter 3 and employed for the comparisons of Chapters 4 & 5.

## 6.1 Tanimoto Similarity Kernels for Binary Data

The Tanimoto similarity coefficient between two compounds represented by binary strings takes the following form:

$$T(\mathbf{x}, \mathbf{z}) = \frac{c}{a + b - c}$$

where $a$ is the number of bits switched on in string $\mathbf{x}$, $b$ is the number of bits switched on in string $\mathbf{z}$, and $c$ is the number of overlapping bits switched on between strings $\mathbf{x}$ and $\mathbf{z}$. This normalised similarity is bounded in the range [0,1] and the conditions required for use of the Tanimoto similarity coefficient are that the data are represented as strings of discrete binary information and that the strings are of the same length.

Binary data is a special, discrete, case of the continuous data described in earlier chapters. Thus, the Tanimoto similarity co-efficient may be represented in a number of interesting ways. For example, because the data consists solely of zeros and ones, the number of bits switched on in a string $\mathbf{x}$, containing $m$ bits in total, can be:

$$a = \sum_{i=1}^{m} x_i \quad \text{or} \quad a = \mathbf{x}^T\mathbf{x}$$

and the same may be shown for bits switched on in another binary string, $\mathbf{z}$, of the same length as $\mathbf{x}$. Bits mutually on in strings $\mathbf{x}$ and $\mathbf{z}$ may be represented in a similar manner:

$$c = \sum_{i=1}^{m} x_i.z_i \quad \text{or} \quad c = \mathbf{x}^T\mathbf{z}$$

which describes a linear Mercer kernel between two examples $\mathbf{x}$ and $\mathbf{z}$

$$K_{\text{LIN}}(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T\mathbf{z}$$

Thus, the Tanimoto similarity coefficient between two compounds represented as binary strings may be represented as a combination of linear Mercer kernels:

$$K_{\text{TAN}}(\mathbf{x}, \mathbf{z}) = \frac{\mathbf{x}^T\mathbf{z}}{\mathbf{x}^T\mathbf{x} + \mathbf{z}^T\mathbf{z} - \mathbf{x}^T\mathbf{z}} \tag{6.2}$$

The Tanimoto similarity is symmetric and a matrix of Tanimoto similarity values, calculated over a finite set of binary strings, was found empirically during this work to be positive and semi-definite (the eigenvalues of the matrix are $\geq 0$) in all cases encountered. Under the conjecture that this should always be the case, the Tanimoto similarity may be referred to as a Tanimoto kernel for use in SVM classification. The new kernel is domain-relevant and

has no free parameters.

To provide some additional context, a well-known assessment of dissimilarity between binary strings, the Hamming distance, may also be represented by a function on the linear Mercer kernel:

$$D_{\text{HAM}}(\mathbf{x}, \mathbf{z}) = m - (\mathbf{x}^T\mathbf{z})$$

where $m$ represents the string length [Kondor and Lafferty, 2002]. The Hamming distance counts differences between the strings, whereas the Tanimoto kernel represents the number of bits mutually on, normalised by the number of bits that *could* be mutually on. The Tanimoto dissimilarity coefficient between two binary vectors $\mathbf{x}$ and $\mathbf{z}$ is simply:

$$T'(\mathbf{x}, \mathbf{z}) = 1 - T(\mathbf{x}, \mathbf{z}) = \frac{\mathbf{x}^T\mathbf{x} + \mathbf{z}^T\mathbf{z} - 2\mathbf{x}^T\mathbf{z}}{\mathbf{x}^T\mathbf{x} + \mathbf{z}^T\mathbf{z} - \mathbf{x}^T\mathbf{z}} \tag{6.3}$$

and Tanimoto kernel output falls away linearly with increasing Tanimoto dissimilarity.

The standard RBF kernel function of two compounds $\mathbf{x}$ and $\mathbf{z}$ is frequently represented as:

$$K_{RBF}(\mathbf{x}, \mathbf{z}) = \exp -(||\mathbf{x} - \mathbf{z}||^2/2\sigma^2)$$

As for the Tanimoto kernel, RBF kernel output assesses similarity and is bounded in the range [0,1]. When data are represented as strings of discrete, binary, attribute values, the RBF kernel may also be represented as:

$$K_{RBF}(\mathbf{x}, \mathbf{z}) = \exp -((\mathbf{x}^T\mathbf{x} + \mathbf{z}^T\mathbf{z} - 2\mathbf{x}^T\mathbf{z})/2\sigma^2) \tag{6.4}$$

The representation of Euclidean distance in feature space is treated in § 3.4 of [Cristianini and Shawe-Taylor, 2000] and confirms the above for vectors of continuous attributes as well. The top term of the fraction in equation 6.4 represents the unnormalised Tanimoto dissimilarity (cf. equation 6.3), which is scaled by a constant parameter, $\sigma$, that controls the range of dissimilarity over which it acts. Substitution of the Euclidean distance in equation 6.4 with the Tanimoto dissimilarity provides a relevant measure of dissimilarity that falls away exponentially (as does the BKD kernel density function when $0.5 < \lambda < 1.0$):

$$K_{\text{T-R}}(\mathbf{x}, \mathbf{z}) = \exp -(T'(\mathbf{x}, \mathbf{z})/\beta) \tag{6.5}$$

The scale constant, $\beta$, in equation (6.5) may be set via an estimate of generalisation error over a range of values, the use of similar heuristics to those described in § 3.3.1, or, potentially, via the use of data description algorithms to assess the typical separation of compounds within the data. $\beta$ appears to perform a similar function to the smoothing parameter, $\lambda$, in the BKD kernel (cf. equation 6.1, p. 120), but does so within the range of Tanimoto dissimilarity. The use of a familiar, bounded pharmaceutical dissimilarity measure results in a more accessible kernel function, the free parameter of which may be related

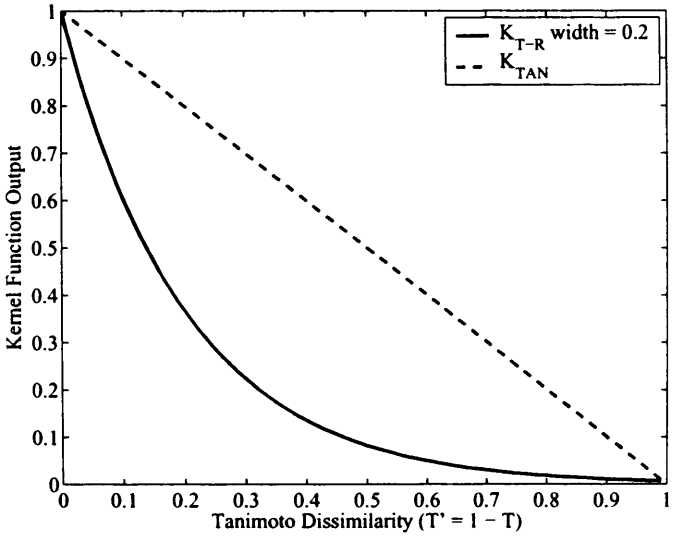


**Figure 6.1:** Output of Tanimoto and Tanimoto-RBF Kernel Functions

simply to pharmaceutical prior knowledge by a non-expert user. Figure 6.1 displays kernel contributions made by the Tanimoto kernel and an exponential Tanimoto kernel with high locality ($\beta$ set to dissimilarity of 0.2) over the range of Tanimoto dissimilarity.

## 6.2 Results

In order to compare SVM performance on both real-valued and Daylight molecular representations, the training and test partitions of Chapter 4 were converted directly to Daylight representation. Prior to a description of the trials performed, it may prove useful to examine the resulting Daylight data sets in greater detail.

Upon consideration of the Tanimoto kernel to represent inter-molecular similarity between Daylight strings, it is apparent that one scenario in which it may provide an advantage over, for example, a linear kernel function is when considering strings that have different bit contents (i.e. the number of bits switched 'on'). As described in section 6.1, the Tanimoto similarity provides a form of normalisation that is intended to make similarity values more comparable for vectors with different bit contents. One might, therefore, expect any positive effect made by use of the Tanimoto kernel to coincide with data sets in which the strings vary widely in bit content. Accordingly, to elucidate the diversity of bit content in each of the GSK Daylight data sets and also in the data classes that comprise each set, the range of bit content in each data set is displayed alongside the 1st, 10th, 50th, 90th and 99th percentiles in table 6.1. Table 6.2 provides similar information for the individual classes of each data set. The two data classes are denoted positive and negative according to the descriptions given in Chapter 3 (section 3.1).

| | Range and Percentiles of Bit Content in Daylight Data | | | | | | |
|---|---|---|---|---|---|---|---|
| **Data** | **Min.** | **P1** | **P10** | **P50** | **P90** | **P99** | **Max.** |
| BBB | 7.0 | 12.0 | 113.0 | 221.0 | 312.0 | 425.5 | 448.0 |
| P-gp | 30.0 | 44.1 | 152.5 | 259.5 | 437.4 | 612.6 | 632.0 |
| Tox | 27.0 | 36.3 | 68.0 | 124.0 | 260.9 | 391.5 | 493.0 |
| Bio | 16.0 | 37.2 | 102.6 | 196.0 | 345.2 | 488.7 | 584.0 |
| PrB | 29.0 | 50.9 | 109.4 | 205.0 | 399.0 | 501.9 | 538.0 |

