Extrinsic Regularization in Parameter Optimization for Support Vector Machines

by
Matthew D. Boardman

Submitted in partial fulfillment of the
requirements for the degree of
Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
June 2006

DALHOUSIE UNIVERSITY

FACULTY OF FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "**Extrinsic Regularization in Parameter Optimization for Support Vector Machines** " by **Matthew D. Boardman** in partial fulfillment of the requirements for the degree of **Master of Computer Science**.

Dated: June 8, 2006

Supervisor: _____
Professor Thomas Trappenberg

Readers: _____
Professor Christian Blouin

_____
Professor Qigang Gao

DALHOUSIE UNIVERSITY

Date: **June 8, 2006**

Author: **Matthew D. Boardman**

Title: **Extrinsic Regularization in Parameter Optimization for Support Vector Machines**

Department or School: **Faculty of Computer Science**

Degree: **M.C.Sc.** Convocation: **October** Year: **2006**

Permission is herewith granted to Dalhousie University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

For my parents,

Robert and Christine.

Table of Contents

List of Tables

List of Figures

Abstract

A heuristic is proposed to optimize free parameter selection for Support Vector Machines, with the goals of improving generalization performance and providing greater insensitivity to training set selection. The main points of the proposed heuristic are the inclusion of extrinsic regularization to improve generalization error; the use of simulated annealing to improve parameter search efficiency in comparison to an exhaustive grid search; and an intensity-weighted centre of mass of the most optimum points to reduce solution volatility. Two standard classification problems are examined for comparison, and the heuristic is applied to protein sequence alignment quality and retinal electrophysiology classification. The heuristic is extended to univariate and multivariate regression problems, examining environmental modelling and periodic gene expression. Input variable selection and sensitivity are explored to determine the most significant segments of the electroretinography waveforms.

List of Abbreviations and Symbols Used

ε-SVR	ε-insensitive Support Vector Regression
ANN	Artificial Neural Network
ANOVA	Analysis of Variances
C-P	Coverage-Performance
ERG	Electroretinogram
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
i.i.d.	independently and identically distributed
KFD	Kernel Fisher Discriminants
KKT	Karush-Kühn-Tucker
KPCA	Kernel Principal Component Analysis
MSE	Mean Squared Error
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
SVC	Support Vector Classification
SVM	Support Vector Machine
SVR	Support Vector Regression
VC	Vapnik-Chervonenkis

Acknowledgements

The author would like to thank Dr. François Tremblay and Dr. Denis Falvey, Dalhousie University, for the retinal electrophysiology data analyzed in this thesis; Dr. Christian Blouin, Dalhousie University, for access to the sequence alignment data analyzed in this thesis; the genomic researchers at the Wellcome Trust Sanger Institute, Hinxton, Cambridge, for making their gene expression data publicly available; Dr. Gavin Cawley, University of East Anglia, for suggesting and holding the Predictive Uncertainty in Environmental Modelling Competition for the 2006 IEEE World Congress on Computational Intelligence; the anonymous reviewers of the IEEE, whose feedback for the published work related to this thesis provided many invaluable suggestions; Dr. Thomas Trappenberg and the Computational Neuroscience Group, Dalhousie University, for their helpful suggestions and many stimulating discussions; and the defence readers, Dr. Qigang Gao and Dr. Christian Blouin, for their constructive feedback. None of the aforementioned bear any responsibility for the inadequacies of this work.

Parts of Chapters 1 through 4 are also to be published in Boardman and Trappenberg (2006). Much of Chapter 6 was also submitted to Dr. Christian Blouin as the course project for *CSCI 6904: Special Topics in Computational Biology*, (Dalhousie University, Winter 2006).

This work was supported in part by the NSERC grant RGPIN 249885-03. MATLAB is a registered trademark of The MathWorks, Inc. Excel is a registered trademark of Microsoft Corporation. Pentium is a registered trademark of Intel Corporation.

Chapter 1

Introduction

In this thesis, we propose a heuristic approach for optimal selection of the free parameters of Support Vector Machines (SVM) (Vapnik, 1995), to improve generalization performance for classification and regression problems. The parameter optimization heuristic includes three main points:

- *Extrinsic Regularization*: SVM are based on the concept of intrinsic regularization (Tikhonov, 1963), for a set of free parameters chosen *a priori*. Here, we propose to take further regularization into account during parameter optimization, extrinsic to the SVM algorithm itself. We improve generalization performance of the selected model by considering a complexity penalty based on the number of support vectors employed in the model representation, in addition to the cross-validated prediction accuracy or mean squared error of the model over a set of known observations.

- *Simulated Annealing*: To improve computational efficiency while traversing the generalization error surface, we use the well-known method of simulated annealing (Kirkpatrick *et al.*, 1983). In contrast to a simple grid search, we show that such a stochastic strategy focuses the evaluated points on the areas of interest to a much greater extent, allowing high-precision results with fewer total evaluations.

- *Intensity-Weighted Centre of Mass*: We show that when using either a grid search or a stochastic search based on simulated annealing, the final selected points may be inconsistent across multiple runs. This is due to the volatile nature of the generalization error surface in non-separable problems, and is exacerbated by N-fold cross-validation. Previous work has found that the same problem exists for pattern search methods (Momma and Bennett, 2002), and has advocated the use of a mean or median of the selected points across several runs, thereby increasing computational complexity. Here, we advocate the calculation of a centre-of-mass of the most optimum points found during a single search through parameter space, weighted to favour those points which achieve the best performance.

The heuristic is applied to two well-known, benchmark classification problems and two real-world classification problems. We will focus on optimal selection of the cost parameter C, which controls the tradeoff between maximization of the margin width and minimizing the number of misclassified samples in the training set (Boser *et al.*, 1992), and the width γ of the Radial Basis Function (RBF) kernel (Cortes and Vapnik, 1995). We then extend the heuristic to Support Vector Regression (SVR) problems, in which we will also optimize the constant noise threshold ε in the ε-insensitive loss function (Vapnik *et al.*, 1997).

The goal of this heuristic is to improve generalization performance for volatile, noisy data sets with a low number of sample observations but a high number of input dimensions. For example, we will examine unprocessed, continuous waveforms from retinal electrophysiology (see for example Sutter and Tran, 1992), and mitotic gene-expression data from DNA microarrays (see for example Gilks *et al.*, 2005), both of which may contain a significant proportion of additive noise. We will explore the problem of input variable selection, another important part of model optimization, and perform a sensitivity analysis to determine the most significant segments of an input waveform.

Extrinsic regularization has been applied to Artificial Neural Networks (ANN) (see for example Haykin, 1999), but has not, to our knowledge, been applied to SVM during free parameter optimization. The main contribution of this thesis is the combined use of extrinsic regularization to improve generalization error, with two well-known techniques to reduce computational complexity, to form a practical heuristic favouring blind application.

1.1 Thesis Structure

This thesis is structured as follows.

In Chapter 2, we introduce some of the concepts which will be discussed in this thesis, give some historical and mathematical background on the development of SVM and discuss similar work related to parameter optimization. We will show the volatile nature of the generalization error surface to be traversed during parameter optimization. In Chapter 3, we describe the heuristic proposed and implemented in this work in further detail.

In Chapter 4, we examine the resulting classification performance for two standard machine learning data sets — the Iris Plants Database and Wisconsin Breast Cancer Database (Newman *et al.*, 1998) — and for real-life retinal electrophysiology and sequence alignment quality data sets obtained from research performed at Dalhousie University by other

groups. In Chapter 5, we extend the heuristic to univariate SVM regression, and apply the heuristic to three real-world environmental modelling problems. In Chapter 6, we further extend the heuristic to regression of gene expression data, with multiple input and multiple output dimensions, with a view to the imputation of missing data and the reduction of additive noise.

In Chapter 7, we examine input variable selection and sensitivity measures, in order to determine which parts of the electroretinogram (ERG) waveform explored in Chapter 4 are most significant in terms of classification performance. Finally, in Chapter 8, we conclude with a discussion of the results obtained in this thesis.

1.2 Conclusions and Future Work

Further analysis is warranted to determine the generality of this approach with a wider array of practical problems, and to compare the results of this heuristic with other parameter optimization methods such as Chapelle *et al.* (2002); Momma and Bennett (2002); Staelin (2003). Our visualization of the generalization error surface suggests a possible analytic solution, warranting further investigation into the nature and origin of the shape of this surface. Another logical further step in this area is waveform estimation, that is multivariate regression of observations $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_\ell, \mathbf{y}_\ell)$ where $\mathbf{x} \in \mathcal{X} = \mathbb{R}^d$ and $\mathbf{y} \in \mathcal{Y} = \mathbb{R}^d$. This may be applied to the problem of approximating the output waveform generated by an electrode directly attached to the optic nerve based on an input waveform generated by a corneal electrode in retinal electrophysiology, or to estimation of the signal characteristics of particular functional areas of the brain from continuous electroencephalography (EEG) signals.

Here we exclude detailed discussions on the intrinsic regularization performed by SVM during training, but rather focus on extrinsic regularization during parameter optimization. We also exclude application of the heuristic to the problems of distribution estimation, outlier detection or multi-class prediction, but instead focus on binary classification and ε-insensitive regression. We exclude implementation of other surface traversal methods such as a geometric pattern search (Momma and Bennett, 2002; Press *et al.*, 1992), instead comparing our heuristic employing simulated annealing with a high-resolution grid search over the same parameter space: we focus on the number of evaluations performed in parameter space, rather than using formal algorithm analysis.

We conclude that support vector machines include robust intrinsic regularization, but naïve choices of the free parameters will often result in unacceptable generalization error. Appropriate selection of free parameters is essential to achieving high performance. By including extrinsic regularization in the optimization of free parameters, we propose an approach that balances model complexity with classification or regression error.

Chapter 2

Background

Support Vector Machines (SVM) (Boser *et al.*, 1992; Cortes and Vapnik, 1995; Vapnik, 1995) map a set of observations from input space into a higher-dimensional feature space using a non-linear transformation, then find a hyperplane in this feature space which optimally separates the known observations by minimizing empirical risk.

In 1968, two Soviet scientists, Vladimir N. Vapnik and Alexey Ja. Chervonenkis, proposed a philosophy for pattern recognition based on statistical learning theory in an article originally titled, О равномерной сходимости частот появления событий к их вероятностям (Vapnik and Chervonenkis, 1968) and later translated into English as, *On the uniform convergence of relative frequencies of events to their probabilities* (Vapnik and Chervonenkis, 1971). This article led to a statistical theory of pattern recognition (Vapnik and Chervonenkis, 1974), also known today as *Vapnik-Chervonenkis Theory* or VC Theory, and later lead to the development of SVM. However, since these works were published in Russian within Soviet academic publications, the work was not generally known in western academia due to the political pressures of that era.

With the fall of the former Soviet Union in the early 1990s, many Russian scientists began moving to western institutions. Vapnik started working for AT&T Bell Labs in 1991, and in 1995 he became a professor of Computer Science and Statistics at the Royal Holloway, University of London (Royal Holloway, 2006). While at AT&T, Vapnik and his colleagues developed the SVM formulation as an optimum-margin classifier for separable data, applied to the problem of handwriting recognition (Boser *et al.*, 1992) based on his earlier work with Chervonenkis. This work was later developed for non-separable data sets using a soft-margin hyperplane (Cortes and Vapnik, 1995). Soon after, Vapnik published the first edition of *The Nature of Statistical Learning Theory* (Vapnik, 1995), summarizing statistical learning philosophy and the formulation of SVM for classification, regression and density estimation.

Today, the popularity of the SVM is growing in a wide variety of applications. *The SVM Applications List* (Guyon, 2006), a web site maintained by Isabelle Guyon since 1999, encourages user submissions to describe the growth of SVM usage in many disciplines.

Table 2.1: List of notation used in the formulation of an SVM.

Symbol	Meaning
$\mathbf{x}$	A vector of inputs for known observations
$\mathcal{X}$	The set from which inputs are drawn
y	A target for known observations
$\mathcal{Y}$	The set from which targets are drawn
ℓ	The total number of known observations
d	The number of dimensions in the input observations
$\mathbb{R}^d$	The set of all real numbers, in d dimensions
n_{sv}	The number of observations defined as support vectors during training
$f()$	An unspecified function, drawn from the set of functions $\mathcal{F}$
$K(\mathbf{x}, \mathbf{x}')$	An unspecified SVM kernel function
α_{sv}	Kühn-Tucker coefficients of support vectors
b	Offset threshold of SVM hyperplane
C	Cost parameter of an SVM
γ	Width parameter of an SVM with RBF kernel
ε	Insensitivity-tube width of ε-SVR

Many supervised-learning problems in the fields of pattern recognition (Degroeve *et al.*, 2005; Vapnik, 1995), medical research (Miller *et al.*, 2003; Wang *et al.*, 2005), economics (Rueda *et al.*, 2004) and bioinformatics (Shan *et al.*, 2003; Wang *et al.*, 2006), which frequently use artificial neural networks (ANN) to evaluate inputs with continuous values and targets with binary or continuous values, may be ideal candidates for SVM classification and regression. SVM technology has been integrated into commercial data mining software such as the Oracle Data Miner (ODM), an optional component of Oracle Database 10g (Milenova *et al.*, 2005). Support Vector Classifiers (SVC) have even been mentioned in popular media, such as the CBS drama *Numb3rs* (CBS Corporation, 2006) in which the statistical learning philosophy was compared to Michelangelo's philosophy of sculpting: chipping away at the irrelevant marble (or irrelevant dimensions and observations) until only the statue (the statistically-ideal model) remains.

2.1 The SVM Formulation

In the following sections, we will examine the formulation of an SVM based on the minimization of empirical risk, then describe the importance of extrinsic regularization and show related work relevant to the optimization of SVM free parameters. A summary of the

notation used in this section is shown in Table 2.1.

2.1.1 Supervised Learning

Suppose we are given a set of ℓ observations

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell) \tag{2.1}$$

with inputs $\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^d$, $i = 1, \ldots, \ell$ that indicate targets $y_i \in \mathcal{Y}$. In the general problem of supervised learning, our goal is to find a function $f(\mathbf{x})$ in the set of functions $\mathcal{F}$ which minimizes a loss functional on future observations (Chapelle *et al.*, 2002). For example, we may wish to find a function that minimizes the binary classification error where $\mathcal{Y} = \{-1, +1\}$, or minimizes the mean squared regression error where $\mathcal{Y} = \mathbb{R}$.

Specifically, we wish to minimize the *risk functional* (Vapnik, 1995)

$$R(\alpha) = \int L(y, f(\mathbf{x}, \alpha))\, dF(\mathbf{x}, y) \tag{2.2}$$

where $L(y, f(\mathbf{x}, \alpha))$ describes the *loss* between the expected targets y and a function $f(\mathbf{x}, \alpha)$ which predicts the targets from a set of inputs $\mathbf{x}$ and some set of model parameters α. We integrate over the joint probability distribution $F(\mathbf{x}, y)$ to find the risk. However, in practice, this distribution is rarely known.

For example, in SVM binary classification, this loss functional may be written as (Vapnik, 1995)

$$L(y, f(\mathbf{x}, \alpha)) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}, \alpha), \\ 1 & \text{otherwise.} \end{cases} \tag{2.3}$$

whereas in SVM regression, the loss functional becomes (Vapnik, 1995)

$$L(y, f(\mathbf{x}, \alpha)) = (y - f(\mathbf{x}))^2 \tag{2.4}$$

Since the joint probability distribution $F(\mathbf{x}, y)$ of the inputs and targets of the observations is unknown, we estimate this risk functional using the observations, to create an empirical risk functional (Vapnik, 1995). In the case of classification, we discretize Equation 2.2 with the loss functional from Equation 2.3, to create the binary classification empirical risk functional (Vapnik, 1995)

$$R_{emp}(\alpha) = \frac{1}{2\ell} \sum_{i=1}^{\ell} |y_i - f(\mathbf{x}_i, \alpha_i)| \tag{2.5}$$

or *classification error*, whereas in regression, we use Equation 2.4 to form the regression empirical risk functional (Vapnik, 1995)

$$R_{emp}(\alpha) = \frac{1}{\ell} \sum_{i=1}^{\ell} |y_i - f(\mathbf{x}_i, \alpha_i)|^2 \tag{2.6}$$

or *mean squared error*.

2.1.2 A Quadratic Optimization Problem

Let us now consider the problem of binary classification, where the targets of each observation $y_i \in \mathcal{Y} = \{-1, +1\}$. In the separable case, where there is no overlap between the distributions of each class, we construct a hyperplane $\mathbf{w}$ to optimally separate the two classes as (Vapnik, 1995)

$$(\mathbf{w} \cdot \mathbf{x}) - b = 0 \tag{2.7}$$

where b is an offset threshold. The *margin* $\Delta = 1 / |\mathbf{w}|$ of the hyperplane then allows classification of the inputs of the observations $\mathbf{x}$ as (Vapnik, 1995)

$$y = \begin{cases} 1 & \text{if } (\mathbf{w} \cdot \mathbf{x}) - b \geq \Delta \\ -1 & \text{if } (\mathbf{w} \cdot \mathbf{x}) - b \leq -\Delta \end{cases} \tag{2.8}$$

The optimal hyperplane is that which maximizes this margin Δ (Boser *et al.*, 1992), as illustrated in Figure 2.1. In order to find this optimal hyperplane, we minimize the functional (Vapnik, 1995)

$$\Phi(\mathbf{w}) = \frac{1}{2} (\mathbf{w} \cdot \mathbf{w}) \tag{2.9}$$

subject to the set of inequality constraints (Vapnik, 1995)

$$y_i [(\mathbf{x}_i \cdot \mathbf{w}) - b] \geq 1, \quad i = 1, \ldots, \ell \tag{2.10}$$

This optimization problem can be solved by minimizing the following Lagrangian objective function (Vapnik, 1995), also referred to as the *primal* problem (Burges, 1998; Smola and Schölkopf, 2004)

$$\mathrm{L}(\mathbf{w}, b, \alpha) = \frac{1}{2} (\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^{\ell} \alpha_i ([(\mathbf{x}_i \cdot \mathbf{w}) - b] y_i - 1) \tag{2.11}$$

with respect to $\mathbf{w}$ and b so that we obtain minimum complexity (Tikhonov, 1963), and maximizing with respect to the Lagrange multipliers $\alpha_i \geq 0$ so that we obtain a maximal margin.

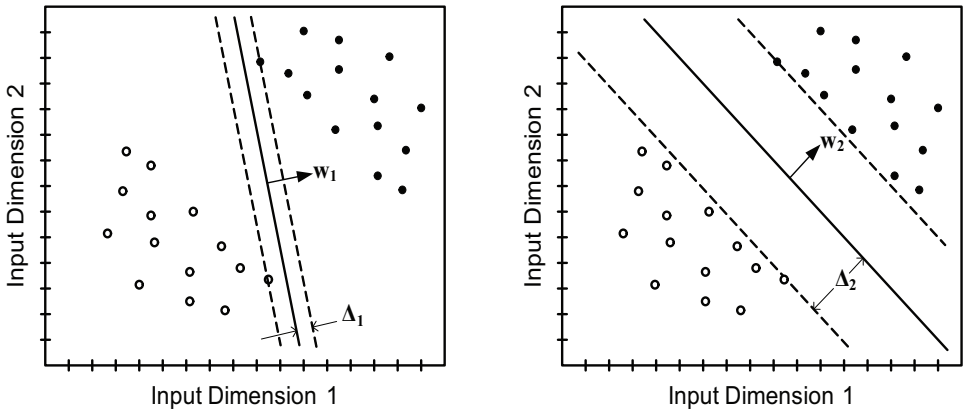


Figure 2.1: A cartoon based on illustrations in Bennett and Campbell (2000), of the maximal margin for a linearly-separable data set with two classes ($\circ$ and $\bullet$). On the *left*, we see a hyperplane $\mathbf{w}_1$ separating the classes. However, the margin Δ_1 (dashed lines), the distance between the hyperplane and the nearest samples of each class, is quite small. On the *right*, we see a hyperplane $\mathbf{w}_2$ with a much larger margin Δ_2. Both hyperplanes separate the samples perfectly, but it is clear that $\mathbf{w}_2$ is much more likely to correctly classify future samples in this simple distribution.

The Karush-Kühn-Tucker (KKT) conditions for solving this primal problem may be stated as (Burges, 1998)

$$\begin{aligned}
\frac{\partial \mathrm{L}}{\partial w_\nu} = w_\nu - \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_{i\nu} = 0, \qquad & i = 1, \ldots, \ell, \qquad \nu = 1, \ldots, d \\
\frac{\partial \mathrm{L}}{\partial b} = -\sum_{i=1}^{\ell} \alpha_i y_i = 0, \qquad & i = 1, \ldots, \ell \\
\alpha_i \left([(\mathbf{x}_i \cdot \mathbf{w}) - b] y_i - 1 \right) = 0, \qquad & i = 1, \ldots, \ell \\
\alpha_i \geq 0, \qquad & i = 1, \ldots, \ell
\end{aligned} \tag{2.12}$$

Since this is a convex problem, the KKT conditions are *necessary* and *sufficient* for $\mathbf{w}, b, \alpha$ to be a solution of the primal problem (Burges, 1998). The optimal hyperplane forming the solution to the primal problem, with these KKT conditions, lies at the saddle point at which $\partial \mathrm{L}/\partial b = \partial \mathrm{L}/\partial \mathbf{w} = 0$ (Vapnik, 1995). We formulate this optimal hyperplane as (Vapnik, 1995)

$$\mathbf{w} = \sum_{i=1}^{\ell} y_i \alpha_i \mathbf{x}_i = 0, \quad \alpha_i \geq 0, \quad i = 1, \ldots, \ell \tag{2.13}$$

We define a set of *support vectors* as inputs for those observations for which we achieve equality in Equation 2.10, such that $\alpha_{sv} > 0$ (Burges, 1998). The optimal hyperplane may

then be reformulated as (Vapnik, 1995)

$$\mathbf{w} = \sum_{\text{support vectors}} y_i \alpha_i \mathbf{x}_i = 0, \quad \alpha_i > 0 \tag{2.14}$$

Combining these new conditions with the primal Lagrangian in Equation 2.11, we obtain the *Wolfe dual* objective function (Burges, 1998)

$$\mathrm{W}(\mathbf{w}, b, \alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \tag{2.15}$$

which should be maximized subject to the conditions (Vapnik, 1995)

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0, \qquad \alpha_i \geq 0, \qquad i = 1, \ldots, \ell \tag{2.16}$$

This dual formulation of the optimization problem removes the dependence of the vector $\mathbf{w}$ (Boser *et al.*, 1992) in the primal formulation from Equation 2.11, and has much simpler constraints (Vapnik, 1995). There is a single, global maximum forming the solution, with no local extrema. This convex problem can quickly be solved using quadratic optimization methods such as Sequential Minimum Optimization (SMO) (Platt, 1998), and is a large part of the reason behind the great computational efficiency of the SVM formulation.

Given the solution α_{sv} to this quadratic optimization, classification of any future observation $\mathbf{x}$ is then made from (Burges, 1998; Vapnik, 1995)

$$f(\mathbf{x}) = \text{sign}\left(\sum_{\text{support vectors}} y_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}) - b \right) \tag{2.17}$$

2.1.3 The Non-Separable Case

In the non-separable case, where there may be some overlap between the two distributions of opposite class, we wish to construct a *soft margin* hyperplane (Cortes and Vapnik, 1995; Vapnik, 1995) which allows some number of observations to be misclassified in order to optimally separate the remaining observations.

We introduce non-negative *slack* (Burges, 1998) variables $\xi_i \geq 0$ to Equation 2.9, such that (Vapnik, 1995)

$$\Phi(\mathbf{w}, \xi) = \frac{1}{2} \left(\mathbf{w} \cdot \mathbf{w}\right) + C \left(\sum_{i=1}^{\ell} \xi_i \right) \tag{2.18}$$

where C is a free parameter provided by the user, referred to as the *cost* parameter (Vapnik, 1995), that imposes a penalty on misclassified samples (Burges, 1998).

We now wish to minimize Equation 2.18 subject to the constraint from Equation 2.10 which, with the introduction of these slack variables, becomes (Vapnik, 1995)

$$y_i \left[(\mathbf{x}_i \cdot \mathbf{w}) - b \right] \geq 1 - \xi_i, \quad i = 1, \ldots, \ell \tag{2.19}$$

Since we have added a series of constants, the dual formulation of the optimization problem remains the same, but we must now maximize Equation 2.15 subject to

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, \ell \tag{2.20}$$

which is as Equation 2.16, but with a new upper bound on the Lagrangian multipliers. Classification on future observations is then performed as above, from Equation 2.17.

2.1.4 The Kernel Trick

To this point, we have considered only the input space: the space in which the inputs of the observations are defined. However, the SVM formulation allows us to translate this input space into a higher-dimensional *feature* space (Boser *et al.*, 1992; Cortes and Vapnik, 1995), then determine the optimum soft-margin hyperplane in this feature space using the dual Lagrangian formulation as above, without the need to determine the transformation itself (Müller *et al.*, 2001; Vapnik, 1995).

This can be done easily, since we require only the convolution between two input vectors in feature space, described by the dot product in Equations 2.15 and 2.17. If the mapping between input space and feature space is given by

$$\Psi : \mathbb{R}^d \mapsto \mathbb{R}^D \tag{2.21}$$

where d is the dimensionality of the input space and D of the feature space, then the dot product $\mathbf{u} \cdot \mathbf{v}$ in input space is given in feature space by (Cortes and Vapnik, 1995)

$$\Psi(\mathbf{u}) \cdot \Psi(\mathbf{v}) \equiv K(\mathbf{u}, \mathbf{v}) \tag{2.22}$$

which is defined to be a *kernel function* (Boser *et al.*, 1992). This kernel function has the Hilbert-Schmidt expansion (Burges, 1998; Cortes and Vapnik, 1995)

$$K(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{\infty} \lambda_i \, \Psi_i(\mathbf{u}) \cdot \Psi_i(\mathbf{v}) \tag{2.23}$$

Table 2.2: The following non-linear kernels (Burges, 1998; Müller *et al.*, 2001; Smola and Schölkopf, 2004; Vapnik, 1995) are commonly used to perform a dot product in mapped feature space in the SVM formulation.

Name	Parameters	Kernel Function
Polynomial	$c \in \mathbb{R}$, $p \in \mathbb{N}$	$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + c)^p$
Radial Basis Function	$\gamma \in \mathbb{R}$	$K(\mathbf{x}, \mathbf{x}') = e^{-\gamma \left\|\mathbf{x}-\mathbf{x}'\right\|^2}$
Sigmoidal or ANN	$\kappa \in \mathbb{R}$, $\delta \in \mathbb{R}$	$K(\mathbf{x}, \mathbf{x}') = \tanh(\kappa(\mathbf{x} \cdot \mathbf{x}') - \delta)$

where $\lambda_i \in \mathbb{R}$. To ensure that the expansion coefficients λ_i are positive in this expansion, which is sufficient to ensure that the kernel defines a dot product in feature space (Cortes and Vapnik, 1995) as Equation 2.22, the kernel function must satisfy Mercer's condition. Briefly, this states that (Cortes and Vapnik, 1995)

$$\iint K(\mathbf{u}, \mathbf{v})\, g(\mathbf{u})\, g(\mathbf{v})\ \mathrm{d}\mathbf{u}\, \mathrm{d}\mathbf{v} \quad > \quad 0 \tag{2.24}$$

must be satisfied for any function $g(\mathbf{x})$ with a finite L_2 norm, that is all functions $g(\mathbf{x})$ which satisfy (Cortes and Vapnik, 1995)

$$\int g(\mathbf{x})^2\ \mathrm{d}\mathbf{x} \quad < \quad \infty \tag{2.25}$$

Three kernels, summarized in Table 2.2, have been shown to satisfy Mercer's condition and are commonly used with SVM. The sigmoidal kernel will only satisfy Mercer's condition for particular values of the free parameters (Burges, 1998; Smola and Schölkopf, 2004), but has been used successfully in practice (Vapnik, 1995). The polynomial kernel, of degree p, is inhomogeneous in that it allows the additive constant c to be larger than zero (Boser *et al.*, 1992; Smola and Schölkopf, 2004) for additional degrees of freedom.

The RBF kernel is translation invariant, that is $K_\gamma(\mathbf{x}, \mathbf{x}') = K_\gamma(\mathbf{x} - \mathbf{x}')$ (Smola and Schölkopf, 2004), and has an infinite number of dimensions (Vapnik, 1995). Another significant advantage of the RBF kernel is that it adds only a single free parameter $\gamma > 0$, which controls the width of the RBF kernel as $\gamma = 1/2\sigma^2$, where σ^2 is the variance of the resulting Gaussian hypersphere. The RBF kernel has been shown to perform well in a wide variety of practical applications (see for example Degroeve *et al.*, 2005; Hsu *et al.*, 2003; Wang *et al.*, 2005).

With this kernel replacing the dot product, the classifier in Equation 2.17 becomes

$$f(\mathbf{x}) = \mathrm{sign}\left(\sum_{\text{support vectors}} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) - b\right) \tag{2.26}$$

This so-called *kernel trick* (Müller *et al.*, 2001) is quite powerful: a problem which is non-separable in input space may have a transformation into an unknown feature space in which the problem becomes linearly separable (Burges, 1998). This kernel trick has also been proposed for many other mathematical techniques, such as probability density estimation using SVM (Vapnik, 1995), Kernel Principal Component Analysis (KPCA) (Müller *et al.*, 2001; Schölkopf *et al.*, 1999b) and Kernel Fisher Discriminants (KFD) (Müller *et al.*, 2001). Other kernels have also been proposed, such as the multiquadric and inverse multiquadric kernels (Müller *et al.*, 2001), and the linear or non-linear spline with a finite or infinite number of nodes (Vapnik, 1995).

In practice, a kernel is chosen *a priori* based on the problem at hand: in this thesis we will primarily use the popular Radial Basis Function (RBF) kernel.

2.1.5 The Vapnik-Chervonenkis Dimension

The SVM formulation allows an estimate of the capacity of the resulting classifier. In classification, we consider the capacity to be the number of observations which can be *shattered* by a family of hyperplanes (Burges, 1998; Vapnik, 1995), as illustrated in Figure 2.2. This capacity measure is called the *Vapnik-Chervonenkis dimension*, or VC dimension.

The VC dimension h is simply measured by the degrees of freedom in the formulation, that is $n + 1$ where n is the number of dimensions available (Vapnik, 1995). However, in practice, the VC dimension may be considerably lower: it is bounded by the inequality (Vapnik, 1995)

$$h \leq \min\left(\frac{R^2}{\Delta^2}\,,\ n\right) + 1 \tag{2.27}$$

where R is the radius of the minimum sphere in feature space that will enclose all inputs $\mathbf{x}_i$, and $\Delta = 1 \,/\, |\mathbf{w}|$ is the margin. This allows us to estimate the VC dimension of an SVM for a given set of observations as simply $R^2\,|\mathbf{w}|^2$ (Vapnik, 1995).