**Table 6.1:** Range and Selected Percentiles of Bit Content in GSK Daylight Data Sets.

| | Percentiles of Bit Content in Daylight Data | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Positive Class | | | | | Negative Class | | | | |
| **Data** | **P1** | **P10** | **P50** | **P90** | **P99** | **P1** | **P10** | **P50** | **P90** | **P99** |
| BBB | 12.0 | 87.8 | 216.0 | 287.6 | 429.7 | 79.0 | 155.6 | 237.5 | 368.7 | 420.0 |
| P-gp | 31.4 | 105.4 | 238.0 | 341.2 | 436.4 | 118.6 | 176.0 | 339.0 | 504.0 | 625.6 |
| Tox | 75.0 | 123.0 | 239.0 | 324.0 | 427.7 | 33.0 | 64.0 | 113.0 | 204.0 | 383.0 |
| Bio | 37.7 | 100.0 | 189.0 | 322.6 | 442.1 | 39.9 | 110.9 | 225.5 | 427.4 | 561.6 |
| PrB | 47.9 | 105.0 | 194.0 | 421.8 | 504.5 | 84.4 | 124.4 | 223.0 | 344.0 | 446.3 |

**Table 6.2:** Selected Percentiles of Bit Content in GSK Daylight Data Classes.

To further elucidate the contents of tables 6.1 & 6.2, the function *ksdensity*, of the *stats* package for the *Matlab* statistical programming language [Mathworks, 2002], was used to provide a kernel density estimate [Parzen, 1962] that reflects the distribution of bit contents in the classes of each data set. The function was employed with default parameters (the aim here was to provide a simple visual impression of bit content distribution) at 100 equally spaced points across the bit content ranges shown in table 6.1 and its output is displayed in the sub-plots of figure 6.2. Each sub-plot reflects bit content distribution in the positive (solid line) and negative (dash-dotted line) data classes of the corresponding data set (cf. table 6.2). The dotted line is the sum of the positive and negative class density estimates and provides an impression of bit content distribution across all strings in the corresponding data set (cf. table 6.1).

The diversity of bit content apparent in tables 6.1 & 6.2 and in figure 6.2 further suggests the potential suitability of Tanimoto kernels for creating SPC relationships on these data sets. Furthermore, and before proceeding to details of the performance comparison, some interesting patterns in the bit content distributions of individual data sets merit brief comment.

- The Acute Toxicity and P-gp data sets display some disparity in the bit content distributions of positive and negative data classes. The toxicity data in particular suggests that the distribution of negative class (low toxicity) bit contents is narrower and located lower in the range than that of the positive class (high toxicity);

- the positive class of the BBB data contains a visible (cf. figure 6.2) subset of strings with particularly low bit content. The positive class represents molecules that pass

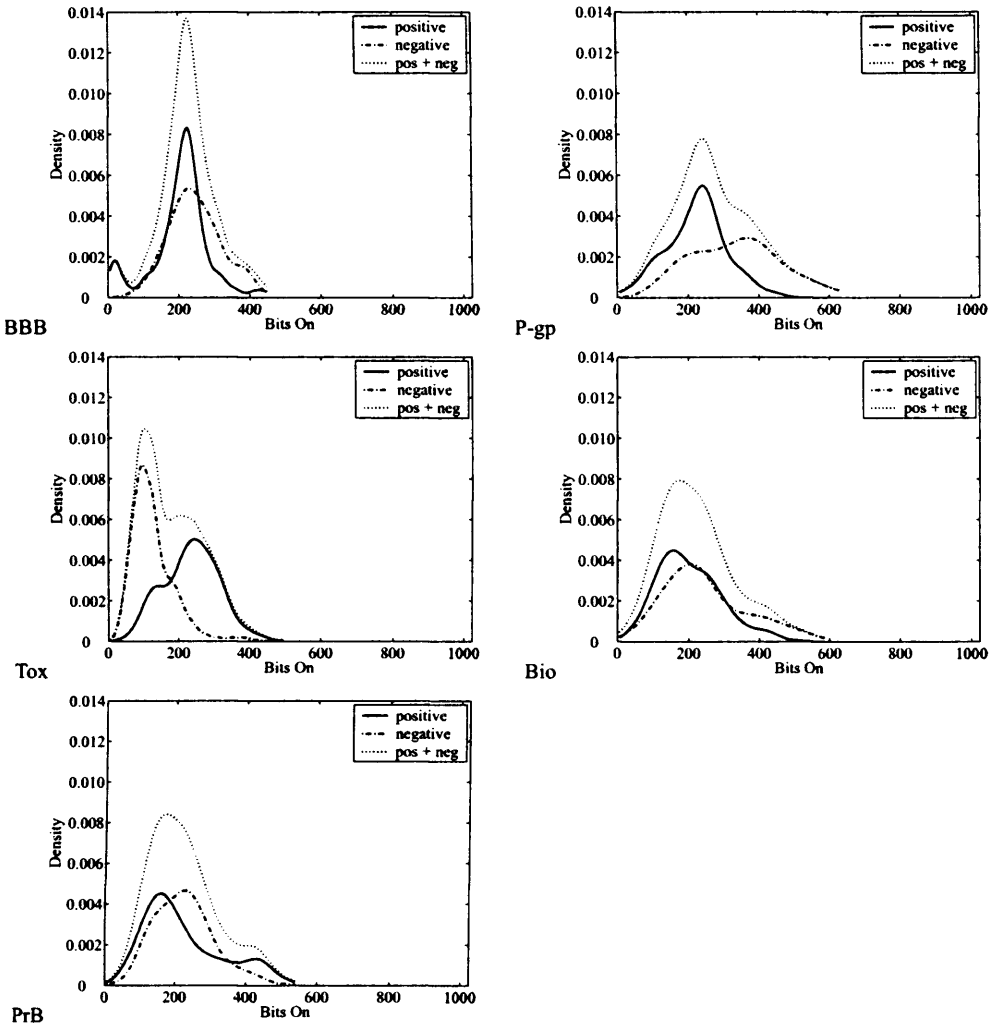


**Figure 6.2:** Estimated Density Plots Reflect Bit Content Distributions in the Positive (solid line) and Negative (dash-dotted line) Classes of the GSK Daylight Data Sets. The sum of positive and negative class densities (dotted line) is employed to represent bit content distribution over entire data sets.

through the blood-brain barrier. Low bit content corresponds to few molecular components and may suggest small molecules better able to pass through the membrane;

- bit contents of positive and negative classes in the Bioavailability and Protein Binding data appear similarly distributed across the ranges observed. Both sets exhibit a visible (cf. figure 6.2) 'tail' of high bit content (potentially large / complex molecules) in both data classes.

The performance comparison between SVM kernels when applied to the GSK Daylight data and corresponding performance on the real-valued data of Chapter 4 was performed as follows. SVMs with linear, Tanimoto, RBF and Tanimoto-RBF kernels were applied to the Daylight data training partitions and the classifiers obtained were assessed on the corresponding test partitions. Results are displayed in table 6.3. SVM performance with linear and RBF kernels on the real-valued data of previous chapters is displayed alongside the results on each Daylight representation for comparison. The classifier with highest

balanced accuracy on each ADMET problem is displayed in bold. To provide a benchmark and also to assess the effect of similarity measures upon lazy learners, a $k$-NN classifier was applied to the same data using both Euclidean distance and Tanimoto dissimilarity to select nearest-neighbours. The results of this comparison are shown in table 6.4 and all results are summarised at the start of the discussion section.

## 6.3 Discussion

From the results presented in table 6.3, it is apparent that:

- linear and RBF SVM kernels yield similar results on Daylight representation as they do on real-valued data. The linear SVM kernel exhibits higher balanced accuracy on Daylight representation on two of the five data sets, once significantly (BBB). The RBF SVM kernel exhibits higher balanced accuracy on Daylight representation on four of the five data sets, but the differences involved are not statistically significant;
- the parameter-free Tanimoto kernel outperforms its linear counterpart on four of the five data sets, twice involving a significant increase in balanced accuracy (BBB & PrB). The performance difference observed between these kernels may relate in part to the examination of bit content diversity in section 6.2;
- the Tanimoto kernel provides significantly similar balanced accuracy to both RBF and Tanimoto-RBF kernels, which perform similarly, on all data sets; and
- all kernels are affected by class imbalance and data paucity on Daylight data, in the same manner as previously observed on real-valued data but to a varying extent.

In addition, the results of table 6.4 suggest that:

- the performance of both Euclidean and Tanimoto $k$-NN is stronger on the Daylight BBB data than on the Volsurf BBB data;
- Tanimoto $k$-NN is stronger than Euclidean $k$-NN on the Daylight Toxicity data, but not significantly stronger than Euclidean $k$-NN on the Volsurf Toxicity data;
- Tanimoto $k$-NN performance is competitive against SVM performance on BBB and Toxicity Daylight data;
- $k$-NN does not perform well on the P-gp Daylight data. The same behaviour as observed for SVMs is observed for $k$-NN on the Bioavailability and Protein Binding data sets; and
- a low number of nearest-neighbours is employed for classification in the majority of applications.

| Data | Kernel | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB Real | Linear | $C = 1$ | 0.849 | 0.732 | 0.946 | 0.519 |
| | RBF | $\sigma$ ='J-M', $C = 10$ | 0.870 | 0.805 | 0.924 | 0.685 |
| BBB Day. | Linear | $C = 1$ | 0.840 | 0.786 | 0.886 | 0.685 |
| | Tanimoto | $C = 100$ | 0.866 | 0.828 | 0.897 | 0.759 |
| | RBF | $\sigma$ ='Mn', $C = 100$ | 0.874 | 0.834 | 0.908 | 0.759 |
| | **Tan. RBF** | $s$ =**'J-M'**, $C = 100$ | **0.878** | **0.836** | **0.913** | **0.759** |
| | | | | | | |
| P-gp Real | **Linear** | $C = 1$ | **0.768** | **0.761** | **0.865** | **0.656** |
| | RBF | $\sigma$ = 'J', $C = 10$ | 0.739 | 0.738 | 0.757 | 0.719 |
| P-gp Day. | Linear | $C = 100$ | 0.725 | 0.718 | 0.811 | 0.625 |
| | Tanimoto | $C = 100$ | 0.725 | 0.718 | 0.811 | 0.625 |
| | RBF | $\sigma$ = 'Md', $C = 100$ | 0.725 | 0.718 | 0.811 | 0.625 |
| | Tan. RBF | $s$ = 'J', $C = 100$ | 0.725 | 0.718 | 0.811 | 0.625 |
| | | | | | | |
| Tox Real | Linear | $C = 10$ | 0.939 | 0.814 | 0.994 | 0.633 |
| | RBF | $\sigma$ = 'H', $C = 100$ | 0.932 | 0.801 | 0.990 | 0.611 |
| Tox Day. | Linear | $C = 1$ | 0.886 | 0.805 | 0.922 | 0.689 |
| | Tanimoto | $C = 10$ | 0.917 | 0.814 | 0.962 | 0.667 |
| | **RBF** | $\sigma$ = **'J-M'**, $C = 10$ | **0.918** | **0.815** | **0.964** | **0.667** |
| | Tan. RBF | $s$ = 'J-M', $C = 10$ | 0.920 | 0.807 | 0.970 | 0.644 |
| | | | | | | |
| Bio Real | Linear | $C = 10$ | 0.822 | 0.603 | 0.904 | 0.303 |
| | RBF | $\sigma$ = 'J', $C = 10$ | 0.867 | 0.655 | 0.947 | 0.364 |
| Bio Day. | Linear | $C = 100$ | 0.826 | 0.644 | 0.894 | 0.394 |
| | Tanimoto | $C = 100$ | 0.842 | 0.654 | 0.914 | 0.394 |
| | **RBF** | $\sigma$ = **'J-M'**, $C = 100$ | **0.847** | **0.656** | **0.918** | **0.394** |
| | Tan. RBF | $s$ = 'J', $C = 10$ | 0.859 | 0.638 | 0.942 | 0.333 |
| | | | | | | |
| PrB Real | Linear | $C = 100$ | 0.734 | 0.659 | 0.894 | 0.423 |
| | RBF | $\sigma$ = 'H', $C = 10$ | 0.716 | 0.667 | 0.821 | 0.513 |
| PrB Day. | Linear | $C = 10$ | 0.664 | 0.643 | 0.709 | 0.577 |
| | Tanimoto | $C = 100$ | 0.707 | 0.688 | 0.748 | 0.628 |
| | RBF | $\sigma$ = 'J-M', $C = 10$ | 0.703 | 0.676 | 0.762 | 0.590 |
| | **Tan. RBF** | $s$ = **'J'**, $C = 100$ | **0.725** | **0.689** | **0.801** | **0.577** |

**Table 6.3:** SVM Kernel Performance on GSK Daylight Test Data

| Data | Metric | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| BBB Real | Euclidean | $k = 1$ | 0.857 | 0.796 | 0.908 | 0.685 |
| BBB Day. | Euclidean | $k = 1$ | 0.866 | 0.828 | 0.897 | 0.759 |
| **BBB Day.** | **Tanimoto** | $k = 3$ | **0.845** | **0.834** | **0.853** | **0.815** |
| | | | | | | |
| **P-gp Real** | **Euclidean** | $k = 15$ | **0.826** | **0.817** | **0.946** | **0.688** |
| P-gp Day. | Euclidean | $k = 15$ | 0.696 | 0.708 | 0.541 | 0.875 |
| P-gp Day. | Tanimoto | $k = 5$ | 0.638 | 0.639 | 0.622 | 0.656 |
| | | | | | | |
| Tox Real | Euclidean | $k = 3$ | 0.920 | 0.812 | 0.968 | 0.656 |
| Tox Day. | Euclidean | $k = 1$ | 0.886 | 0.783 | 0.932 | 0.633 |
| **Tox Day.** | **Tanimoto** | $k = 3$ | **0.912** | **0.825** | **0.950** | **0.700** |
| | | | | | | |
| Bio Real | Euclidean | $k = 1$ | 0.830 | 0.634 | 0.904 | 0.364 |
| **Bio Day.** | **Euclidean** | $k = 1$ | **0.817** | **0.652** | **0.880** | **0.424** |
| Bio Day. | Tanimoto | $k = 1$ | 0.784 | 0.633 | 0.841 | 0.424 |
| | | | | | | |
| Prb Real | Euclidean | $k = 3$ | 0.694 | 0.654 | 0.782 | 0.526 |
| **Prb Day.** | **Euclidean** | $k = 1$ | **0.712** | **0.661** | **0.821** | **0.500** |
| Prb Day. | Tanimoto | $k = 3$ | 0.694 | 0.644 | 0.801 | 0.487 |

**Table 6.4:** $k$-NN Performance on GSK Daylight Test Data

The first observation from tables 6.3 & 6.4 signifies that learning ADMET structure-property relationships from small collections of compounds represented by Daylight fingerprints is a feasible alternative to learning on sets of explicit, real-valued whole-molecular descriptors. An SVM with Tanimoto kernel is competitive against standard kernels and provides a domain-specific, non-linear kernel for pharmaceutical classification that does not require a free parameter value to control range or expression. The Tanimoto-RBF kernel is also competitive, although it provides no significant performance advantage over the Tanimoto kernel on the data employed here at the cost of a free parameter. Further comparisons on larger data-sets may provide more information regarding the extent to which Tanimoto similarity kernels (and other kernels designed upon the same principle) may benefit the contemporary drug discovery process.

The one problem upon which learning on Daylight fingerprints performs worse than learning on real-valued representation, regardless of the kernel function employed, is the classification of P-gp binding. An immediate conclusion is that the small P-gp training partition (69 examples) produces too sparse a coverage of the Daylight input space hypercube to produce good generalisation performance. A further consideration is that the target property is not fully described by Daylight information, which is limited to describing the presence or otherwise of molecular sub-structures. The P-gp data encountered during Chapters 4 & 5 employs five real-valued Abraham descriptors [Zhao et al., 2003] of known relevance to the problem, which are observed to provide higher levels of generalisation performance. Combination of the Tanimoto kernel similarity measure on Daylight data with an RBF kernel

similarity on Abraham descriptors may have the potential to improve performance on this ADMET problem. As described during the introduction of this chapter, similar measures are also suggested in [Fröhlich et al., 2006] and the fusion of different similarity scores in order to improve virtual library screening is also introduced by Hert et al. [2006]. Table 6.5 displays P-gp prediction performance of an RBF kernel on real-valued descriptors, a Tanimoto kernel on Daylight data and combination of both kernel outputs. RBF width was set using the same heuristic as selected previously for the real-valued data (Chapter 4, table 4.2) and the regularisation parameter for all SVMs set via stratified cross-validation as in previous comparisons.

| Data | Kernel | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|---|
| P-gp Real | RBF | $\sigma =$ 'J', $C = 10$ | 0.739 | 0.738 | 0.757 | 0.719 |
| P-gp Day. | Tanimoto | $C = 100$ | 0.725 | 0.718 | 0.811 | 0.625 |
| **P-gp Both** | **Combined** | $C = 100$ | **0.870** | **0.870** | **0.865** | **0.875** |

**Table 6.5:** Combined RBF and Tanimoto Kernels on Abraham and Daylight P-gp Data

Performance of combined kernels is significantly increased against their separate applications to real-valued and Daylight data respectively. As discussed in § 4.1.2, however, the RBF-SVM is capable of good performance on real-valued P-gp data, but was hindered by the regularisation parameter selected by cross-validation. Here, it is more likely that similarity assessed on Daylight data reinforces similarities on the real-valued data and induces selection of a higher regularisation parameter than that selected on the real-valued data alone, thereby raising performance. Hence, this small example may provide an over-stated demonstration of the effects of kernel-based data fusion. Kernel fusion on real-valued and Daylight representations of the Toxicity and Protein Binding data sets displays a mild increase in performance over the best of the individual results (unreported), but the real-valued information involved is less specific than that of the P-gp data and is, thus, less suited to fusion. A full assessment of any benefit to be gained by fusing kernel-based similarity scores on real-valued and Daylight data, possibly involving a parameter that controls the relative contributions of the respective representations, represents future work.