The VC dimension is useful for determining a bound on the expected error: a smaller VC dimension will lead to a smaller estimate of the probability of misclassifying a sample (Burges, 1998). Specifically, with probability $1 - \eta$, the probability that any given test observation is classified incorrectly is (Vapnik, 1995)

$$P_{error} \leq \frac{m}{\ell} + \frac{\hat{\varepsilon}}{2}\left(1 + \sqrt{1 + \frac{4m}{\ell\hat{\varepsilon}}}\right) \tag{2.28}$$

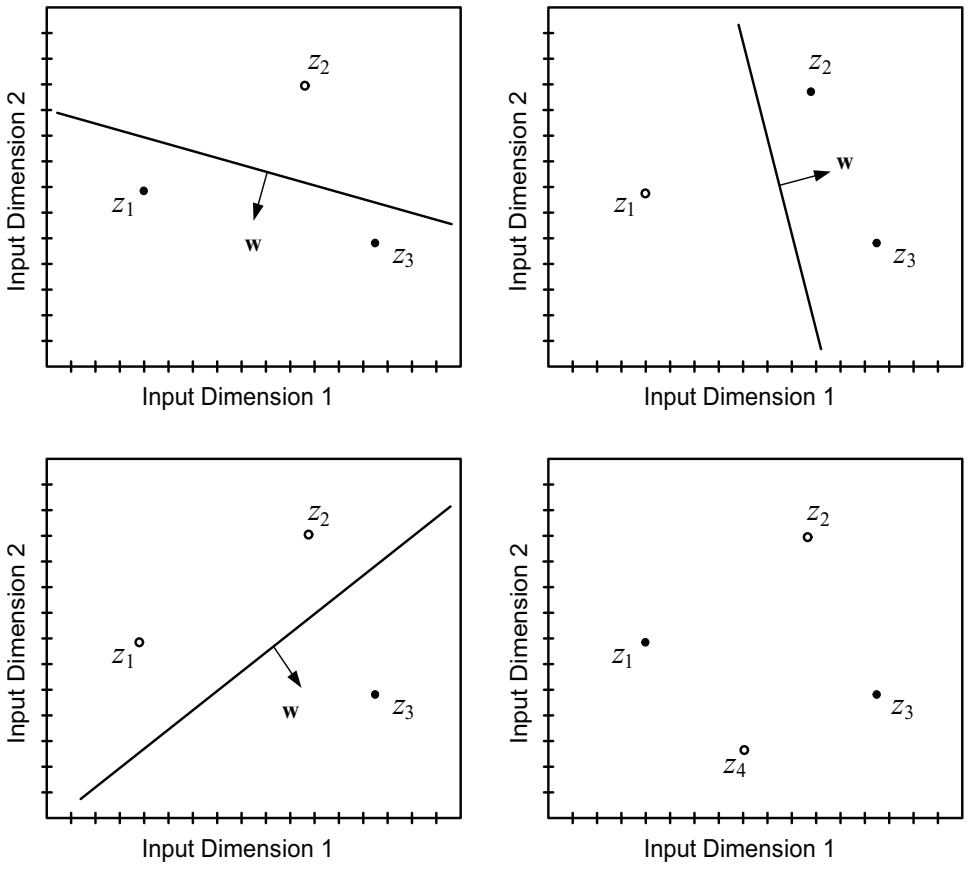


Figure 2.2: A cartoon based on illustrations in Burges (1998); Hastie *et al.* (2001); Vapnik (1995), showing the concept of a hyperplane *shattering* a set of points. A hyperplane w in this two-dimensional feature space can always separate binary classes for any three observations z_i, regardless of how those observations are labelled. Three example class distributions are shown (note that there are actually $2^\ell = 8$ possible class-label combinations (Burges, 1998); only three are shown here for clarity). But two-dimensional space is insufficient when a fourth observation is added, as shown in the *bottom right*: for example, z_2 and z_4 cannot be separated from z_1 and z_3 when the class labels are as shown. The VC dimension of this set of hyperplanes is therefore $n + 1 = 3$, as we would expect from Equation 2.27.

where (Vapnik, 1995)

$$\hat{\varepsilon} = 4\,\frac{h\,\left(\ln\frac{2\ell}{h}+1\right)-\ln\frac{\eta}{4}}{\ell} \tag{2.29}$$

and m is the number of misclassified training observations.

2.1.6 ε-Insensitive Support Vector Regression

An SVM for regression or function estimation can be constructed in a similar fashion, using the loss function described by Equation 2.4 to minimize the empirical risk in Equation 2.6 which describes the mean squared error.

However, a powerful technique suggested in Vapnik (1995), and later refined in Drucker *et al.* (1997); Smola (1996); Smola and Schölkopf (2004); Vapnik *et al.* (1997), is the use of a loss function which is insensitive to a small, constant amount of additive noise, described by a free parameter ε. This so-called ε-*insensitive*, linear loss function is described as (Vapnik, 1995)

$$L(y, f(\mathbf{x},\alpha)) = \begin{cases} 0 & \text{if } |y - f(\mathbf{x},\alpha)| \leq \varepsilon \\ |y - f(\mathbf{x},\alpha)| - \varepsilon & \text{otherwise.} \end{cases} \tag{2.30}$$

and is shown in Figure 2.3. Observations that are within ε from the predicted value of the SVM will have zero loss, allowing the SVM to tolerate a small amount of noise without that noise affecting the resulting model. Outside of this tube of width 2ε, the loss grows linearly according to the cost parameter C, as a function of the absolute distance between the observation and prediction. A quadratic ε-insensitive loss function has also been proposed as simply the square of Equation 2.30, however the solution becomes somewhat more complex (Vapnik, 1995).

In SVM classification, the cost parameter C controls the tradeoff between classification accuracy and preserving a large margin width. However, in SVM regression, the cost parameter controls the tradeoff between the *flatness* (described by the norm of the hyperplane $\|\mathbf{w}\|^2$) of the function resulting from the regression model, and the number and magnitude of deviations larger than ε that will be allowed (Smola and Schölkopf, 2004).

In a similar manner to the introduction of slack variables in the objective function for non-separable data in Equation 2.18, we introduce two non-negative slack variables ξ_i, ξ_i^*

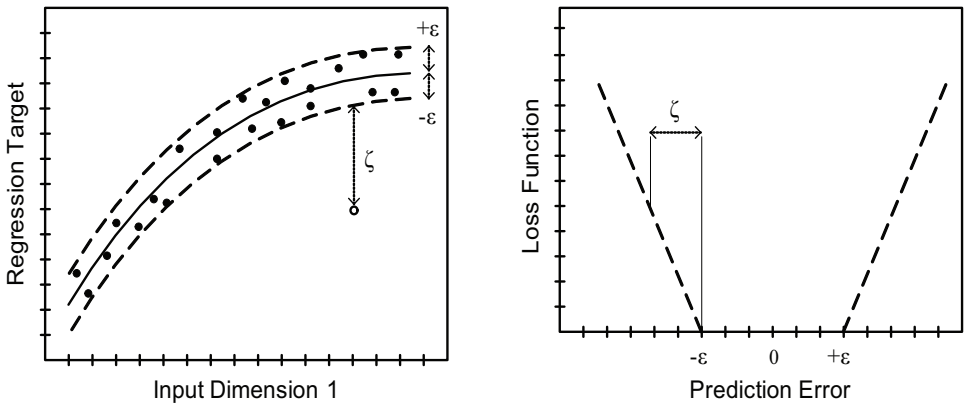


Figure 2.3: A cartoon based on illustrations in Bennett and Campbell (2000); Smola and Schölkopf (2004); Vapnik (1995), showing the linear, ε-insensitive loss function and its use for a small toy data set. If the error between a particular observation's target value and that observation's predicted value is less than ε, the loss function is zero. If, however, the prediction error is outside this "tube" of width 2ε, the loss function grows linearly according to the difference ζ. On the *left*, we a small number of samples ($\bullet$) with a single outlier ($\circ$). The outlier is a distance $\zeta + \varepsilon$ from the predicted target value, and so will be assigned a loss as shown on the *right*. The loss of the remaining points, however, will be zero, as the target values for each of the remaining observations fit within the ε bounds.

(corresponding to the positive and negative parts of the ε-tube) to form the functional (Vapnik, 1995)

$$\Phi(\mathbf{w}, \xi, \xi^*) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) + C\left(\sum_{i=1}^{\ell} \xi_i + \sum_{i=1}^{\ell} \xi_i^*\right) \tag{2.31}$$

which should be minimized according to the constraints (Vapnik, 1995)

$$\begin{aligned} y_i - [(\mathbf{w} \cdot \mathbf{x}_i) + b] &\leq \varepsilon + \xi_i^*, \quad i = 1, \ldots, \ell \\ (\mathbf{w} \cdot \mathbf{x}_i + b) - y_i &\leq \varepsilon + \xi_i, \quad i = 1, \ldots, \ell \\ \xi_i, \xi_i^* &\geq 0, \quad i = 1, \ldots, \ell \end{aligned} \tag{2.32}$$

From these, we desire to find the vector (Vapnik, 1995)

$$\mathbf{w} = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i)\mathbf{x}_i \tag{2.33}$$

where the solution is defined by two sets of Lagrange multipliers α and α^* in the objective

function (Vapnik, 1995)

$$\begin{aligned} \mathrm{W}(\alpha, \alpha^*) = & - \varepsilon \sum_{i=1}^{\ell} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{\ell} y_i (\alpha_i^* - \alpha_i) \\ & - \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(\mathbf{x}_i \cdot \mathbf{x}_j) \end{aligned} \tag{2.34}$$

subject to the constraints (Vapnik, 1995)

$$\begin{aligned} \sum_{i=1}^{\ell} \alpha_i^* &= \sum_{i=1}^{\ell} \alpha_i \\ 0 \leq \alpha_i^* \leq C, & \qquad i = 1, \ldots, \ell \\ 0 \leq \alpha_i \leq C, & \qquad i = 1, \ldots, \ell \end{aligned} \tag{2.35}$$

Future targets may then be predicted from (Smola and Schölkopf, 2004)

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i)(\mathbf{x}_i \cdot \mathbf{x}) + b \tag{2.36}$$

2.2 The Importance of Generalization

SVM typically deliver excellent accuracy, computational efficiency and generalization performance, with a sparse model representation and few free parameters (Bennett and Campbell, 2000; Burges, 1998; Hsu *et al.*, 2003; Vapnik, 1995). However, when these free parameters are improperly selected, just as with neural networks or any other classifier, SVM will yield poor generalization performance and poor computational efficiency.

One of the strengths of the SVM lies in the ability to form a general solution from a low number of samples. In Figure 2.4, we see an illustration of the necessity of generalization in classification. A linear classifier (the dashed line) and a polynomial classifier (the solid line) both classify the small subset of observations shown on the left, but the polynomial *overfits* the data in order to eliminate any classification error. As more observations are added on the right, it is clear that the simpler model is actually the correct one. This extreme example shows the importance of achieving a tradeoff between classification accuracy and generalization performance.

In the case of regression problems, generalization is just as important. In Figure 2.5, we show a real bioinformatics univariate regression problem, based on the mitotic-gene expression data examined in Chapter 6. All four examples perfectly fit the data, with zero

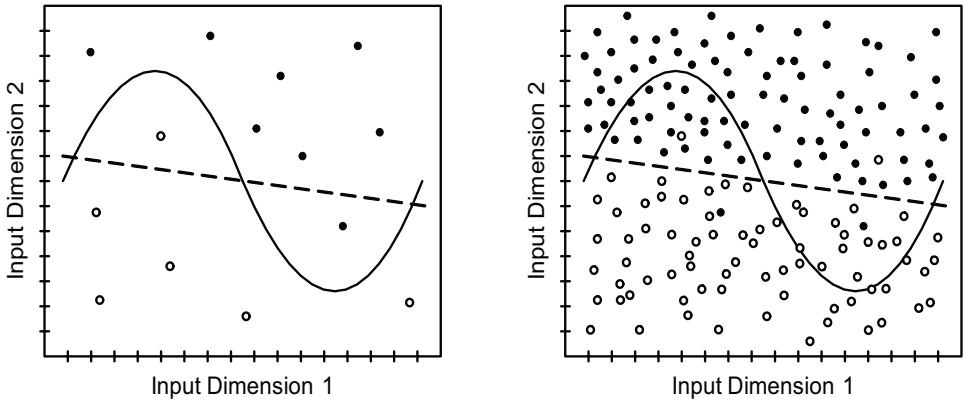


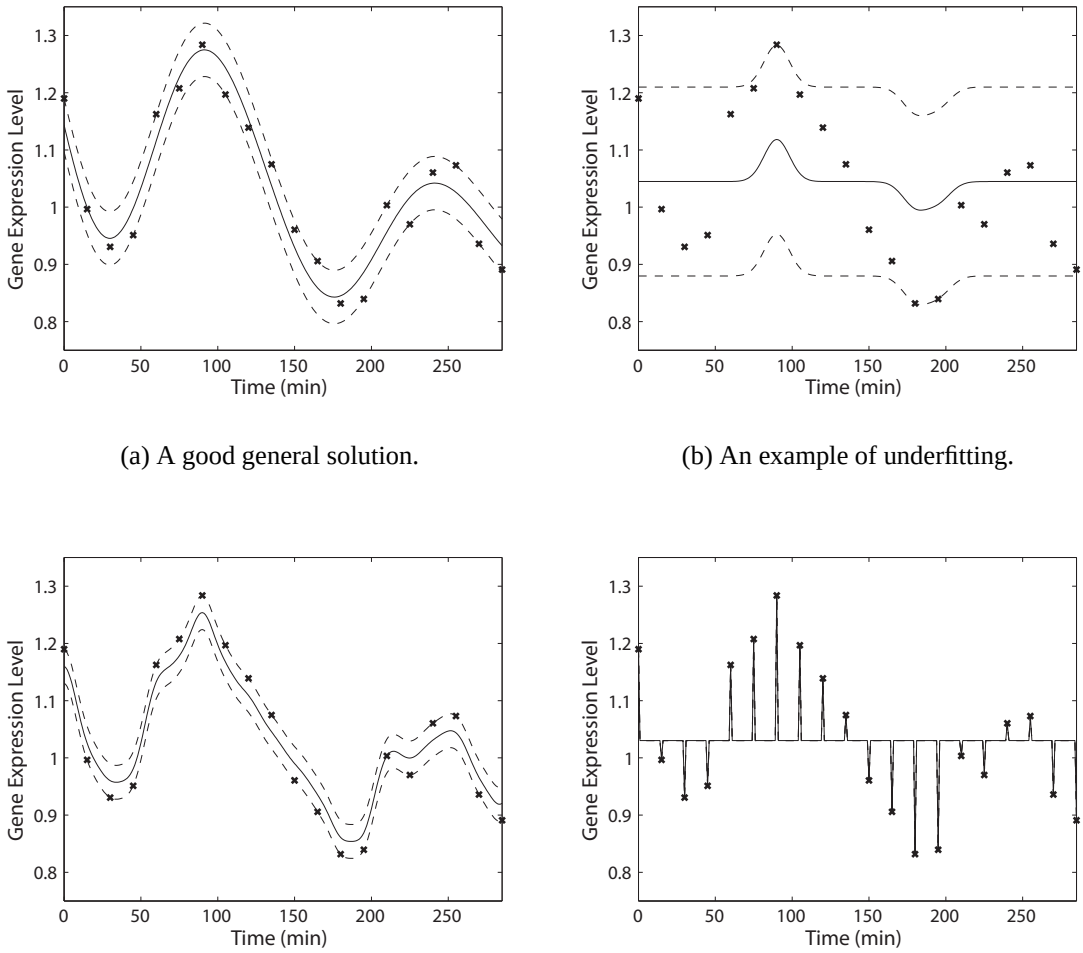
Figure 2.4: A cartoon based on illustrations in Müller *et al.* (2001), showing the importance of generalization. On the *left*, we see binary classification problem with a low number of observations (○ and ●). Both the polynomial (solid line) or linear (dashed line) decision boundaries separate the two classes, but which is correct? The polynomial achieves greater accuracy (100%), whereas the linear hyperplane allows two samples to be misidentified. As more observations are added on the *right*, however, we see that the linear classifier is actually the correct one: in this case, we are better off with the simpler linear model, even though it did not perform as well on the original training set in terms of classification accuracy.

mean squared error, since in all four cases each observation fits within the specified ε-insensitive tube. But which is correct? It is clear that the model in (d) is overfitting, as any future observations would be given simply the mean value of the observations. Since we expect periodicity in this data set, we might see (a) as the ideal solution. But both (b) and (c) might be correct as well. This example, with thousands of such genes to analyze, shows the importance of having an automatic mechanism to balance the tradeoff between allowing regression outliers and achieving a low mean squared error.