Class performance imbalance is again apparent, especially on Bioavailability and Protein Binding data, despite learning on a different molecular representation with different kernel functions. This appears to confirm conclusions drawn in Chapters 4 & 5, i.e. that the minority class sample is of neither sufficient size nor quality for a generalisable distinction to be drawn between it and the majority class. Balancing via strategic removal of majority class examples was introduced in section 4.2 and shown, on the BBB and Toxicity data in particular, to increase balanced accuracy of many classifiers toward acceptable levels of generalisation on both data classes. The Mahalanobis reduction is not available for use on Daylight data, because string length outweighs the number of examples available. The use of Kennard & Stone sampling was suggested as an alternative and, here, is applied to the Daylight represented ADMET problems of section 6.2. An SVM with Tanimoto kernel is applied to K&S balanced Daylight data sets, using the same experimental practice as

in previous comparisons. The sole difference is the selection of regularisation parameter not by cross-validated balanced accuracy, but by the minimum difference between separate cross-validated accuracies for each class, so as to prefer classifiers with level majority and minority class accuracies over outright balanced accuracy (cf. discussion in § 4.2.2).

| Data | Parameters | Overall | Balanced | Majority | Minority |
|---|---|---|---|---|---|
| BBB | $C = 10$ | 0.807 | 0.816 | 0.799 | 0.833 |
| P-gp | $C = 100$ | 0.783 | 0.785 | 0.757 | 0.813 |
| Tox | $C = 1$ | 0.830 | 0.831 | 0.829 | 0.833 |
| Bio | $C = 100$ | 0.668 | 0.680 | 0.664 | 0.697 |
| PrB | $C = 100$ | 0.651 | 0.692 | 0.563 | 0.821 |

**Table 6.6:** Tanimoto Kernel Performance on K&S Reduced GSK Daylight Data

Balancing the data works well for BBB and Toxicity data sets, producing generalisation accuracy above, or very close to, 80% on both classes of both problems. Performance is improved on the minority class of the P-gp data at the expense of the majority class, but, as observed in table 6.5 and during Chapter 5, other methods are available that produce better balanced accuracy on the P-gp data and which do not require the removal of training examples. Poor results on the Bioavailability and Protein binding data suggest that Daylight data may be more fragile to majority reduction that real-valued representations of lower cardinality (cf. table 4.7). In these circumstances, it may be worth the extra tuning required either to remove fewer majority class examples or to enforce asymmetric regularisation (cf. Chapter 4, section 4.2). Balancing these data sets by majority reduction does not overcome the problems described in § 4.2.2 of Chapter 4. A further consideration is that the real-valued representations of both the Bioavailability and Protein Binding data are based on fragment calculations, as are their Daylight representations. It may be the case, therefore, that a balanced accuracy in the region of 0.700 is the best that may be expected without the incorporation of additional information, e.g. 3D structure or a greater number of molecular property descriptors, and that Daylight fingerprints provide no more information to these problems that do their original representations. The generic nature of both properties may be responsible for limiting the effect of a purely 2D approach. For example, 60% accuracy of Bioavailability prediction is reported as an example in [van de Waterbeemd, 2003] and the prediction of Bioavailability using optimal assignment kernels [Fröhlich et al., 2006] results in a prediction error $> 30\%$ and is the only problem upon which optimal assignment kernels do not improve markedly upon Euclidean RBF kernels applied to thousands of structural descriptors.

The lazy $k$-NN classifier performs competitively against SVMs, especially when using Tanimoto dissimilarity to assess a small number of nearest-neighbours on the BBB and Toxicity data. The small data sets employed here do not provide conclusive evidence that one method is better than the other and it would be interesting to observe the performance of SVM and lazy classification methods on larger data sets. The SVM holds theoretical advantages over lazy classification, such as the optimised placement of a margin hyperplane

supported by a subset of the training data, and may generalise better to a wider sample of test data than is employed here.

The discovery that SVM learning upon Daylight molecular representation is able to produce effective SPC relationships prompts consideration of further work. The amount of known compounds available on which to build SPC classifiers is expected to grow significantly over the coming years. As chemical space represented by Daylight fingerprints is filled with larger populations of training data, it is expected that the separation problems treated here will become better described. The development of Tanimoto similarity kernels may be viewed, therefore, as having greater potential importance than displayed by the results of this chapter.

Future work of immediate interest includes the addition of Tanimoto SVMs to the recent comparison of Chen et al. [2006], who evaluated BKD performance when using several familiar fingerprint similarity scoring functions. An SVM-related method that appears well-suited to the description and retention of a single class of data is the one-class SVDD classifier [Tax and Duin, 1999, 2004], mentioned earlier in Chapter 4 § 4.2.2. The use of a margin-based one-class classifier with domain-relevant kernel function to locate and define a body of active reference compounds in chemical space would provide an interesting approach to the identification or similarity-ranking of further structures. The addition of SVM and SVDD classifiers with Tanimoto-based kernel functions to the taxing multi-class classification problem treated by a standard polynomial SVM and Euclidean BKD in [Wilton et al., 2006] would provide a further test of this approach.

Another comparison could comprise the application of Tanimoto kernel SVMs to the Daylight representation of data used to assess optimal assignment kernels in [Fröhlich et al., 2006], were the data employed for the original work available for conversion to Daylight fingerprints. Wider Tanimoto kernel assessments are also apparent, including their application to binary structural keys, rather than the abstract representation employed here. Application to more descriptive structural keys would provide more information than the 2D structural information of Daylight fingerprints, which may be necessary in order to model generic ADMET target properties. The concept that valid kernels may be constructed in order to represent domain-relevant similarities also invites the appraisal of kernels designed to represent other specialised pharmaceutical similarity measures, e.g. the chemical environment code introduced by Xu and Yang [1998] for the assessment of similarity between molecular NMR spectra and shown to be competitive against the Tanimoto similarity.

A theme of future work described in previous chapters is the use of semi-supervised, or transductive, methods to improve generalisation via the incorporation of unlabelled data beyond that of the small labelled collections employed traditionally for ADMET SPC analysis. For example, the application of domain-relevant kernels to structural fingerprints within a query learning framework [Campbell et al., 2000; Warmuth et al., 2002] would represent a particularly interesting approach. Moreover, the use of a single representation of molecular structure, such as Daylight fingerprints, enables the interaction of classification tasks from

different areas of the drug discovery process.

Recent advances in the field of machine learning describe the concept of inductive transfer [Wu and Dietterich, 2004; Marx et al., 2005; Rosenstein et al., 2005], which involves the use of data from tasks related to the primary classification objective in order to strengthen generalisation performance. For example, the classification of individual leaf silhouettes is reinforced by the simultaneous consideration of a larger body of curated samples, which are of lesser quality and focus but are available in greater number [Wu and Dietterich, 2004]. The motivation is to transfer information learned on data-rich applications to related applications that may be data-poor. The five ADMET classification tasks encountered here are all sampled from different sources during the lead optimisation stage of the process and are employed to predict different target properties. Nevertheless, they all ask the same, more generic, question, i.e. whether compounds should be rejected from the design process or retained for further development. Relationships learned on the individual tasks combine to cordon off a 'drug-like' region of chemical space. Via the use of an encoded representation of whole-molecular structure and the formulation of all problems of molecular classification as select / reject, one may be able to transfer knowledge gained from learning on large collections of data at the lead generation stages, e.g. to predict 'drug-likeness', to the more specific data-poor scenarios encountered during lead optimisation. Conversely, specific ADMET relationships may be employed as an ensemble of auxiliary tasks [Marx et al., 2005] in order to improve the prediction of drug-likeness earlier in the process.

The ability to learn upon abstract representations of molecular structure and, moreover, upon relevant abstract mappings of explicit molecular representation without feature selection is a powerful feature of the support vector machine algorithm. Choice of molecular representation and varying methods of data treatment throughout the discovery process inhibit fully automated procedures for the target-to-lead identification of novel pharmaceutical products. The ability of kernel methods, such as those described above and in the literature, to create accurate predictors on high-dimensional whole-molecular representations of compound structure may facilitate such an approach. The experimental findings are that the Tanimoto kernel treats Daylight molecular representation at least as effectively as standard SVM kernels treat a variety of generic real-valued representations. Although some deficiency is observed on problems that are likely to require structural description more expressive than the 2D information provided here, the overall suggestion is of a procedure that facilitates the integration of SVM classification into present *in silico* drug discovery practice. Further development of standardised molecular representations and domain-relevant kernel functions with which to assess them may yield a system within which all structure-property analysis, from target to lead, may be performed using the same, uniform representation and the same assessment of inter-molecular similarity.

# Chapter 7

# Conclusion

## 7.1 Summary & Contributions

The classification of biological properties of interest according to aspects of molecular structure is becoming vital to the contemporary drug discovery process. Data drawn from industrial processes often challenges the creation of predictive relationships between process and outcome. Pharmaceutical classification is no exception. There is no universal standard for the computational representation of molecular structure, training data is subject to bias and erroneous information during extraction from the process and there exist complex, non-linear relationships between target and descriptive attributes. A sensible approach to such challenges is to take a well-founded technique and adapt it to cope with real-world applications in a manner that affects its analytical strengths as little as possible.

This thesis has investigated the application of supervised machine learning to the analysis of data drawn from the lead optimisation stage of the contemporary drug discovery process. *Support vector machines* (SVMs) are demonstrated a suitable technique with which to build classifiers capable of distinguishing discrete classes of biological behaviour according to the structures of pharmaceutical compounds. Further investigations yield adaptations, both to the technique and to the practice of its application to drug discovery, with the potential to improve performance.

SVMs were applied to a series of separation problems drawn from the lead optimisation stage of the drug discovery process. The target properties in these problems ranged from specific, such as compound ability to cross a particular membrane, to abstract, such as the effect of metabolism upon *in vivo* compound concentrations. The research hypotheses tested were that the application benefits from the inclusion of another technique in its analysis, the technique is capable of analysing the application successfully and that adapting both technique and application in a domain-relevant manner may increase performance levels. Therefore, the contributions made by this work to the field of machine learning and its role in drug discovery may be summarised as follows.

Chapter 3, *ADMET Data and Experimental Practice*, defined an experimental framework for the comparison of supervised machine learning algorithms when applied to create

ADMET SPC relationships for lead optimisation. Considerations embodied in the practice include the provision of representative training and test partitions from limited data, balanced performance assessment, a context against which to measure observed differences in balanced performance and assessment across equivalent algorithmic free-parameter ranges.

Section 4.1 of Chapter 4, *Support Vector Machines for ADMET Property Classification*, compared SVM performance against that of several state-of-the-art supervised machine learning techniques when applied to small collections of real-world ADMET SPC data. The SVM algorithm was observed to be competitive against the other methods assessed and the comparison demonstrated the challenges posed to the successful application of supervised machine learning by the data encountered. In concurrence with arguments presented in the background material (Chapter 2), the presence of training data class imbalance, data paucity and sub-optimal molecular representation suggest that any supervised learning method that is able to cope with such impediments is welcome in this learning scenario.

Section 4.2 of Chapter 4 considered existing pharmaceutical outlier assessment and data sampling methods to strategically reduce a majority data class in order to increase predictive balance. Data reduction, even on small collections of training data, was shown to provide results similar to direct algorithmic weighting and a weighted SVM regularisation procedure. That data reduction improves balanced performance by reducing a majority class so at to retain its fundamental properties, rather than ignoring examples near the decision boundary, suggests further measures to typify the training data and alternative SVM applications that also employ a subset of the available training data.

Consideration of data typification led to work performed in Chapter 5, *Neighbourhood Influence on Support Vector Machine Classification*, which investigated SVM kernel composition from multiple sub-samples of training data in order to strengthen the information from which SPC relationships are learned. Simple linear SVMs were shown to be particularly susceptible to the effects of local weighting and their performance improved to the level of standard non-linear SVMs in the majority of cases. A proof-of-concept examined the use of an existing pharmaceutical clustering method for SVM kernel construction. Further to this, a domain-weighted SVM was applied to strategically balanced training data and shown to improve previously observed levels of balanced generalisation accuracy.

Chapter 6, *Tanimoto Kernels for Support Vector Machine Classification*, demonstrated that small-scale SPC relationships may be learned effectively on a sparse, binary representation of encoded molecular structure that is normally associated with earlier stages of the discovery process. Formulation of a domain-relevant similarity measure as a valid SVM kernel function was fundamental to this approach and improved performance on the representation employed. Thus, a domain-relevant SVM displayed ability to treat the five ADMET SPC problems when represented by a schema suitable for use from the very start of the discovery process through to its later stages. This approach is discussed as a contribution towards a unified discovery process. The representation and treatment of data drawn from different tasks involved in *in silico* drug discovery by the same method invites consideration of some

interesting developments in contemporary machine learning and how they may contribute further to the drug discovery process.

The research hypotheses are answered positively, by the investigations performed and by the future work that they suggest. The domain requires greater predictive accuracy in order to focus the drug discovery process and to reduce inefficiency. A standard form of the SVM algorithm provides acceptable predictive performance across a variety of molecular representations. Support vector machines are shown to be capable tools for the analysis of focused ADMET SPC relationships found in the early lead optimisation stage of the discovery process, but it is not demonstrated here that the standard SVM algorithm comprehensively outperforms all other techniques when applied to the domain. Instead, it is the flexibility of the algorithm to domain-relevant adaptation and consideration of its future role for *in silico* screening that denote it as a particularly useful technique for drug discovery, in terms both of the predictive accuracy provided and ease of integration into present practice. There do exist some problems upon which generalisation performance of all methods is limited, primarily due to a paucity of training data and an uncertain relationship between 2D representations of molecular structure and abstract target properties. However, such issues are more likely to be overcome by revision of the process than the inclusion of a new machine learning method. The theoretical strengths of the SVM technique and domain-relevant adaptations of it may be applied process-wide and invite the prospect of a unified treatment of pharmaceutical classification.

## 7.2 Suggested Future Work

The investigations of chapters 4, 5 & 6 prompt suggestions for future work in order to further the research aims. The strategic reduction of majority class data, via outlier removal and even sampling respectively, suggests the further involvement of robust class identity measures prior to classifier creation as an alternative to regularisation. SVM-related advances may be provided by different formulations of the algorithm, e.g. the one-class SVDD classifier [Tax and Duin, 1999] to direct the strategic removal of training data, or the Reduced SVM classifier (RSVM) [Lee and Mangasarian, 2001] to learn on a strategically reduced kernel matrix. An automated approach to data typification may be provided by the application of SVMs within a query learning framework [Campbell et al., 2000]. For example, the training data may be treated as unlabelled except for an initial balanced subset, employed to seed the query learning process.

The use of an existing pharmaceutical clustering procedure is introduced to influence the SVM kernel matrix and focus the learned classifier upon relevant patterns in the training data. Of immediate interest is an expanded trial of this approach on a wider selection of data and across a wider range of cluster thresholds in order to provide more information with which to refine the procedure. Potential refinements include weighting partial kernel matrix contributions according to considerations of balanced generalisation performance or via an optimisation procedure, as recently performed on attribute subsets [Rätsch et al.,

2006].

The application of SVMs to a binary encoding of molecular structure and the formulation of a widely-used pharmaceutical similarity measure as a valid SVM kernel function yields several avenues for development of both technique and application. Of immediate interest is extension of the work presented in Chapter 6 to consider the use of more expressive representations of whole-molecular structure. The design of SVM kernel functions to provide domain-relevant assessment of other representations, such as NMR spectra [Xu and Yang, 1998], is also an interesting direction. Further considerations involve the potential of a standardised representation of molecular structure, in tandem with domain-relevant methods of analysis, to promote a unified approach to classifier creation across all stages of the discovery process.

The largely unexplored theme that recurs throughout this investigation is the use of unlabelled data emerging from combinatorial synthesis, virtual or *in vitro*, to reinforce the relationships inferred from small collections of labelled compounds. With respect to previous suggestions for future work, a single representational schema yields a large body of data upon which to apply such domain-relevant techniques within a transductive framework. The design of domain-relevant feature spaces has recently entered current thinking on the prediction of biological properties from aspects of molecular structure. The combination of domain-relevant kernel design within appropriate reinforcement strategies appears to provide a promising direction for future research in this field, especially in light of the success demonstrated thus far by these methods when applied individually [Warmuth et al., 2002; Fröhlich et al., 2006]. The recent publication of work regarding the application of domain-relevant similarity measures to QSAR classification in lead generation [Chen et al., 2006] and custom SVM kernel function design for the creation of ADMET SPC relationships [Fröhlich et al., 2006] suggest that the work presented in Chapter 6 should be applied further to binary lead generation data and compared to the existing works towards the creation of a unified *in silico* discovery process [Beresford et al., 2002; van de Waterbeemd, 2003].

Further to this approach, a wider direction for *in silico* SPC analysis is suggested upon consideration of recent advances in the field of machine learning. As concluded by this thesis, the use of an abstract encoding of molecular structure, upon which the machine learning of small SPC relationships is shown to be feasible, promotes similar treatment of different classification tasks drawn from across the discovery process. Also apparent is that the five ADMET classification tasks encountered here all consider different target properties during the lead optimisation stage of the process but all ask the same, more generic, question, i.e. whether compounds should be rejected from the design process or retained for further development.

The concept of inductive transfer [Wu and Dietterich, 2004; Marx et al., 2005; Rosenstein et al., 2005] involves the use of data from tasks related to the primary classification objective in order to strengthen generalisation performance. The motivation is to transfer information learned on data-rich applications to related applications that may be data-poor. It may be seen that, via the use of a relevant uniform representation of molecular structure

and the formulation of all problems of molecular classification as select / reject, one may be able to transfer knowledge gained from learning on large collections of data at the lead generation stages, e.g. to predict 'drug-likeness', to the more specific data-poor scenarios encountered during lead optimisation [Wu and Dietterich, 2004]. Conversely, specific ADMET relationships may be employed as an ensemble of auxiliary data [Marx et al., 2005] in order to improve the prediction of drug-likeness earlier in the process.

## 7.3 Closing Statement

The investigation of SVMs for *in silico* prediction of ADMET properties during lead optimisation has demonstrated the algorithm's potential for successful application. The optimised, non-stochastic solution of the SVM learning algorithm, which embodies the structural risk minimisation principle, combined with the ability to incorporate domain-relevance via the design of appropriate kernel functions, make SVMs a particularly useful technique for drug discovery. Moreover, the SVM algorithm may be formulated for regression and outlier detection tasks as well as for classification. The findings of this thesis, alongside contemporaneous work, and the number of directions for further research suggest that support vector machines have the potential to represent a step-change improvement in *in silico* screening practices, in the same manner as did ANNs a decade before them. This statement is tempered, however, by issues of varied molecular representation - noted by Hansch [1969] as a barrier to successful structure-property classification nearly forty years ago - and a lack of realistic publicly-available SPC data drawn from industrial processes. Were these impediments treated in a manner sympathetic to the use of machine learning as an integral part of drug discovery, the fully automated discovery processes widely anticipated by the literature [Beresford et al., 2002; van de Waterbeemd, 2003] may become feasible.

# Bibliography


Atkinson F, Cole S, Green C, van de Waterbeemd H (2002). Lipophilicity and Other Parameters Affecting Brain Penetration. Current Medicinal Chemistry Central Nervous System Agents 2(3): 229–240.

Bajorath J (2000). Integration of virtual and high-throughput screening. Nature Drug Discovery 1: 882–894.

Bannwarth W, Felder E (2000). Combinatorial Chemistry. Methods and Principals in Medicinal Chemistry. Wiley-VCH, Weinheim, Germany.

Bennett KP, Blue J (1997). A support vector machine approach to decision trees. Math Report No. 97-100. Rensselaer Polytechnic Institute. Troy, NY.