2.3 Free Parameter Selection

Initially, Vapnik (Boser *et al.*, 1992; Cortes and Vapnik, 1995; Vapnik, 1995) recommended direct setting of the kernel parameters and cost function by experts, based on knowledge of the particular data set to be evaluated. However, in practical situations, such *a priori* knowledge may well not be available. An automatic method is therefore desired to optimize selection of the free parameters, to obtain the desired level of accuracy and generalization performance for any given supervised learning problem.

Grid searches over an arbitrary range of parameter values are a common technique



(a) A good general solution.

(b) An example of underfitting.

(c) An example of overfitting.

(d) Severe overfitting.

Figure 2.5: Inappropriate selection of the free model parameters in ε-SVR is likely to lead to improperly modelled data. In this figure, we see examples from a real bioinformatics data set describing the periodic expression of mitotic genes in fission yeast, which is examined in Chapter 6. Four regression models for the C222.06 gene are shown. Incidentally, this gene was not identified as periodic in prior analyses of this data (Gilks *et al.*, 2005; Rustici *et al.*, 2004) due to its small change in expression, however our model does find statistically significant periodic activity. (a) An ε-SVR model trained using the heuristic proposed in this thesis, adapted for SVM regression. (b) An example of underfitting: in this case, the ε-tube width, corresponding to the expected noise level, is too high. Such a model overgeneralizes, ignoring all but the most extreme values in the training data. (c) An example of overfitting: in this case, the C cost parameter is too high, disallowing outliers such that every observed point must be within the ε-tube bounds. A model such as this contains many support vectors and is not likely to generalize well. (d) An example of severe overfitting: in this case the model fits all observed points, but such a model would be useless for imputing missing observations.

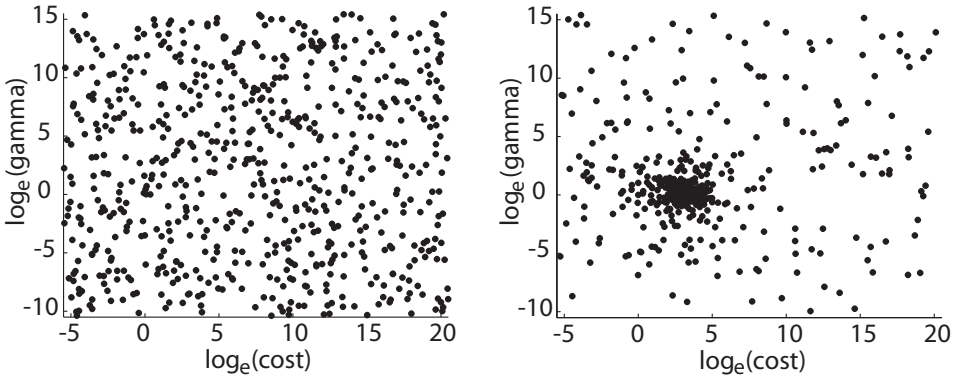


Figure 2.6: Comparison of the positions of evaluated points in a uniform random search pattern (*left*) and a stochastic algorithm based on simulated annealing (*right*). Although both searches contain exactly 660 evaluations, the simulated annealing approach focuses on the area of interest to a much greater extent. These figures were created from the Protein Sequence Alignment Quality data set (Inadequate vs. other) examined later in this thesis. An example of the stochastic path followed through three-dimensional parameter space for regression problems is shown in Figure 5.1.

when such knowledge is unavailable (Hsu *et al.*, 2003; Schölkopf *et al.*, 1999a; Staelin, 2003). However, such searches may be computationally expensive, and the precision of the results is subject to the chosen granularity of the grid. In Chapelle *et al.* (2002); Fan *et al.* (2005), gradient descent methods are proposed based on the minimizing the generalization error, allowing a larger number of parameters to be considered. However, in practical problems such methods may be affected by the presence of local extrema (Imbault and Lebart, 2004). This effect may be exacerbated in N-fold cross-validation from random partitioning of the training data. Leave-one-out cross-validation (where $N = \ell$) helps to reduce the effects of local extrema through the complete evaluation of all permutations of the training set at each point in the parameter search (Momma and Bennett, 2002), but this becomes computationally prohibitive when ℓ is large. In Friedrichs and Igel (2004), an evolutionary approach to SVM parameter optimization employs a genetic algorithm based on a covariance matrix, in order to greatly increase the number of hyperparameters which may be considered. In Momma and Bennett (2002), a nearest-neighbor sampling pattern is progressively evaluated as an alternative to gradient descent, but due to the volatile nature of the evaluated surface, the average of multiple locally-optimal models is used, increasing the computational burden.

In Chapelle and Vapnik (1999), an analytical approach is proposed based on rescaling

the inputs $\mathbf{x}_i$ in relation to the *span* of the support vectors. For ε-insensitive Support Vector Regression (ε-SVR) in particular, in Cherkassky and Ma (2004) an analytical approach to selection of the cost parameter C is proposed based on the mean and standard deviation of the target values y_i. Milenova *et al.* (2005) describes the analytical approach of the Oracle Data Miner (ODM) product, which estimates C based on the distribution of the Lagrangian multipliers calculated for a small, random sample of training data. Analytical selection of ε is proposed in Vapnik (1995) based on the noise level of the inputs $\mathbf{x}_i$, and in Cherkassky and Ma (2004) also considering the number of training samples ℓ. In Smola and Schölkopf (2004), a combination of analytical and combinatorial parameter selection is proposed, such that the choice of ε is tuned to a particular noise density but the choice of C is chosen through a numerical approach.

The well-known parameter optimization method of simulated annealing (Kirkpatrick *et al.*, 1983) has recently been proposed as a stochastic method for traversing SVM free parameter space. In Figure 2.6, a comparison between a purely random search and such a guided, stochastic search is presented. The points evaluated by the simulated annealing algorithm concentrate on the area of interest to a much greater extent. Such techniques have been applied to synthetic and noisy image data for optimization of the cost and kernel parameters (Imbault and Lebart, 2004), feature selection for audio classification with a linear SVM (Degroeve *et al.*, 2005) and colon cancer recognition using radial basis function classifiers (RBFC) (Wang *et al.*, 2005).

Regularization for artificial neural networks is well understood. An excellent description of Tikhonov's method for the extrinsic regularization of ill-posed problems (Tikhonov, 1963) as applied to ANN may be found in Haykin (1999), in which the complexity penalty is a function of the norm of the internal synaptic weights matrix. Indeed, the formulation of the SVM includes intrinsic regularization from this same principle, as we have seen in the derivation of the Lagrangian functional from Equation 2.9. While this intrinsic regularization allows a tradeoff between model complexity and classification accuracy to enhance generalizability in the non-separable case (Vapnik, 1995), the overall generalization performance is still highly dependent on appropriate selection of the C and γ free parameters.

Thus, in this thesis, we propose to take this regularization into account extrinsic to the SVM itself, when tuning free parameters to find the most optimum solution. We will refer to this as *extrinsic regularization* in this work.

2.4 Visualizing Generalization Performance

If we examine the topology of a surface representing the generalization performance of an SVM classifier, training the classifier using parameters selected by varying the cost parameter C and the width parameter γ of the RBF kernel over a $\log_e$ range of values, some interesting patterns begin to develop.

In Figures 2.7, 3.1, 3.2, 3.3 and 4.1, light areas correspond to parameter values which yield high accuracy and dark areas to those that yield poor accuracy. In many cases, we see a surface fraught with many sharp local extrema, narrow valleys and sharp cliffs. This effect may be exacerbated by the noisy nature of N-fold cross-validation, resulting from the random partitioning of training data. These effects may be reduced by taking the mean of several evaluations at each point, thereby increasing computational complexity, or by using leave-one-out cross-validation, which is naturally less prone to these random fluctuations due to the completeness of the combinatorial search. While the effects of these sharp extrema may well be magnified by the log operation, we must take them into account since the $\log_e$ space is the surface we wish to traverse.

Although the shape of this generalization error surface is naturally problem dependent, for many data sets the surface follows a quite similar overall shape (see for example Hsu *et al.*, 2003; Staelin, 2003). However, this general shape is far from guaranteed: a surface with two fully linearly-separable Gaussian distributions, for example, may allow a much wider region of parameter values to achieve high accuracy (see Figure 3.2).

The difficulties of traversing such a complex, volatile surface are immediately apparent. Gradient ascent methods (Chapelle *et al.*, 2002; Fan *et al.*, 2005) may become stuck in local extrema, while hill-climbing algorithms such as the geometric approaches in Momma and Bennett (2002) may traverse such space inefficiently as they wind their way through a long, narrow valley (Press *et al.*, 1992). In addition, if we choose an optimum point in parameter space near the edge of a sharp cliff or other local extrema (Imbault and Lebart, 2004), it is quite possible that small variations in the sample data may cause the surface to subtly shift, causing the classifier to "fall" from the cliff to an area of lower accuracy.

When a high-resolution close-up of a small region of the surface is viewed in Figure 2.7 (lower right), smoothed over the mean of several iterations at each point, the chaotic patterns seen at a high level seem to be formed by the convergence of multiple, smaller regions. In this image, over $40\,000$ points were evaluated, with several iterations at each evaluated

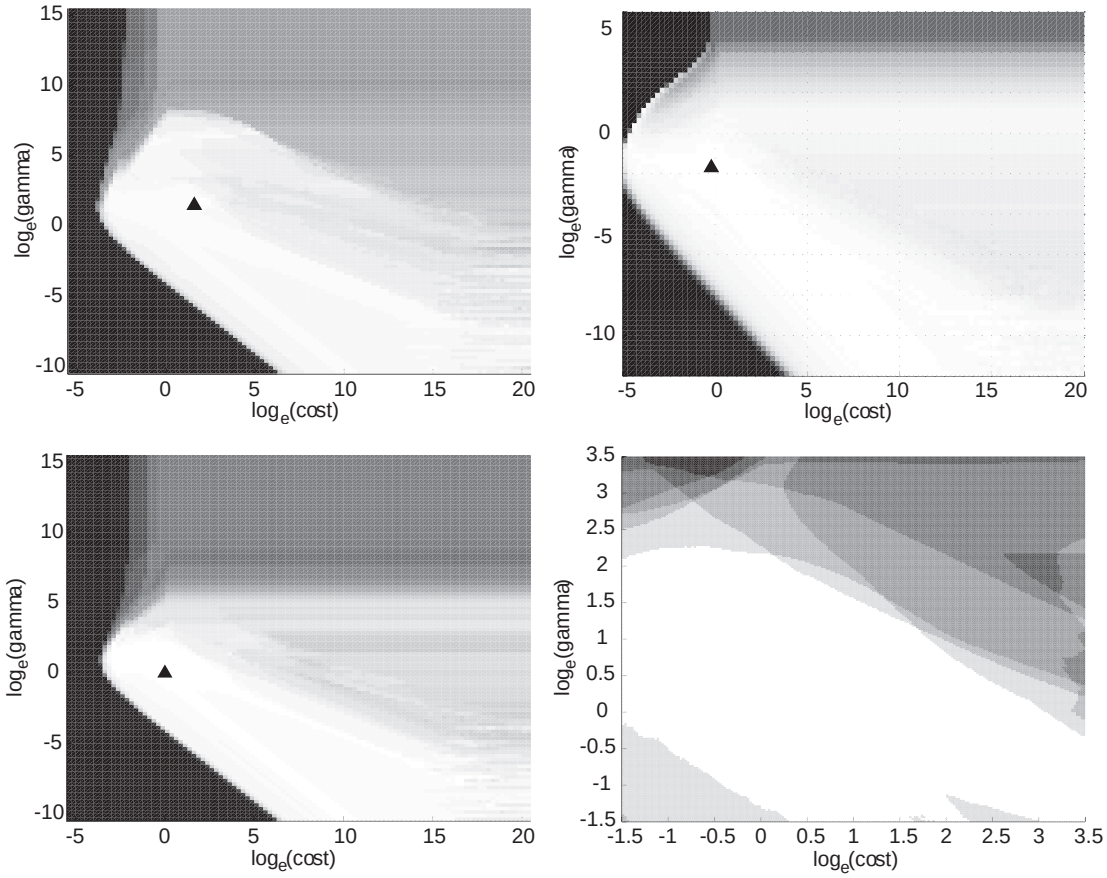


Figure 2.7: Error surfaces resulting from SVM classification by varying the C and γ free parameters over a $\log_e$ range of values. Dark areas correspond to high error, whereas light areas correspond to high accuracy. Triangles (▲) indicate the optimum point found through a grid search of $\log_e$ space, considering only the classification accuracy and disregarding model complexity. The *upper right* image shows the result of a typical grid search on the Wisconsin Breast Cancer Database (Newman *et al.*, 1998). The remaining images show results from a high-resolution grid search on the three-class Protein Sequence Alignment Quality data set examined later in this thesis (Shan *et al.*, 2003): Valid vs. other (*upper left*), Inadequate vs. other (*lower left*) and a close-up of Inadequate vs. other (*lower right*) showing the convergence of multiple smaller regions. The optimum point is not shown on the close-up for clarity. Additional examples of this visualization of the generalization error surface in $\log_e$ space are shown throughout this thesis.

point to smooth the noise from 10-fold cross-validation. Closer investigation may determine that the optimal point may really be formed from the convergence of such regions to a single point in parameter space. One might speculate that a mathematically-rigorous analysis of such regions may one day lead to a widely-applicable, analytical solution to the problem of SVM parameter optimization.

Chapter 3

Proposed Heuristic

Waveform classification is a challenging problem for any classifier, as the number of dimensions may be much larger than the number of available observations ($\ell \ll d$): with such a large number of dimensions, it is statistically more likely that one or more of those dimensions may be fully separable simply by chance, and therefore a classifier may achieve high accuracy by basing predictions on only that dimension during training. This is a significant advantage for SVM (Smola and Schölkopf, 2004), which are much less sensitive to any one input as the margin classification takes place in mapped (feature) space, where each dimension is created from many input dimensions (Boser *et al.*, 1992). An example of this is shown in Section 7.3.2. To further reduce the impact of this effect, proper cross-validation is critical for generalization performance. We will use N-fold cross-validation in the majority of experiments in this thesis, with the important exception of the retinal electrophysiology data set: in this binary classification experiment, due to the low number of observations, leave-one-out cross-validation was used resulting in a smoother error surface.

In this chapter, we present the heuristic employed throughout this thesis for the problems of classification and regression using SVM. We then demonstrate how this heuristic improves generalization performance for separable and non-separable data sets, in comparison to an SVM with default parameters, an SVM with parameters optimized by a grid search and a classical neural network trained using backpropagation. A summary of the notation used in this section is shown in Table 3.1.

3.1 Simulated Annealing

Simulated Annealing is a well-known, stochastic technique for combinatorial optimization based on a thermodynamic analogy of slowly cooling metals, which in nature reach a minimum energy state (Kirkpatrick *et al.*, 1983). Use of a stochastic algorithm such as simulated annealing allows efficient searching through a noisy, multidimensional parameter space, such as simultaneous optimization of the RBF kernel parameter γ and the cost parameter C, without the need for gradient calculations at each point. A comparison of the guided, stochastic approach in simulated annealing to a uniform, random search pattern is

Table 3.1: List of notation used to define the proposed heuristic.

Symbol	Meaning
λ	Extrinsic regularization parameter
$\mathscr{E}(f)$	Cost functional measuring performance of function f
$\mathscr{E}_s(f)$	Cost functional component describing generalization performance
$\mathscr{E}_c(f)$	Cost functional component describing model complexity penalty
Γ	Non-linearity introduced to complexity penalty
$\mathbf{P}$	A point in $\log_e$ parameter space
$\mathbf{P}_\varnothing$	The origin in $\log_e$ parameter space: $\mathbf{P}_\varnothing = \{\log_e(0), \log_e(0)\}$
$\mathbf{P}_{opt}$	The optimum point in $\log_e$ parameter space
$\mathbf{P}_{sugg}$	The suggested point in $\log_e$ parameter space
α	Proportion of origin bias in heuristic
β	Maximum threshold of detrimental jumps in heuristic
δ	Proportion of temperature reduction in heuristic
m	In heuristic, the temperature is reduced every m steps
T_0	Initial temperature in heuristic
T_C	Ending temperature in heuristic
r_0	Maximum radius threshold from $\mathbf{P}_{opt}$
ξ	Maximum cost functional threshold from $\mathbf{P}_{opt}$
p_{acc}	Probability of accepting a detrimental jump in heuristic
p_{res}	Probability of reset to best point found so far in heuristic

illustrated in Figure 2.6.

In this thesis, we use this approach to traverse the generalization error surface examined in Section 2.4. Here we adapt the continuous, N-dimensional implementation of this algorithm shown in Press *et al.* (1992), but employ a simple move generator, with a small bias towards the origin $\mathbf{P}_\varnothing = \{\log_e(0), \log_e(0)\}$. This bias will favour accurate models with low complexity: we have found that in practice, the extremes of these free parameters will tend to result in a model which largely overfits with very high complexity, or underfits with very low complexity, and which prevents the SVM training algorithm from quickly converging to a solution. We also implement occasional restarts to the most optimum points found thus far, with low probability, an alternative suggested in Press *et al.* (1992).

With this heuristic, the ending point of the simulated annealing path is not guaranteed to be the optimum point, as small ascents in the cost functional are allowed — with a lower probability of acceptance — in order to climb from local extrema. Rather than the path's end point, therefore, the overall best point found throughout the search is used as

the optimum point $\mathbf{P}_{opt}$. Details of the specific implementation of the simulated annealing mechanism used in this thesis are summarized in Section 3.5.

3.2 Extrinsic Regularization

At each evaluated point in this stochastic path through parameter space, a score is computed based not only on the cross-validated accuracy or mean square error obtained from testing a model trained with these parameters, but also on a measure of the model complexity, in order to improve generalization performance and lower sensitivity to volatile input fluctuations even at some expense of the absolute accuracy of the model on the training set.

Adopting the notation of Haykin (1999), which applied a similar Tikhonov regularization (Tikhonov, 1963) to ANN, we wish minimize the regularization functional

$$\mathscr{E}(f) = \mathscr{E}_s(f) + \lambda\, \mathscr{E}_c(f) \tag{3.1}$$

The regularization parameter λ allows control over the tradeoff between classification accuracy and model generalizability. Using this functional, the cost $\mathscr{E}(f)$ at the point $\mathbf{P}_i = \{C_i, \gamma_i\}$ in parameter space is determined not only by the loss functional $\mathscr{E}_s(f)$ of an SVM trained using the parameters defined by that point, for example from the empirical, binary classification risk in Equation 2.5:

$$\mathscr{E}_s(f) = \frac{1}{2\ell} \sum_{i=1}^{\ell} |y_i - f(\mathbf{x}_i)| \tag{3.2}$$

for which $\mathscr{E}_s(f) \in [0, 1]$, or by evaluating the overall risk across a separate validation data partition, but also by a complexity penalty $\mathscr{E}_c(f) \in (0, 1]$ defined by the number of support vectors n_{sv} in the model representation of $f(\mathbf{x}_i)$, expressed as a ratio to the total number of observations ℓ, as

$$\mathscr{E}_c(f) = \left(\frac{n_{sv}}{\ell}\right)^{\Gamma} \tag{3.3}$$

where the free parameter Γ introduces a non-linearity to shift the sharpest effect of the complexity term to either the upper range of values, which our experiments found was useful for classification problems (see Chapter 4), or the lower range of values, which our experiments found was useful for regression problems (see Chapter 6).

The number of support vectors was obtained by training the SVM model using parameters determined by the test point $\mathbf{P}_t$ for all observations in the training set, and counting

the number of support vectors in the resulting model. For the classification problems in this thesis, we give equal weight to both accuracy and complexity by setting $\lambda = 1$ for each of the following experiments. We also set the free parameter $\Gamma = \frac{1}{2}$ to sharply penalize solutions which obtain high accuracy through high complexity. For the univariate and multivariate regression problems, this functional is modified somewhat to include the mean squared error rather than classification error, as will be discussed in Chapters 5 and 6.

In Figure 3.1 (*lower right*), this complexity penalty is added to the generalization error surface from a grid search through parameter space for the *Iris versicolour* class in the Iris Plant Database, examined in Chapter 4: the resulting surface is smoother than that of Figure 3.1 (*lower left*), which considers only generalization error, indicating that the number of support vectors in the complexity penalty is less volatile than the cross-validation accuracy.

One might also consider use of the VC dimension (Vapnik, 1995) as an alternative model complexity measure, which can be estimated in practice from Equation 2.27.

3.3 Intensity-Weighted Centre of Mass

Once the cooling schedule has elapsed, we select the absolute best point found $\mathbf{P}_{opt}$ as that which achieves the minimum possible value of Equation 3.1. In the case that several points achieve the same score, which is possible in classification with a finite training set, we select the point closest to the origin $\mathbf{P}_{\varnothing}$ as above.

We examine the points surrounding $\mathbf{P}_{opt}$ to select those within a small $\log_e$ radius r_0 and with a cost functional $\mathscr{E} \leq (1+\xi)\mathscr{E}_{opt}$ where $\xi > 0$ is small, then, borrowing a standard method from the field of image processing (see for example Stelzer, 1998), we calculate an intensity-weighted centre of mass of these points where the intensity is $(1-\mathscr{E})$. This has the effect of reducing the volatility of the resulting end-point arising from the random nature of the generalization error surface, as we show in Section 4.1.4. The resulting point in parameter space $\mathbf{P}_{sugg}$ defines the suggested parameters to be used for a particular problem.

Figure 3.1 illustrates the importance of this centre-of-mass operation. In the *upper left*, we show a surface for the highly separable *Iris setosa* class, for which many parameter values will achieve 100% accuracy: in this case, the optimum point $\mathbf{P}_{opt}$ is selected as that closest to the origin $\mathbf{P}_{\varnothing}$. The centre-of-mass operation selects those points within a radius r_0 from this optimum point: in this case, the resulting point $\mathbf{P}_{sugg}$ shifts somewhat to the

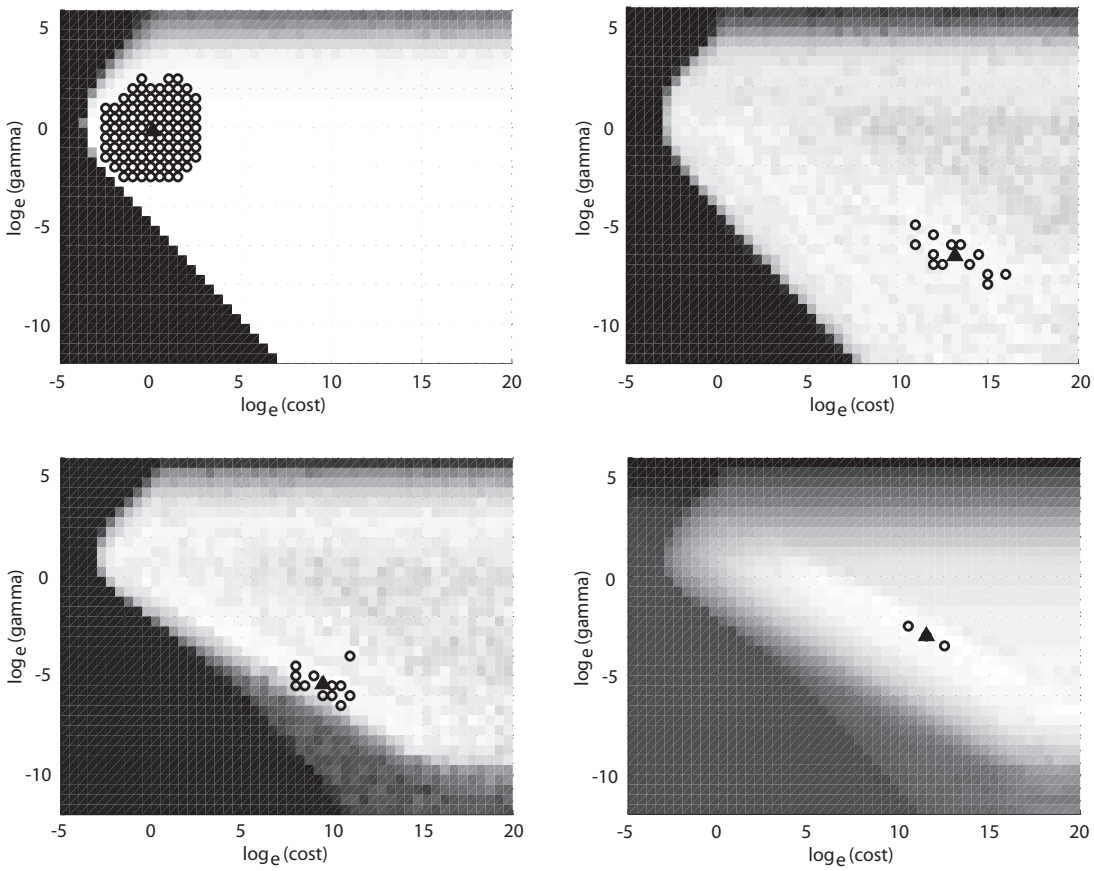


Figure 3.1: Visualizing the generalization performance of a grid search through $\log_e$ space for Iris Plant Database (Newman *et al.*, 1998), showing use of the centre of mass and the difference in the resulting plot when a complexity penalty is employed. Circles (○) indicate the group of best points found. The triangle (▲) shows the optimum point calculated from this group of points, using an intensity-weighted centre-of-mass operation. In the *upper left*, we show a surface for the highly separable *Iris setosa* class. In the *upper right*, we show a surface for the non-separable *Iris virginica* class. On the *lower left*, we show a surface for the non-separable *Iris versicolour* class. In the *lower right*, a complexity penalty is added to the generalization error surface from the *Iris versicolour* class.

right, away from the sudden drop in accuracy on the leftmost portion of this illustration.

In the *upper right*, we show a surface for the non-separable *Iris virginica* class. In this case, the centre-of-mass operation places the suggested point at the centre of a shallow oval: 100% accuracy cannot be achieved, so the centre-of-mass operation selects the overall best point $\mathbf{P}_{opt}$ and then a group of points near this point that have an accuracy within ξ of the accuracy achieved at $\mathbf{P}_{opt}$. On the *lower left*, we show a surface for the non-separable *Iris versicolour* class. In this case, we see an example of how the centre-of-mass operation moves the optimum point further away from a region of lower accuracy, improving generalization performance for finite training sets since the addition of further training examples, without adjusting the training parameters, could potentially shift the generalization error surface such that the selected points are no longer optimum.

Although this can potentially reduce accuracy somewhat in comparison to the optimum point $\mathbf{P}_{opt}$, the resulting point in parameter space is likely to be further from any steep cliffs in the evaluated generalization surface. Since we have a finite set of observations, the decision boundary of the classifier will likely change as additional training samples are evaluated. For example, if we have a large number of observations, it may be prudent to use a small subset of the training data to find optimum parameters, but employ the full set of training data to train the final classifier: this is the approach we take in Chapter 5. This volatility may cause the generalization surface to shift slightly as more samples are added to the training, such that a point in parameter space selected without those samples, close to an edge such as illustrated here, may "fall" from the edge to a region of lower accuracy. This is largely prevented by using a centre-of-mass operation, resulting in a "safer" model.

3.4 Examining Known Distributions

To illustrate how this heuristic is affected by separability in binary classification, we first examine two binary classification problems with known distributions.

In Figure 3.2, we see classification of a balanced, toy data set consisting of two Gaussian distributions centred at (5,5) and (-5,-5), with 50 sample points per class. The upper figure shows the generalization error surface formed by varying the cost parameter C and the width parameter of the RBF kernel γ over a $\log_e$ range of values, as in Figure 2.7. The triangle (▲) indicates the suggested point in parameter space resulting from the grid search, but with an intensity-weighted centre of mass operation as discussed in Section 3.3. Note

that high cross-validation accuracy is achieved across much of the parameter space, as we would expect for such highly-separable data.

In the lower figures, we compare the decision boundary resulting from several different classifiers. The data points for positive ($\circ$) and negative ($+$) classes are shown for comparison, and the solid line indicates the decision boundary for each classifier. In Figure 3.2(*a*), an SVM is trained with default parameters which, in the LIBSVM implementation employed here, are an RBF kernel width of $\gamma = 0.5$ and a cost parameter of $C = 1$. The classifier model contains 23 support vectors, selected from the 100 training samples. In Figure 3.2(*b*), an SVM is trained using parameters obtained from the grid search using 10-fold cross-validation. The classifier model contains 19 support vectors. In Figure 3.2(*c*), an SVM is trained using the fast-cooling heuristic proposed in this thesis. The classifier model contains only two support vectors, one from each class. Note the greatly increased generalizability, even with a fully separable data set. In Figure 3.2(*d*), the results from a multilayer perceptron (MLP) with 5 hidden nodes is shown for comparison: here we use a neural network implementation from Netlab Nabney (2002), trained using backpropagation with quasi-Newtonian optimization. The class separation is nearly identical to the SVM classifier regularized by the proposed heuristic. All four classifiers achieve 100% cross-validation accuracy on the training set, but (c) and (d) clearly will achieve better accuracy on future predictions as they both offer a far more general solution.