Bennett KP, Demiriz A (1998). Semi-supervised support vector machines. MS Kearns, SA Solla, DA Cohn, editors, Advances in Neural Information Processing Systems, 11. MIT Press, Cambridge, MA. p. 368–374.

Beresford AP, Selick HE, Tarbit MH (2002). The Emerging Importance of Predictive ADME Simulation in Drug Discovery. Drug Discovery Today 7(2): 109–116.

Bishop C, Tipping M (2000). Variational relevance vector machines. Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers. p. 46–53.

Bishop CM (1995a). Neural Networks for Pattern Recognition. Oxford University Press, Oxford, UK.

Bishop CM (1995b). Training with noise is equivalent to Tikhonov regularization. Neural Computation 7(1): 108–116.

Blake CL, Merz CJ (1998). UCI repository of machine learning databases. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Blanz V, Schölkopf B, Bulthoff HH, Burges C, Vapnik V, et al. (1996). Comparison of View-Based Object Recognition Algorithms Using Realistic 3D Models. ICANN. p. 251–256.

Böcker A, Schneider G, Teckentrup A (2004). Status of HTS Data Mining Approaches. QSAR and Combinatorial Science 23: 207–213.

Böhm HJ, Schneider G (2000). Virtual Screening for Bioactive Molecules. Wiley-VCH, Weinheim, Germany.

Böhm HJ, Stahl M (2000). Structure-Based Library Design: Molecular Modelling Merges with Combinatorial Chemistry. Current Opinion in Chemical Biology 4: 283–286.

Brailovsky VL, Barzilay O, Shahave R (1999). On Global, Local, Mixed and Neighbourhood Kernels for Support Vector Machines. Pattern Recognition Letters 20: 1183–1190.

Breiman L (1994). Bagging predictors. Technical Report 421. Department of Statistics, Berkeley.

Brown MPS, Grundy WN, Lin D, Cristianini N, Sugnet CW, et al. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. Proceedings of the National Academy of Science USA 97: 262–267.

Bryant CH, Adam AE, Taylor DR, Rowe RC (1997). Using inductive logic programming to discover knowledge hidden in chemical data. Chemometrics and Intelligent Laboratory Systems 36: 111–123.

Burbidge R, Trotter MWB, Holden SB, Buxton BF (2001). Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis. Computers and Chemistry 26(1): 4–15.

Burbidge RD (2004). Heuristic Methods for Support Vector Machines with Applications to Drug Discovery. Ph.D. thesis. Department of Computer Science, UCL, London, UK.

Burges CJC (1998). A tutorial on support vector machines for pattern recognition. Knowledge Discovery and Data Mining. volume 2. p. 121–167.

Butina D (1999). Unsupervised data base clustering based on Daylight's fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets. Journal of Chemical Information and Computer Sciences 39(4): 747–750.

Buxton BF, Treleaven PC, Holden SB, Langdon WB (2002). Robust classification and knowledge engineering techniques (rocket). URL `http://www.cs.ucl.ac.uk/research/rocket`. University College London, UK.

Byvatov E, Fechner U, Sadowski J, Schneider G (2003). Comparison of support vector machine and artificial neural network systems for drug / nondrug classification. Journal of Chemical Information and Computer Sciences 43: 1882–1889.

Campbell C, Cristianini N, Smola A (2000). Query learning with large margin classifiers. Proceedings of ICML2000 (Stanford, CA, 2000). p. 8.

Catell RB (1966). The scree test for the number of factors. Multivariate Behavioural Research 1: 245–276.

Chang CC, Hsu CW, Lin CJ (2000). The Analysis of Decomposition Methods for Support Vector Machines. IEEE Transactions on Neural Networks 11(4): 1003–1008.

Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002). Choosing Multiple Parameters for Support Vector Machines. Machine Learning 46(1): 131–159.

Chen B, Harrison RF, Pasupa K, Willett P, Wilton DJ, et al. (2006). Virtual screening using binary kernel discrimination: Effect of noisy training data and the optimization of performance. J Chem Inf Model 46: 478–486.

Chu W (2003). Bayesian approach to support vector machines. Ph.D. thesis. National University of Singapore.

Cover TM, Hart PE (1967). Nearest neighbour pattern classification. IEEE Transactions on Information Theory 13: 21–27.

Cristianini N, Shawe-Taylor J (2000). An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods). Cambridge University Press.

Cruciani G, Pastor M, Guba W (2000). Volsurf - A New Image Processing method for 3D Grid Map and QSAR Studies. Europ J Pharm Sci 11: 29–39.

Daylight (2006). Fingerprints - screening and similarity. URL `http://www.daylight.com/dayhtml/doc/theory/`.

Debouck C, Metcalf B (2000). The impact of genomics on drug discovery. Annual Review of Pharmacology and Toxicology 40: 193–208.

Devillers J (1996a). Genetic Algorithms in Molecular Modeling. Academic Press, New York, USA.

Devillers J (1996b). Neural Networks in QSAR and Drug Design. Academic Press, New York, USA.

Dixon GK, Major JS, Rice MJ, editors (2000). High Throughput Screening: The Next Generation. BIOS Scientific Publishers Ltd., Oxford, UK. Papers presented at the SCI conference *High Throughput Screening: The Next Generation*, University of Surrey, Guildford, UK, 7–9 June 1999.

Dominik A (2000). Computer-assisted library design. Bannwarth and Felder [2000].

Doninger S, Hofmann T, Yeh J (2002). Predicting CNS permeability of drug molecules: Comparison of neural network and support vector machine algorithms. Journal of Computational Biology 9(6): 849–864.

Drewry DH, Young SS (1999). Approaches to the Design of Combinatorial Libraries. Chemometrics and Intelligent Laboratory Design 48: 120.

Duda RO, Hart PE, Stork DG (2000). Pattern Classification. John Wily & Sons Inc., New York, USA. 2nd edition.

Ecker GF, Noe CR (2004). In Silico Prediction Models for Blood-Brain Barrier Permeation. Current Medicinal Chemistry 11(12): 1528–1617.

Eisen MB, Spellman PT, Brown PO, Botstein D (1998). Cluster analysis and display of geneome-wide expression patterns. PNAS 95(25): 14863–14868.

Embrechts M, Breneman C, Bennett K (2003). Automated design and discovery of novel pharmaceuticals using semi-supervised learning in large molecular databases. URL `www.drugmining.com`. rensselaer Polytechnic Institute, New York, USA.

Eriksson L, Johansson E (1996). Multivariate Design and Modeling in QSAR. Chemometrics and Intelligent Laboratory Systems 34: 1–19.

Evgeniou T (2000). Learning with Kernel Machine Architectures. Ph.D. thesis. Department of Electrical Engineering and Computer Science, MIT, USA.

Filzmoser P, Reimann C, Garrett RG (2005). Multivariate outlier detection in exploration geochemistry. Computers and Geosciences 31: 579–587.

Franke R, Gruska A (1995). Principal component and factor analysis. van de Waterbeemd [1995]. R. Mannhold, P. Krogsgaard-Larsen and H. Timmerman (Eds.).

Franklin S, Brodeur M (1997). A practical application of a robust multivariate outlier detection method. Proceedings of the Survey Research Methods Section, American Statistical Association. p. 186–191.

Freund Y (1995). Boosting a weak learning algorithm by majority. Information and Computation 121(2): 256–285.

Fröhlich H, Wegner JK, Sieker F, Zell A (2005). Optimal assignment kernels for attributed molecular graphs. Proceedings of the 22nd International Conference on Machine Learning. Omnipress. p. 225–232.

Fröhlich H, Wegner JK, Sieker F, Zell A (2006). Kernel functions for attributed molecular graphs - a new similarity-based approach to ADME prediction in classification and regression. QSAR and Combinatorial Science 25(4): 317–326.

Fröhlich H, Wegner JK, Zell A (2004). Towards optimal descriptor subset selection with support vector machines. QSAR and Combinatorial Science 23: 311–318.

Fung GM, Mangasarian OL, Smola AJ (2002). Minimal Kernel Classifiers. Journal of Machine Learning Research 3: 303–321.

Gillet V, Willett P, Bradshaw J (1998). Identification of Biological Activity Profiles Using Sub-structural Analysis and Genetic Algorithms. Journal of Chemical Information in Computer Science 38: 165–179.

Gini G, Testaguzza V, Benfenati E, Todeschini R (1998). Hybrid toxicology expert system: architecture and implementation of a multi-domain hybrid expert system for toxicology. Chemometrics and Intelligent Laboratory Systems 43: 135–145.

Goldberg DE (1989). Genetic Algorithms in Search, Optimization and Machine Learing. Addison Wesley.

Goodford PJ (1995). A Computational Procedure for Determining Energetically Favourable Binding Sites on Biologically Important Macromolecules. J Med Chem 28: 849–857.

Goonatilake S, Khebbal S (1995). Intelligent Hybrid Systems. University College London.

Guyon I, Gunn S, Ben-Hur A, Dror G (2005). Result Analysis of the NIPS 2003 Feature Selection Challenge. LK Saul, Y Weiss, L Bottou, editors, Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, MA. p. 545–552.

Hammond P, Hutton T, Allanson J, Campbell L, Hennekam R, et al. (2004). 3D analysis of facial morphology. Am J Med Genet A 126(4): 339–348.

Hand DJ (1981). Discrimination and Classification. Wiley, Chichester.

Hansch C (1969). A Quantitative Approach to Biochemical Structure-Activity Relationships. Acc Chem Res 2: 232–239.