In Figure 3.3, we see a similar comparison, but now with 25 times the variance in the sample data to allow some overlap, creating a non-separable data set. We see a much more volatile error surface, with narrow valleys of high accuracy. Note that all four cases do not generalize well: the SVM trained with default parameters has severe overfitting, with 89 support vectors in the underlying model, whereas the decision boundaries created by the other three classifiers appear to approach a more logical linear solution. The models trained using the heuristic and using the grid search have only 12 support vectors, and the resulting models are quite similar. However, in the illustration of the generalization error surface, we find that the cost parameter found by the grid search is near the extreme of the range of values considered, whereas the cost parameter chosen by the heuristic is closer to the origin. The decision boundary chosen by the heuristic seems quite different than that of the MLP, however the generalization performance is about the same on the training set. Notice, however, that there is a limited region of inaccuracy for both SVM models, whereas the region of inaccuracy for the MLP appears to stretch to infinity: again, an SVM

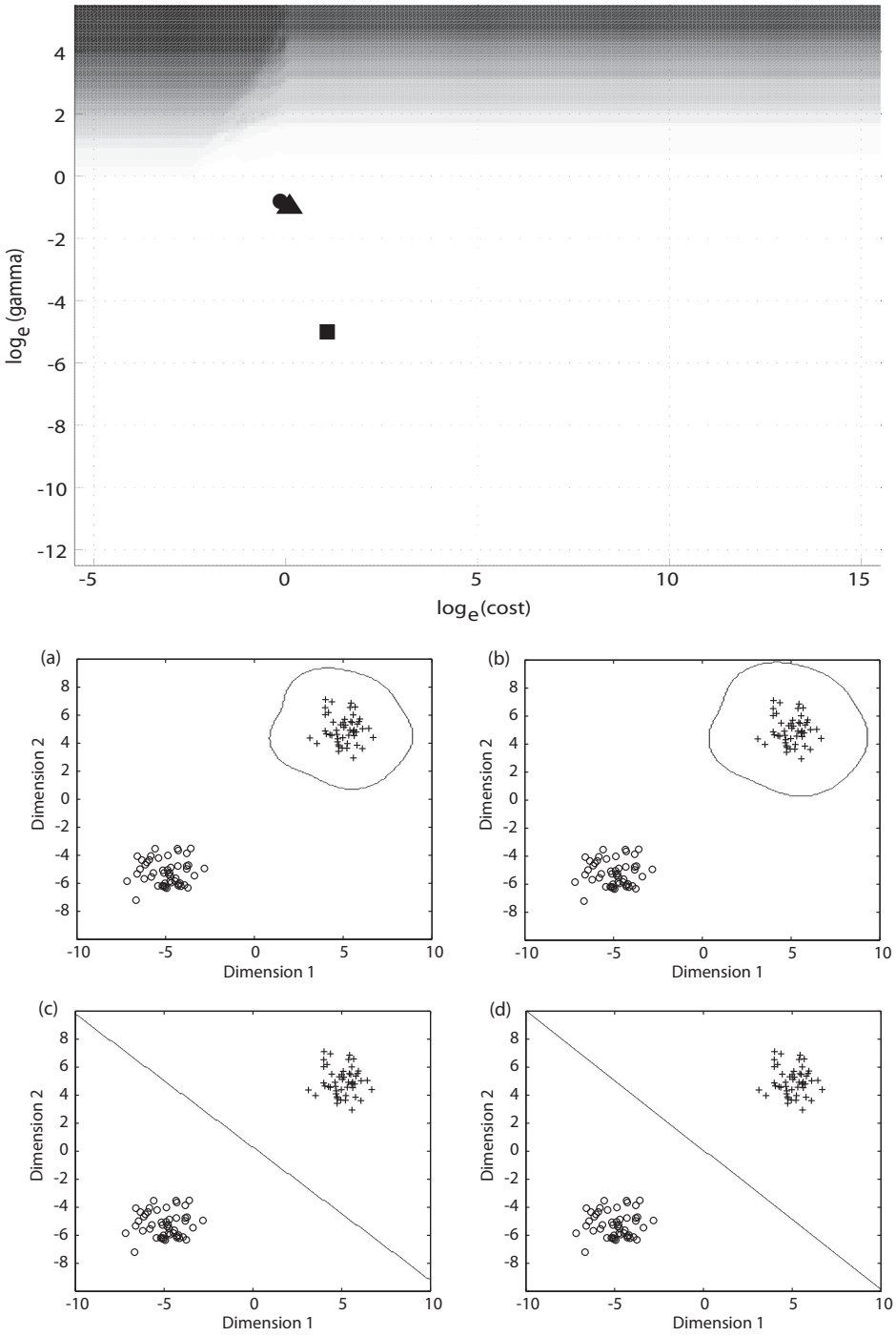


Figure 3.2: Applying SVM classification to a linearly-separable toy data set. The *upper* image shows the generalization error surface resulting from a grid search using cross-validation. The *lower* images show decision boundaries resulting from classification with (*a*) SVM with default parameters (•), (*b*) SVM from above grid search (▲), (*c*) SVM using the heuristic proposed in this thesis including extrinsic regularization (■), and (*d*) an MLP with five hidden nodes. Training data for positive (○) and negative (+) classes are shown for comparison.

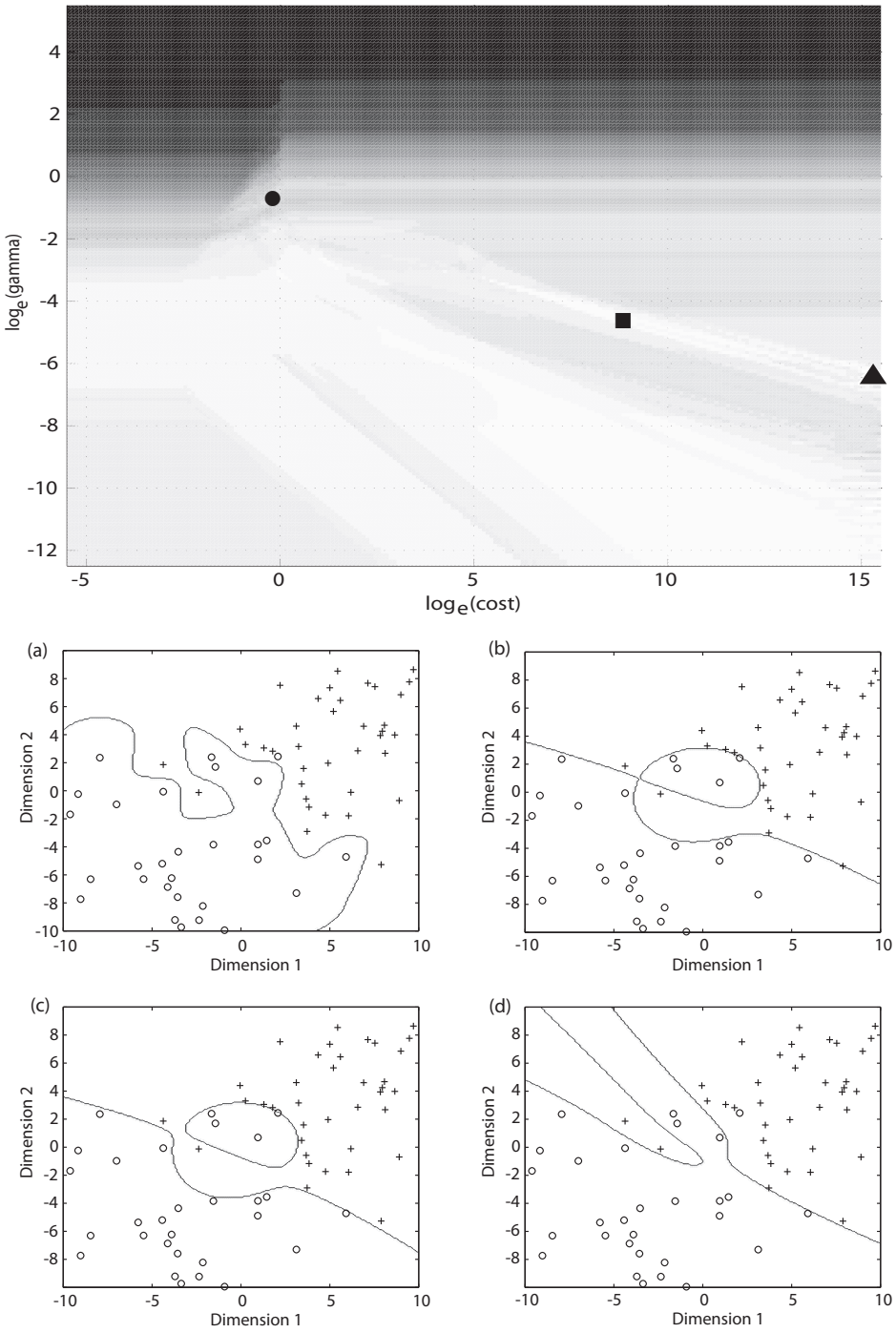


Figure 3.3: Applying SVM classification to a non-separable toy data set. The *upper* image shows the generalization error surface resulting from a grid search using cross-validation. The *lower* images show decision boundaries resulting from classification with (*a*) SVM with default parameters (●), (*b*) SVM from above grid search (▲), (*c*) SVM using the heuristic proposed in this thesis including extrinsic regularization (■), and (*d*) an MLP with five hidden nodes. Training data for positive (○) and negative (+) classes are shown for comparison.

with optimized parameters offers a more general solution.

We first conclude from these experiments that proper optimization of the free parameters in SVM is critical to achieving high generalization performance: in both cases, the default parameters (with no optimization) do not perform well. We also find in both experiments that parameters chosen from a reasonable grid search will achieve as good or better accuracy on the finite training set, however the generalization performance to classify future observations is clearly made better by including a model complexity term, such as is done in the heuristic proposed in this thesis. By including this model complexity term, we achieve high generalization performance on the training set, but also high accuracy on future observations, comparable to a well-optimized neural network.

3.5 Implementation Details

The details of the simulated annealing heuristic used in this thesis are as follows. These steps are illustrated in Figures 3.4 and 3.5.

1. Initialize a high temperature $T \leftarrow T_0$, and set the starting point $\mathbf{P}_i \leftarrow \mathbf{P}_0$ within the $\log_e$ parameter space.

2. Determine a new test point $\mathbf{P}_t$, taken in a random direction from $\mathbf{P}_i$ (with uniform distribution), and a random scalar distance (with normal distribution) multiplied by the ratio T/T_0 and the width (or height) of the parameter space.

3. Add a small origin bias $\mathbf{P}_t \leftarrow \alpha(\mathbf{P}_i - \mathbf{P}_\varnothing)$ where $\mathbf{P}_\varnothing$ defines the $\log_e$ origin and α is a small scalar. Check that the current point $\mathbf{P}_t$ lies within the boundary conditions of the parameter space: if not, select a new point at random anywhere within the parameter space.

4. Determine a scalar cost functional $\mathscr{E}_t$ for the current position $\mathbf{P}_t$, including the classification error and model complexity of an SVM model trained using the parameters at this point. If the cost functional $\mathscr{E}_t$ for $\mathbf{P}_t$ is less than that of $\mathscr{E}_i$ of $\mathbf{P}_i$ (or if this is the first evaluated point), accept this point as the new position in the parameter space. If not, but the resulting ascent in cost is small, perhaps accept this point with a small probability p_{acc}. Otherwise, reject it.

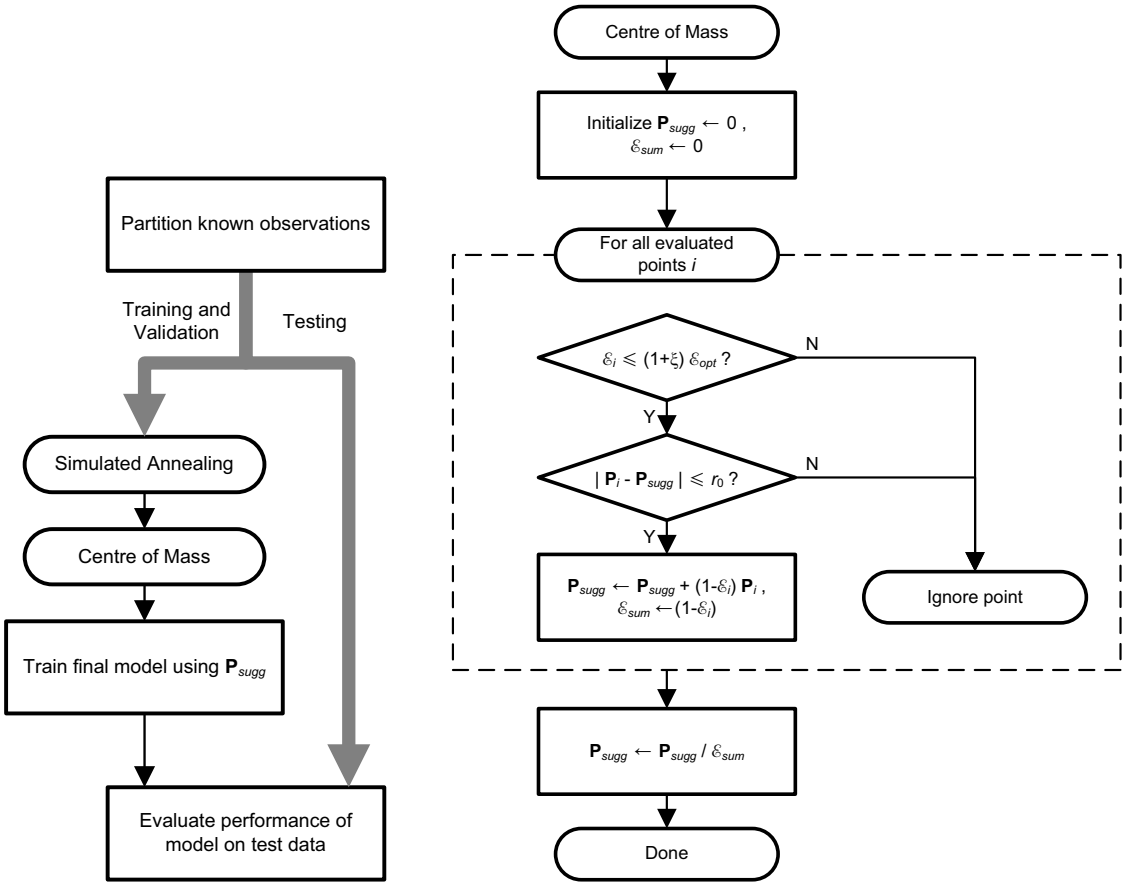


Figure 3.4: Illustration of proposed heuristic, detailed in Section 3.5. (a) Overall process of training an SVM on a set of known observations. (b) Intensity-weighted centre-of-mass operation. Here we assume a single *training and validation* partition, as is done in most problems in this thesis due to the low number of observations available: we therefore use N-fold cross-validation. However, in Chapter 5, we have sufficient available observations to create a separate *validation* partition: in this case, as each point in parameter space is evaluated, we will use the regression performance on the validation set rather than cross-validation.

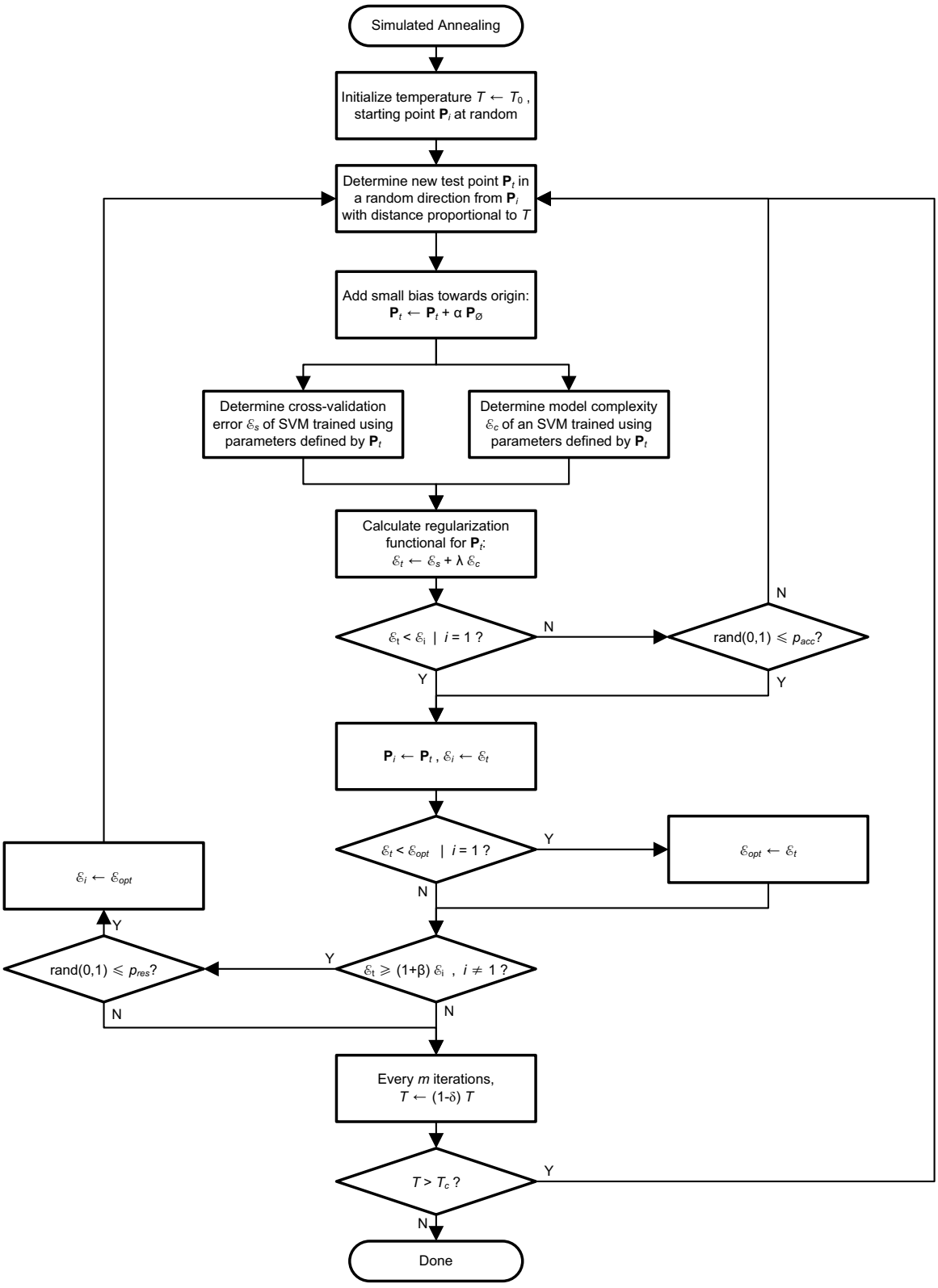


Figure 3.5: Illustration of simulated annealing in proposed heuristic, detailed in Section 3.5. Some details have been removed for clarity, such as bounds checking.