Hardin J, Rocke DM (1999). The distribution of robust distances. Technical report. University of California at Davis, USA.

Haussler D (1999). Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10. Department of Computer Science, University of California at Santa Cruz.

Hawkins DM, Young SS, Rusinko A (1997). Analysis of a Large Structure-Activity Data Set Using Recursive Partitioning. Quantitative Structure Activity Relationships 16(4): 296–302.

Hearst MA (1998). Trends and controversies: support vector machines. IEEE Intelligent Systems 13(4): 18–28.

Hert J, Willett P, Wilton DJ (2006). New methods for virtual screening: Use of data fusion and machine learning to enhance the effectiveness of similarity searching. J Chem Inf Model 46: 462–470.

Hertz J, Krogh A, Palmer RG (1991). An Introduction to the Theory of Neural Computation. Lecture Notes Volume I. Addison Wesley.

Hibbert DB (1993). Genetic algorithms in chemistry. Chemometrics and Intelligent Laboratory Systems 19: 277–293.

Hirst JD, King RD, Sternberg MJE (1994). Quantitative structure-activity relationships by neural networks and inductive logic programming. I. the inhibition of dihydrofolate reductase by pyrimidines. Journal of Computer Aided Molecular Design 8: 405–420.

Holliday JD, Hu CY, Willett P (2002). Grouping of coefficients for the calculation of inter-molecular similarity and dissimilarity using 2D fragment bit-strings. Combinatorial Chemistry and High Throughput Screening 5: 155–166.

Hollinger M (1997). Introduction to Pharmacology. Taylor & Francis, Washington DC, USA.

Hsu CW, Lin CJ (2001). A comparison of methods for multi-class support vector machines. Technical Report 19. National Taiwan University, Taipei, Taiwan.

Hsu CW, Lin CJ (2002). A Simple Decomposition Method for Support Vector Machines. Machine Learning 46(1-3): 291–314.

Jaakkola T, Diekhans M, Haussler D (1999). Using the Fisher kernel method to detect remote protein homologies. T Lengauer, R Schneider, P Bork, D Brutlag, J Glasgow, HW Mewes, R Zimmer, editors, Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology, ISMB'99 (Heidelberg, Germany, August 6-10, 1999). AAAI Press, Menlo Park. p. 149–158.

Jaakkola TS, Meila M, Jebara T (2000). Maximum entropy discrimination. SA Solla, TK Leen, KR Müller, editors, Advances in Neural Information Processing Systems, 12. MIT Press, Cambridge, MA.

James CA, Weininger D, Delaney J (2000). Daylight Theory Manual - Daylight 4.71. Daylight Chemical Information Systems Inc., 27401 Los Altos, Suite #360, Mission Viejo, CA.

Jarvis R, Patrick E (1973). Clustering Using a Similarity Measure Based on Shared Near Neighbours. IEEE Trans Comput C-22: 1025–1034.

Joachims T (1998a). Making Large-Scale SVM Learning Practical. MI Jordan, MJ Kearns, SA Solla, editors, Advances in Neural Information Processing Systems, 10. The MIT Press, Cambridge, MA.

Joachims T (1998b). Text categorization with support vector machines: learning with many relevant features. C Nédellec, C Rouveirol, editors, Proceedings of ECML-98, 10th European Conference on Machine Learning. Springer Verlag, Heidelberg, DE. p. 137–142.

Joachims T (1999). Transductive inference for text classification using support vector machines. I Bratko, S Dzeroski, editors, Proceedings of the 16th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA. p. 200–209.

Joachims T (2000). Estimating the generalization performance of a SVM efficiently. P Langley, editor, Proceedings of ICML-00, 17th International Conference on Machine Learning. Morgan Kaufmann Publishers, San Francisco, USA. p. 431–438.

Jolliffe IT (1986). Principal Component Analysis. Springer-Verlag, New York, USA.

Joseph-McCarthy D (1999). Computational Approaches to Structure-Based Ligand Design. Pharmacology Therapeutics 84: 179–191.

Jurs PC, Dixon SL, Egolf LM (1995). Representations of molecules. van de Waterbeemd [1995]. R. Mannhold, P. Krogsgaard-Larsen and H. Timmerman (Eds.).

Kaiser HF (1960). The application of electronic computers to factor analysis. Psychol Meas 20: 141–151.

Kennard RW, Stone LA (1969). Computer Aided Design of Experiments. Technometrics 11: 137–148.

Kier LB (1995). Atom-level descriptors for QSAR analyzes. van de Waterbeemd [1995]. R. Mannhold, P. Krogsgaard-Larsen and H. Timmerman (Eds.).

King RD, Muggleton S, Lewis RA, Sternberg MJE (1992). Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. Proc Natl Acad Sci 89: 11322–11326.

Kittler J, Hatef M, Duin RPW, Matas J (1998). On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3): 226–239.

Kohavi R (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. IJCAI. p. 1137–1145.

Kondor R, Lafferty J (2002). Diffusion kernels on graphs and other discrete input spaces. Proceedings of the International Conference on Machine Learning (ICML).

Kövesdi I, Dominguez M, Ôrfi L, Náray-Szabó G, Varró A, et al. (1999). Application of neural networks in structure-activity relationships. Med Res Rev 19(3): 249–269.

Kruskal JB (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29: 1–27.

Labute P (1998). Quasar-binary: A new method for the analysis of high throughput screening data. URL `http://www.netsci.org/Science/CompChem/feature21.html`. Presented at the Charleston Conference, March 1998.

Langdon WB, Barrett SJ, Buxton BF (2001). Genetic programming for combining neural networks for drug discovery. Presented at the WSC6 Conference, September 2001.

Lee KK, Gunn SR, Harris CJ, Reed PAS (2001). Classification of Imbalanced Data with Transparent Kernels. Proceedings of the INNS-IEEE International Joint Conference on Neural Networks. p. 2410–2415.

Lee YJ, Mangasarian OL (2001). RSVM: Reduced support vector machines. Proceedings of the First SIAM International Conference on Data Mining.

Li T, Mei H, Cong P (1999). Combining nonlinear PLS with the numeric genetic algorithm for QSAR. Chemometrics and Intelligent Laboratory Systems 45: 177–184.

Lin K, Lin C (2003). A study on reduced support vector machines. IEEE Transactions on Neural Networks 14(6): 1449–1459.

Lin Y, Lee Y, Wahba G (2000). Support vector machines for classification in nonstandard situations. Technical Report 1016. Department of Statistics, University of Wisconsin. Madison WI, USA.

Lipinski CA, Lombardo F, Dominy BW, Feeney PJ (1997). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Advanced Drug Delivery Reviews 23: 3.

Lodhi H, Saunders C, Shawe-Taylor J, Cristianini N, Watkins C (2000). Text classification using string kernels. Neural Information Processing Systems. MIT Press, Cambridge, MA. p. 563–569.

Manallack D, Livingstone D (1999). Neural Networks in Drug Discovery: Have They Lived Up to Their Promise? Eur J Med Chem 34: 195–208.

Martens H, Næs T (1989). Multivariate Calibration. John Wiley & Sons.

Marx Z, Rosenstein MT, L. PK, Dietterich TG (2005). Transfer Learning with an Ensemble of Background Tasks. *Inductive Transfer: 10 Years Later. NIPS 2005 Workshop.*

Mathieson M (2001). Applying Active Learning Using Support Vector Machines to Drug-Finding Data. Master's thesis. University of California, Santa Cruz, CA, USA.

Mathworks (2002). Matlab v6.5. URL `http://www.mathworks.com`.

Matter H, Baringhaus KH, Naumann T, Klabunde T, Pirard B (2001). Computational Approaches Towards the Rational Design of Drug-like Compound Libraries. Combinatorial Chemistry and High Throughput Screening 4: 453–475.

Matthews BW (1975). Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. Biochim Biophys Acta 405: 442–451.

McNemar Q (1947). Note on the Sampling Error of the Difference between Correlated Proportions or Percentages. Psychometrika 12(2): 153–157.

MDL (1994). MACCS-II Menu Reference Manual - v2.2. MDL Information Systems, San Leandro, CA.

Mercer J (1909). Functions of Positive and Negative Type and their Connection with the Theory of Integral Equations. Philos Trans Roy Soc London A 209: 415–446.

Mitchell M (1996). An Introduction to Genetic Algorithms. Complex Adaptive Systems. MIT-Press, Cambridge, MA.

Mitchell TM (1997). Machine Learning. McGraw-Hill.

Molinaro AM, Simon R, Pfeiffer RM (2005). Prediction error estimation: a comparison of resampling methods. Bioinformatics 21(15): 3301–3307.

Morris H (2004). Support vector domain description for marker gene identification. Master's thesis. Dept. of Computer Science, UCL, London, UK.

Murthy SK, Kasif S, Salzberg S (1998). A system for oblique induction of decision trees. Journal of Artificial Intelligence Research 2: 1–32.

Osuna EE, Freund R, Girosi F (1997). Support vector machines: Training and applications. Technical Report AIM-1602. Massachusetts Institute of Technology.

Panfili P (1999). Assay miniaturization for high-throughput screening. American Biotechnology Laboratory 17(10): 12.

Parzen E (1962). On estimation of a probability density function and mode. Ann Math Stat 33: 1065–1076.

Provost FJ, Jensen D, Oates T (1999). Efficient progressive sampling. Knowledge Discovery and Data Mining. p. 23–32.

Quinlan JR (1986). Induction of decision trees. Machine Learning 1(1): 81–106.