5. If the point was accepted, set $\mathbf{P}_i \leftarrow \mathbf{P}_t$ and $\mathscr{E}_i \leftarrow \mathscr{E}_t$, then compare the cost functional $\mathscr{E}_i$ with $\mathscr{E}_{opt}$ of the most optimum points obtained thus far: if the cost is lower, replace the existing list with the current point; if the cost is approximately equivalent within a small margin of error, add the current point to the existing list.

6. If the cost $\mathscr{E}_i > (1+\beta)\mathscr{E}_{opt}$, where β is a small scalar, then with a very small reset probability $p_{res} \ll p_{acc}$, jump to a randomly selected point from the list of optimum points.

7. Drop the temperature $T \leftarrow T(1-\delta)$ every m iterations. If the temperature is still higher than the termination criteria T_C, continue through further iterations from step 2. Otherwise, determine a single optimum point $\mathbf{P}_{opt}$ as the point in the set of optimum points that lies closest to the origin $\mathbf{P}_{\varnothing}$.

8. Gather a set of points from the list of all evaluated points which have a cost within $\mathscr{E} \leq (1+\xi)\mathscr{E}_{opt}$ of the best cost $\mathscr{E}_{opt}$, where ξ is a small scalar, and which lie within a small radius r_0 from the optimum point $\mathbf{P}_{opt}$.

9. Determine the suggested point $\mathbf{P}_{sugg}$ as the intensity-weighted centre of mass of this set of points, using $(1-\mathscr{E}_i)$ as the intensity for each point i, and retrain the model with the parameters determined by this suggested point using all training points.

A MATLAB implementation of this heuristic using either LIBSVM (Chang and Lin, 2001) or SVM^{light} (Joachims, 1999) may be downloaded from `http://www.cs.dal.ca/~tt`.

Chapter 4

Classification Results

In this chapter, we apply the heuristic proposed in Chapter 3 to two well-known classification problems and to two real-world classification problems arising from research performed at Dalhousie University by other groups.

In our experiments, we have obtained reasonable results with the proposed heuristic by setting the arbitrary cooling schedule to start at $T_0 = 100$ and cool to $T_C = 0.1$, reducing the temperature every $m = 10$ iterations by $\delta = 0.01$ (for *slow* cooling) or by $\delta = 0.1$ (for *fast* cooling). The fast cooling schedule therefore results in 660 evaluations, whereas the slow cooling schedule results in 6880 evaluations. We chose these schedules such that the number of evaluations would approximately match those of coarse- and fine-grained grid searches over the same parameter space, and did not tune these schedules for any particular experiment as we desire an automatic method for parameter selection. The ϵ termination criteria for the quadratic optimization (not to be confused with the ε-insensitive tube width for regression problems) was found to make little difference in practice, so this was left at the LIBSVM default $\epsilon = 0.001$ (Chang and Lin, 2001).

In each of the following experiments, the origin bias was set to $\alpha = 0.1$ (one tenth the magnitude of the move selected by random walk). The probability of accepting a test point with a detrimental cost was set to $p_{acc} = 0.1$, so long as the cost is within $\beta = 0.1$ of the current point. At any point in the search, the probability of abandoning the current path in favour of a random selection amongst the best points found so far was $p_{res} = 0.01$. The intensity-weighted centre of mass calculation after the completion of the cooling schedule included those points within a $\log_e$ radius of $r_0 = 1$ from the best point found $\mathbf{P}_{opt}$, and which have a cost functional $\mathscr{E} \leq (1 + \xi)\mathscr{E}_{opt}$ where $\xi = 0.02$. No fine-tuning of these parameters was performed for any particular experiment, since we wish to evaluate generalization performance when the heuristic is employed blindly.

4.1 Classic Classification Problems

We first apply this heuristic to two standard classification problems, in order to compare the results of the stochastic heuristic above including the model complexity measure, with

Table 4.1: Sample classification results using the slow- and fast-cooling heuristic on the Wisconsin Breast Cancer Database (Mangasarian and Wolberg, 1990) and Iris Plant Database (Fisher, 1936), compared to a reasonably–sized grid search: *Best* indicates the overall best point found during the grid search, and *Suggested* adds an intensity-weighted centre of mass, as in the heuristic. For each test, the 10-fold cross-validation accuracy, the number of support vectors, and the number of evaluated points in $\log_e$ parameter space are shown.

Database	Search Method	Accuracy	n_{sv}	Evals.
WBCD	Fast-Cooling Heuristic	96.5	36	660
	Slow-Cooling Heuristic	96.0	34	6880
	Grid Search: Best	97.4	129	7373
	Grid Search: Suggested	97.1	77	7373
Iris database:	Fast-Cooling Heuristic	100	3	660
Iris setosa	Slow-Cooling Heuristic	100	3	6880
(linear)	Grid Search: Best	100	12	7373
	Grid Search: Suggested	100	12	7373
Iris database:	Fast-Cooling Heuristic	95.3	13	660
Iris versicolour	Slow-Cooling Heuristic	94.7	9	6880
(non-linear)	Grid Search: Best	98.0	28	7373
	Grid Search: Suggested	96.7	35	7373
Iris database:	Fast-Cooling Heuristic	96.7	8	660
Iris virginica	Slow-Cooling Heuristic	98.0	6	6880
(non-linear)	Grid Search: Best	98.0	33	7373
	Grid Search: Suggested	97.3	35	7373

a reasonably–sized grid search based only on cross-validation classification accuracy.

4.1.1 Wisconsin Breast Cancer Database

The Wisconsin Breast Cancer Database (WBCD) is a binary classification problem with non-separable data. It was donated to the UCI Machine Learning repository (Newman *et al.*, 1998) in 1992 by Dr. William H. Wolberg, University of Wisconsin Hospitals (Mangasarian and Wolberg, 1990) based on anonymous clinical data. There are two classes, *malignant* or *benign*. The database contains 699 instances and nine discretized, numeric attributes on a scale from one to ten describing aspects of each tumor, summarized in Table 4.2. 16 of the instances contain missing data in one or more columns; in our tests these instances were not included, leaving a total of 683 observations.

Table 4.2: Numeric inputs in the Wisconsin Breast Cancer Database and Iris Plant Database. *Source:* Newman *et al.* (1998).

Database	Attribute	Range
WBCD	Clump Thickness	$\{ 1, \ldots, 10 \}$
	Uniformity of Cell Size	$\{ 1, \ldots, 10 \}$
	Uniformity of Cell Shape	$\{ 1, \ldots, 10 \}$
	Marginal Adhesion	$\{ 1, \ldots, 10 \}$
	Single Epithelial Cell Size	$\{ 1, \ldots, 10 \}$
	Bare Nuclei	$\{ 1, \ldots, 10 \}$
	Bland Chromatin	$\{ 1, \ldots, 10 \}$
	Normal Nucleoli	$\{ 1, \ldots, 10 \}$
	Mitoses	$\{ 1, \ldots, 10 \}$
Iris	Sepal Length (cm)	$\mathbb{R}$
	Sepal Width (cm)	$\mathbb{R}$
	Petal Length (cm)	$\mathbb{R}$
	Petal Width (cm)	$\mathbb{R}$

4.1.2 Iris Plant Database

The Iris Plant Database was donated to the UCI Machine Learning repository in 1988 by Michael Marshall (Newman *et al.*, 1998). It was originally created by Ronald A. Fisher in 1936 (Fisher, 1936), and includes 50 instances for each of three classes — *Iris setosa*, *Iris versicolour* and *Iris virginica* — for a total of 150 observations. There are four continuous-valued numeric attributes, summarized in Table 4.2. There are therefore three binary classification problems, as each class is to be separated from the remaining two. The *Iris setosa* class is known to be linearly separable from the remaining two classes (Newman *et al.*, 1998).

4.1.3 Generalization Performance

In both of these classic classification problems, we left the classes unbalanced, with the natural class distribution, but centered and scaled all numeric attributes based on the mean and maximum magnitude of each attribute to approximate independently and identically distributed (i.i.d.) data. We also adjusted the class labels, such that

$$\mathbf{x}_i \in [-1, +1], \quad y_i \in \{-1, +1\}, \quad i = 1, \ldots, \ell \tag{4.1}$$

Sample results from these tests are summarized in Table 4.1 and compared to a reasonably-sized grid search, as shown in Figures 2.7 (*upper right*), 3.1 and 4.1. We find that the heuristic results in a model with comparable accuracy, often obtained with fewer calculations. Due to our inclusion of the number of support vectors n_{sv} when evaluating the cost functional at each point, the resulting models all have lower complexity than that obtained through a grid search using cross-validation accuracy as the only measure. For example, the slow-cooling heuristic for the *Iris virginica* classifier achieved 98.0% 10-fold cross-validation accuracy with six support vectors, whereas the best point from a grid search yields the same accuracy with 33 support vectors.

Some sacrifice of accuracy may be necessary as a tradeoff to favour low complexity: for example, the WBCD classifier with the slow-cooling heuristic obtained 96.0% 10-fold cross-validation accuracy, whereas the grid search obtained 97.4%. On closer examination, however, we see that this slightly higher accuracy was obtained at the expense of high complexity: the grid search results required 129 support vectors, whereas the slow-cooling heuristic required only 34, representing a significant reduction in model complexity.

4.1.4 Consistency of Results

Since the heuristic takes a random path through parameter space, we may wish to determine how consistent the results are when run several times on the same data set. For this purpose, we use the *Iris setosa* and *Iris virginica* classes from the Iris data set, which are linearly separable and non-separable respectively. The results from these tests are shown in Figure 4.1, and summarized in Tables 4.3 and 4.4.

The grid search evaluated 7373 points in parameter space, whereas the fast-cooling heuristic evaluated 660 and the slow-cooling heuristic evaluated 6880. The grid search considered only cross-validation accuracy, whereas the heuristic considers both cross-validation accuracy and model complexity. Both the slow- and fast-cooling heuristics obtain nearly the same accuracy as the grid search, but with far fewer support vectors, indicating that the resulting model is much less complex. The results are reasonably consistent with the slower cooling rate, but have higher variability with the faster cooling rate. In the highly separable case (*left*), both the fast- and slow-cooling heuristics find the same optimum points, to the lower right of the grid search which does not consider model complexity. However, in the non-separable case, the heuristics have less consistency, as can be seen by the higher spread

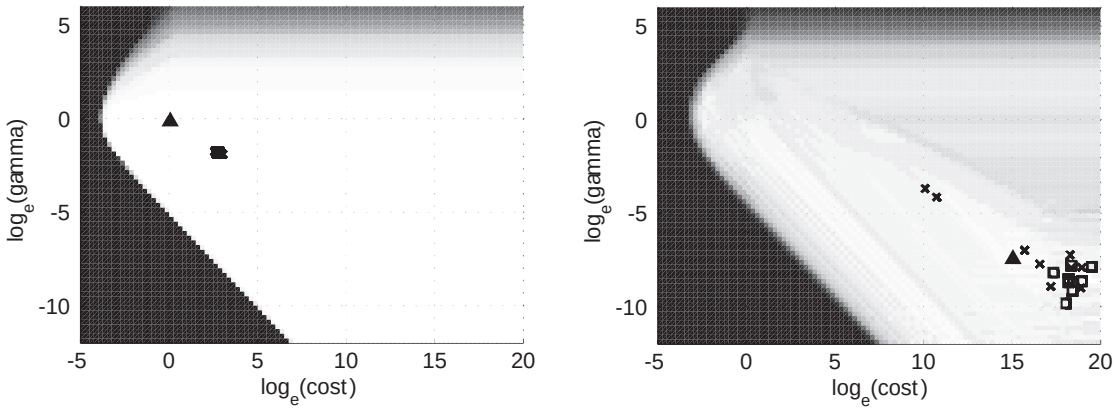


Figure 4.1: Consistency of results from ten sample runs for the Iris database, comparing the results from the slow- and fast-cooling heuristic with a grid search: the linearly-separable *Iris setosa* class (*left*) and the non-separable *Iris virginica* class (*right*). Solid triangles (▲) indicate the suggested optimum point resulting from the grid search including an intensity-weighted centre of mass. Squares (□) indicate suggested positions from the slow-cooling heuristic, while crosses (×) indicate suggested positions from the fast-cooling heuristic.

in the distribution of suggested points. A slower cooling schedule might correct for this, as the points found by the slow-cooling schedule are much more closer together than those of the fast-cooling schedule.

Although the linear separability of the *Iris setosa* class allows a wide range of values that will achieve 100% accuracy with a very low number of support vectors, the suggested parameters for both the fast- and slow-cooling heuristics overlap. For the non-separable problem, however, the variability is relatively high for the fast-cooling heuristic. The range of suggested parameter values for the slow-cooling heuristic is much more narrow, indicating that for non-separable data, a slower cooling schedule should be used.

4.2 Protein Sequence Alignment Quality Data Set

The Protein Sequence Alignment Quality data set (Shan *et al.*, 2003) includes three measures (gap ratio g, normalized site log likelihood ratio h and consistency index CI) used to determine the quality of alignment of $17\,821$ gene and protein sequences in preparation for phylogenetic analysis ($12\,625$ of these were available to us for this experiment). A fourth measure, the site rate, was available in our test set but was not used in our experiments, so that we could compare our results with Shan *et al.* (2003); it seems likely that including the site rate will improve classification performance. The alignment quality of each sequence

Table 4.3: Consistency of results for Iris database, *Iris setosa* class (linearly separable). The results shown for the fast- and slow-cooling heuristics are the mean over ten runs. Parentheses denote standard deviation.

Search Method	$\log_e(\gamma)$	$\log_e(C)$	Accuracy	n_{sv}
Fast-Cooling Heuristic	-1.82	2.92	100	3
Fast-Cooling Heuristic	-1.76	2.58	100	3
Fast-Cooling Heuristic	-1.79	2.72	100	3
Fast-Cooling Heuristic	-1.88	2.87	100	3
Fast-Cooling Heuristic	-1.80	2.72	100	3
Fast-Cooling Heuristic	-1.91	3.04	100	3
Fast-Cooling Heuristic	-1.79	2.69	100	3
Fast-Cooling Heuristic	-1.82	2.75	100	3
Fast-Cooling Heuristic	-1.87	2.93	100	3
Fast-Cooling Heuristic	-1.84	2.78	100	3
Fast-Cooling Heuristic	-1.83	2.80	100	3
(Standard Deviation)	(0.05)	(0.14)	(0)	(0)
Slow-Cooling Heuristic	-1.82	2.75	100	3
Slow-Cooling Heuristic	-1.82	2.76	100	3
Slow-Cooling Heuristic	-1.80	2.77	100	3
Slow-Cooling Heuristic	-1.80	2.75	100	3
Slow-Cooling Heuristic	-1.83	2.79	100	3
Slow-Cooling Heuristic	-1.84	2.79	100	3
Slow-Cooling Heuristic	-1.81	2.75	100	3
Slow-Cooling Heuristic	-1.78	2.71	100	3
Slow-Cooling Heuristic	-1.80	2.78	100	3
Slow-Cooling Heuristic	-1.81	2.78	100	3
Slow-Cooling Heuristic	-1.81	2.76	100	3
(Standard Deviation)	(0.02)	(0.02)	(0)	(0)
Grid Search: Best	0.00	0.00	100	12
Grid Search: Suggested	-0.15	0.06	100	12

Table 4.4: Consistency of results for Iris database, *Iris virginica* class (non-separable). The results shown for the fast- and slow-cooling heuristics are the mean over ten runs. Parentheses denote standard deviation.