Rätsch G, Onoda T, Müller KR (2000). Soft margins for adaboost. Machine Learning p. 1–35.

Rätsch G, Sonnenburg S, Schäfer C (2006). Learning interpretable SVMs for biological sequence classification. BMC Bioinformatics 7(S1): 9.

Rishton GM (2003). Nonleadlikeness and leadlikeness in biochemical screening. Drug Discovery Today 8(2): 86–96.

Roland M, Tozer T (1995). Clinical Pharmacokinetics: Concepts and Applications. Williams & Wilkins, Philadelphia, USA. 3rd edition.

Rosenblatt F (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psych Rev 65: 386–407. (Reprinted in *Neurocomputing* (MIT Press, 1988).).

Rosenstein MT, Marx Z, L. PK, Dietterich TG (2005). To Transfer or Not To Transfer. *Inductive Transfer: 10 Years Later. NIPS 2005 Workshop.*

Russell S, Norvig P (2003). Artificial Intelligence: A Modern Approach. Pearson Education Inc., New Jersey, USA. 2nd edition.

Russo M, Santagati NA, Lo Pinto E (1998). Medicinal chemistry and fuzzy logic. Journal of Information Sciences 105: 299–314.

Sadowski J (2000). Optimization of Chemical Libraries by Neural Networks. Current Opinion in Chemical Biology 4: 280–282.

Santos Magalhães NS, De Holanda Cavalcanti SC, Alencar De Menezes IR, De Sousa Araújo AA, De Oliveira HM, et al. (1999). Automated search for potentially active compounds by using cluster trees. European Journal of Medicinal Chemistry 34: 83–92.

Schneider G (2000). Neural Networks are Useful Tools for Drug Design. Neural Networks 13: 15–16.

Schneider G, Downs G (2003). Machine learning methods in QSAR modelling. QSAR and Combinatorial Science 22(5). Special issue. G. Schneider and G. Downs (Eds.).

Schölkopf B, Simard P, Smola AJ, Vapnik V (1998). Prior Knowledge in Support Vector Kernels. MI Jordan, MJ Kearns, SA Solla, editors, Advances in Neural Information Processing Systems. The MIT Press, USA. volume 10.

Schölkopf B, Smola AJ, Müller KR (1999). Kernel principal component analysis. B Schölkopf, C Burges, AJ Smola, editors, Advances in Kernel Methods - Support Vector Learning. The MIT Press, USA. p. 327–352.

Schölkopf B, Sung K, Burges C, Girosi F, Niyogi P, et al. (1997). Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers. IEEE Transactions on Signal Processing 45(11): 2758–2765.

Schwenk H (1998). The Diabolo Classifier. Neural Computation 10(8): 2175–2200.

Schwenk H, Bengio Y (1998). Training methods for adaptive boosting of neural networks. MI Jordan, MJ Kearns, SA Solla, editors, Advances in Neural Information Processing Systems. The MIT Press, USA. volume 10. p. 647–653.

Scott MJJ, Niranjan M, Prager RW (1998). Realisable classifiers: improving operating performance on variable cost problems. British Machine Vision Conference. BMVC, September 1998.

Shi LM, Fan Y, Myers TG, O'Connor PM, Paull KD, et al. (1998). Mining the NCI anticancer drug discovery databases: Genetic function approximation for the QSAR study of anticancer ellipticine analogues. Journal of Chemical Information and Computer Sciences 38: 189–199.

Shin H, Cho S (2003). How to deal with large dataset, class imbalance and binary output in SVM based response model. Proceedings of the Korean Data Mining Conference. p. 93–107.

Shipp MA, Ross KN, Tamayo P, Weng AP, Kutok JL, et al. (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. Nature Medicine 8(1): 68–74.

Skurichina M (2001). Stabilizing Weak Classifiers. Ph.D. thesis. T. U. Delft.

Sollich P (2000). Probabilistic Methods for Support Vector Machines. S Solla, T Leen, KR Müller, editors, Advances in Neural Information Processing Systems. The MIT Press. volume 12. p. 349–355.

Sollich P (2002). Bayesian Methods for Support Vector Machines: Evidence and Predictive Class Probabilities. Machine Learning 46(1-3): 21–52.

Spotfire Inc. (2005). DecisionSite 8.1. URL `http://www.spotfire.com/`.

SPSS (2002). Clementine v7.1. URL `http://www.spss.com`.

Tax DMJ, Duin RPW (1999). Data Domain Description using Support Vectors. M Verleysen, editor, Proceedings of the European Symposium on Artificial Neural Networks. D.Facto, Brussels. p. 251–256.

Tax DMJ, Duin RPW (2004). Support Vector Data Description. Machine Learning 54(1): 45–66.

Tax DMJ, Ypma A, Duin RPW (1999). Pump failure detection using support vector data descriptions. Lecture Notes in Computer Science 1642: 415.

Tong C, Svetnik V (2002). Novelty detection in mass spectral data using a support vector machine method. Computing Science and Statistics .

Trotter M, Buxton B, Holden S (2001). Support Vector Machines in Combinatorial Chemistry. Measurement and Control 34(8): 235–239.

Trotter MWB, Holden SB (2003). Support Vector Machines for ADME Property Classification. QSAR and Combinatorial Science 22(5): 533–548.

Tu Y, Stolovitzky G, Klein U (2002). Quantitative noise analysis for gene expression microarray experiments. Proceedings of the National Academy of Science USA 99(22).

van de Waterbeemd H (1995). Chemometric Methods in Molecular Design. VCH, Weinheim, Germany. R. Mannhold, P. Krogsgaard-Larsen and H. Timmerman (Eds.).

van de Waterbeemd H (2003). ADMET *In Silico* Modelling: Towards Prediction Paradise? Nature Drug Discovey Reviews 2: 192–204.

Van Hijfte L, Marciniak G, Froloff N (1999). Combinatorial Chemistry, Automation and Molecular Diversity: New Trends in the Pharmaceutical Industry. Journal of Chromatography B 725: 315.

Vapnik VN (1995). The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc.

Vapnik VN (1998). Statistical Learning Theory. John_Wiley.

Vapnik VN (1999). An Overview of Statistical Learning Theory. IEEE Transactions on Neural Networks 10(5): 988.

Vert JP (2002). A Tree Kernel to Analyze Phylogenetic Profiles. Bioinformatics 18(S1): S276–S284.

Walczak B, Massart DL (1996). The radial basis functions - partial least squares approach as a flexible non-linear regression technique. Analytica Chimica Acta 331: 177–185.

Walters WP, Murcko MA (2002). Prediction of 'drug-likeness'. Advanced Drug delivery Reviews 54: 255–271.

Wang HW, Trotter MWB, Lagos D, Bourboulia D, Henderson S, et al. (2004). Kaposi sarcoma herpesvirus-induced cellular reprogramming contributes to the lymphatic endothelial gene expression in kaposi sarcoma. Nature Genetics 36(7): 687–693.

Ward J, McGuffin L, Buxton B, Jones D (2003). Secondary structure prediction with support vector machines. Bioinformatics 19(13): 1650–1655.

Warmuth MK, Rätsch G, Mathieson M, Liao J, Lemmen C (2002). Active learning in the drug discovery process. S Becker, TG Dietterich, Z Ghahramani, editors, Neural Information Processing Systems 14. MIT Press, Cambridge, MA.

Watkins C (1999). Dynamic alignment kernels. Technical Report CSD-TR-98-11. Royal Holloway, University of London. Egham, Surrey, UK.

Weston J, Pérez-Cruz F, Bousquet O, Chapelle O, Elisseeff A, et al. (2003). Feature selection and transduction for prediction of molecular bioactivity for drug design. Bioinformatics 19(6): 764–771.

Weston J, Watkins C (1999). Support vector machines for multiclass pattern recognition. Proceedings of the Seventh European Symposium on Artificial Neural Networks.

Wilton DJ, Harrison RF, Willett P (2006). Virtual screening using binary kernel discrimination: Analysis of pesticide data. J Chem Inf Model 46: 471–477.

Wold S, Eriksson L (1995). Validation tools. van de Waterbeemd [1995]. R. Mannhold, P. Krogsgaard-Larsen and H. Timmerman (Eds.).

Wu P, Dietterich TG (2004). Improving SVM accuracy by training on auxiliary data sources. Proceedings of the Twenty-First International Conference on Machine Learning. Morgan Kaufmann. p. 871–878.

Xu H, Agrafiotis DK (2002). Retrospect and prospect of virtual screening in drug discovery. Current Topics in Medicinal Chemistry 2: 1305–1320.

Xu J, Hagler A (2002). Chemoinformatics and drug discovery. Molecules 7: 566–600.

Xu L, Yang JA (1998). Chemical environment code and measure of molecular similarity. Computers in Chemistry 22: 393–398.

Yap CW, Cai CZ, Xue Y, Chen YZ (2004). Prediction of torsade-causing potential of drugs by support vector machine approach. Toxicological Sciences 79: 170–177.

Zhao YH, Abraham MH, Hersey A, Luscombe CN (2003). Quantitative relationship between rat intestinal absorption and Abraham descriptors. European Journal of Medicinal Chemistry 38: 939–947.

Zickus M, Greig AJ, Niranjan M (2002). Comparison of four machine learning methods for predicting $PM_{10}$ Concentrations in Helsinki, Finland. Water Air and Soil Pollution Focus 2: 717–729.

Zien A, Schölkopf B, Rätsch G, Mika S, Lengauer T (2000). Engineering Support Vector Machine Kernels That Recognize Translation Initiation Sites. Bioinformatics 16(9): 799–807.