Search Method	$\log_e(\gamma)$	$\log_e(C)$	Accuracy	n_{sv}
Fast-Cooling Heuristic	-7.0	15.7	96.7	8
Fast-Cooling Heuristic	-8.9	17.2	94.0	9
Fast-Cooling Heuristic	-4.1	10.7	96.0	8
Fast-Cooling Heuristic	-7.2	18.3	96.0	8
Fast-Cooling Heuristic	-7.9	19.0	96.0	9
Fast-Cooling Heuristic	-9.0	18.9	95.3	8
Fast-Cooling Heuristic	-3.7	10.1	96.7	9
Fast-Cooling Heuristic	-7.7	16.6	96.7	8
Fast-Cooling Heuristic	-7.9	18.4	94.7	8
Fast-Cooling Heuristic	-4.2	10.8	96.0	8
Fast-Cooling Heuristic	-6.8	15.6	95.8	8.3
(Standard Deviation)	(2.02)	(3.62)	(0.89)	(0.5)
Slow-Cooling Heuristic	-8.7	18.2	95.3	8
Slow-Cooling Heuristic	-7.9	19.5	98.0	6
Slow-Cooling Heuristic	-8.2	17.4	98.0	8
Slow-Cooling Heuristic	-7.7	18.4	96.0	7
Slow-Cooling Heuristic	-8.6	19.0	94.7	7
Slow-Cooling Heuristic	-7.8	18.4	96.0	7
Slow-Cooling Heuristic	-8.5	18.2	96.0	10
Slow-Cooling Heuristic	-8.7	18.4	96.7	8
Slow-Cooling Heuristic	-9.2	18.4	96.7	7
Slow-Cooling Heuristic	-9.8	18.0	96.7	9
Slow-Cooling Heuristic	-8.5	18.4	96.4	7.7
(Standard Deviation)	(0.66)	(0.56)	(1.05)	(1.2)
Grid Search: Best	-1.0	0.8	98.0	33
Grid Search: Suggested	-1.9	1.3	97.3	35

was manually categorized by experts into three classes: *Valid*, *Inadequate* and *Ambiguous*. The authors of Shan *et al.* (2003) compared the classification performance of a linear SVM (SVM-Lin), to a C4.5 decision tree algorithm (C4.5) and a Naïve Bayesian classifier (NB). In this work, the performance of the linear SVM classifier was found to be inadequate. However, they mentioned that their SVM implementation did not attempt to identify optimal parameters: in the Weka software (Witten and Frank, 2005) employed in this article, the default cost value is $C = 1$.

In our experiments with this data set, we used $(1 - \frac{1}{CI})$ rather than CI in order to more closely compare the distribution to those of the other dimensions, which are heavily zero-weighted. Since all values in the set are positive, all three dimensions were then scaled such that $\mathbf{x}_i \in [0, 1]$, but were not centred on their means. Three binary SVM classifiers were trained independently on randomly-selected, class-balanced subsets of 100 sequences — 50 of the class to be selected, and 50 of any other class with the natural class distribution — using the above heuristic to determine optimal γ and C parameters for each classifier. A coarse grid search was also performed to compare classification results, and a high-resolution grid search is shown in Figure 2.7 to visualize the geography of the generalization error surface.

The results of these tests are illustrated in Figure 4.2 and summarized in Table 4.5. Our results did not match those of Shan *et al.* (2003), although the same software Weka (Witten and Frank, 2005) was used for the C4.5, NB and SVM-Lin classifiers: we used Weka version 3.4.3 (released September 29, 2004) for these three classifiers. For example, in Shan *et al.* (2003) the NB classifier was found to achieve 84.4% cross-validation accuracy on the highly non-separable Ambiguous vs. other classification, for 100 class-balanced samples. There were some differences in testing: we used 10-fold cross validation rather than 5-fold, to match the methodology in the other experiments in this thesis, and only $12\,625$ of the $17\,821$ known data points were available to us for these experiments. In addition, we have scaled our inputs to approximate i.i.d. data; in Shan *et al.* (2003) these inputs may have been scaled differently. We did not find any cases where the SVM-Lin classifier could not be properly trained (shown as NaN in Table 4.5).

We therefore reach a somewhat different conclusion: for all three binary classification problems, the SVM, with parameters optimized using a grid search, achieves the highest cross-validation accuracy on the 100 randomly-selected, class-balanced samples. The heuristic achieves a slightly lower cross-validation accuracy on the training set, but higher than

Table 4.5: Classifier performance comparison for the Protein Sequence Alignment Quality data set, with the results (†) obtained in Shan *et al.* (2003). Results shown for the heuristic in each case are the mean and standard deviation over ten runs, each with a different, randomly-selected set of 100 class-balanced training samples. Results for the Valid vs. Inadequate classification (1000 samples) are a single run only. *Optimization Accuracy* is the cross-validated classification accuracy obtained during parameter optimization, with 100 training samples, whereas *All Data Accuracy* is the total classification accuracy obtained by the classifier trained on the same 100 points, but tested on all known data points. Parentheses denote standard deviation. NaN indicates all values were classified as other.

Class	Search Method	Optimization Accuracy	All Data Accuracy
Valid vs. other	Heuristic	$84.7\ (3.9)$	$84.0\ (0.5)$
	Grid Search	$87.8\ (4.1)$	$83.5\ (1.4)$
	SVM ($\dagger = NaN$)	$80.5\ (3.4)$	$83.4\ (0.1)$
	C4.5 ($\dagger = 87.2$)	$81.2\ (3.5)$	$84.2\ (0.3)$
	NB ($\dagger = 55.4$)	$81.7\ (3.5)$	$84.0\ (0.4)$
Ambiguous vs. other	Heuristic	$68.7\ (5.4)$	$59.5\ (9.5)$
	Grid Search	$71.5\ (4.8)$	$58.5\ (6.4)$
	SVM ($\dagger = NaN$)	$62.5\ (7.0)$	$48.4\ (8.5)$
	C4.5 ($\dagger = 64.7$)	$60.3\ (4.7)$	$48.2\ (14.7)$
	NB ($\dagger = 84.4$)	$62.0\ (5.8)$	$47.2\ (7.6)$
Inadequate vs. other	Heuristic	$94.6\ (1.3)$	$94.4\ (0.6)$
	Grid Search	$96.4\ (1.8)$	$94.6\ (0.9)$
	SVM ($\dagger = 97.0$)	$94.1\ (2.2)$	$95.1\ (0.3)$
	C4.5 ($\dagger = 93.8$)	$93.8\ (3.7)$	$93.8\ (1.6)$
	NB ($\dagger = 96.6$)	$94.2\ (2.4)$	$94.7\ (0.3)$
Valid vs. Inadequate: 100 samples	Heuristic	$99.1\ (0.9)$	
Valid vs. Inadequate: 1000 samples	Heuristic	99.5	

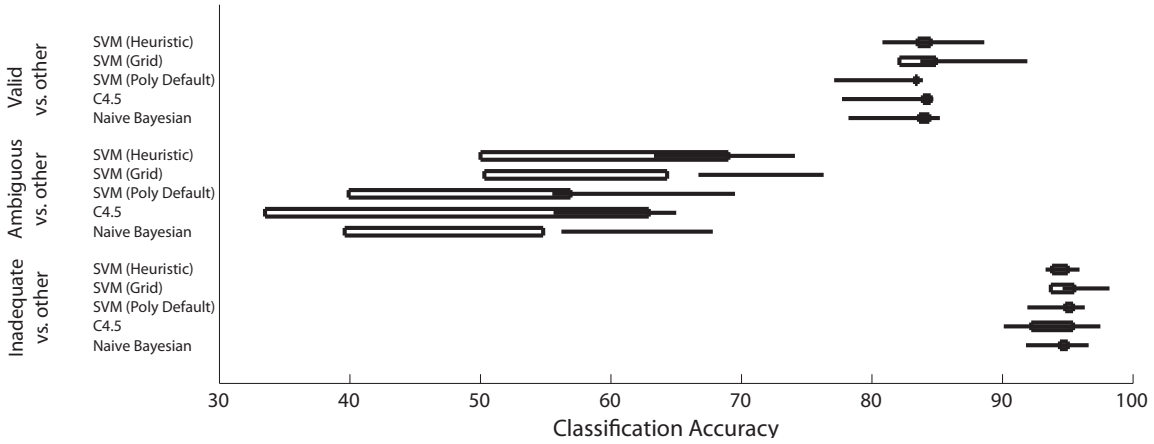


Figure 4.2: Visualizing classifier performance for the Protein Sequence Alignment Quality data set. For each of the three classification problems in this set, we compare the performance of an SVM using the proposed heuristic, an SVM using a grid search, a linear SVM and default parameters as Shan *et al.* (2003), a C4.5 tree algorithm as Shan *et al.* (2003) and a Naïve Bayesian classifier as Shan *et al.* (2003). For each classifier, the horizontal line shows the mean cross-validation performance, plus and minus the standard deviation, from training the classifier on 100 randomly-selected, class-balanced samples; and the box shows the overall classification accuracy, plus and minus the standard deviation, from training the classifier on the same 100 points then testing on the entire data set. The detailed data for this visualization is shown in Table 4.5.

the remaining three classifiers, in each of the three problems.

When the resulting models trained on these 100 data points are tested on the full data set, the overall classification accuracy drops somewhat for the grid search. However, importantly, this effect is much reduced for the heuristic: for the Inadequate vs. other classifier, for example, the overall classification accuracy for the SVM using a grid search drops from 96.4% to 94.6%, a drop of nearly 2%, but the SVM using the heuristic only drops from 94.6% to 94.4%, a drop of 0.2%. This effect is more drastic in the highly non-separable Ambiguous vs. other classification, in which that SVM using a grid search drops from 71.5% to 58.5%, but the SVM using the heuristic drops from 68.7% to 59.5%, becoming the best classifier in this category.

The remaining classifiers achieve comparable performance for the separable Inadequate vs. other and Valid vs. other classifications, but do significantly worse in the non-separable Ambiguous vs. other classification. The unoptimized SVM-Lin algorithm works well, with comparable performance to the NB and C4.5 classifiers in all three cases. However, for the non-separable classification problems, the optimized SVM using the RBF kernel are clearly

superior.

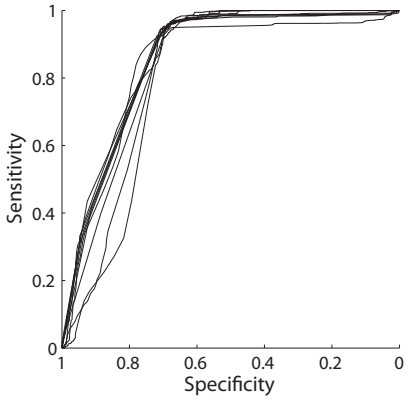
The overall results for the remaining two classification problems are very consistent between the five classifiers: there appears to be no clear winner in either case. To confirm this, a single-factor Analysis of Variances (ANOVA) was performed using the Excel (Microsoft) Data Analysis Toolpack. The results were somewhat inconclusive for the Inadequate vs. other and Valid vs. other classifiers, with P-values of 0.03 and 0.05 respectively, indicating that there is a high likelihood that these means are roughly equivalent in light of the variance: a P-value less than 0.05 may be considered significant (Norman and Streiner, 2003), but these values are both very close to 0.05. However, for the Ambiguous vs. other classifier, the P-value drops to 0.008, nearly an order of magnitude lower. This indicates that despite the higher variance for all classifiers in this binary classification problem, there is a high likelihood that the higher means for the Heuristic and Grid Search classifiers are *not* attributable to chance alone. We may therefore consider the differences between these means to be statistically significant.

We can conclude from these experiments that taking the model complexity into account in the heuristic allows the solution to be more general, preventing any overfitting as we expected, and that optimizing parameters, through any method, is greatly advantageous when the binary class distributions are highly non-separable.

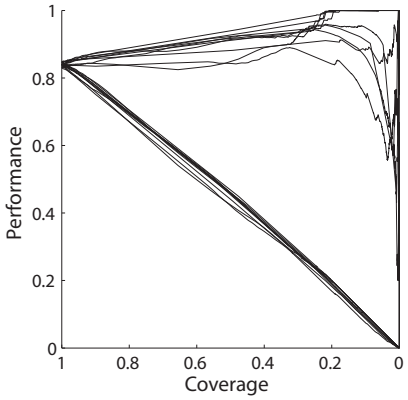
4.2.1 Visualizing Classifier Performance

To gain a better understanding of these models, Receiver Operating Characteristic (ROC) curves (see for example Hastie *et al.*, 2001) are shown in Figures 4.3 and 4.4 (*left*). The ROC curve plots the *sensitivity* of the classifier, or the fraction of correctly classified positive samples in the complete data set, against the *specificity*, or the fraction of correctly classified negative samples. Ten example curves are shown for each classifier, in order to show the variation between results. The Inadequate vs. other classifier (e) shows a large area under the ROC curves, with small variance between each of the ten curves: this indicates excellent classification performance. In contrast, much less area is shown in the Ambiguous vs. other classifier.

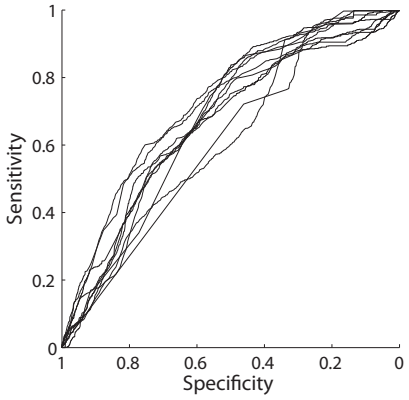
Another classifier performance visualization, which is perhaps more intuitive, is the Coverage-Performance (C-P) curve (Trappenberg, 2005) shown in Figures 4.3 and 4.4



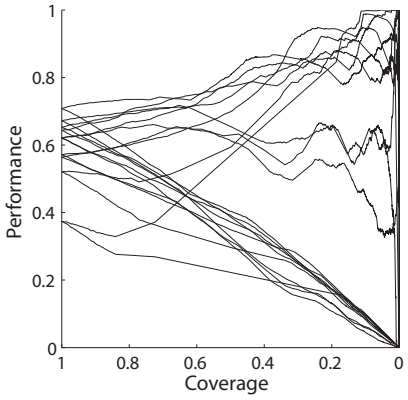
(a) Valid vs. other



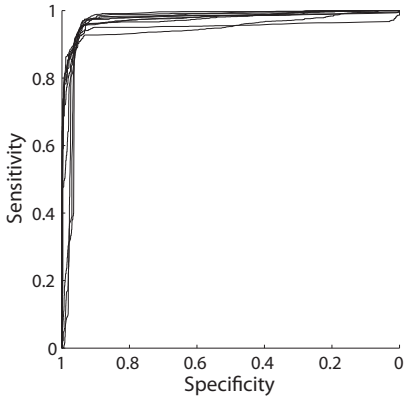
(b) Valid vs. other



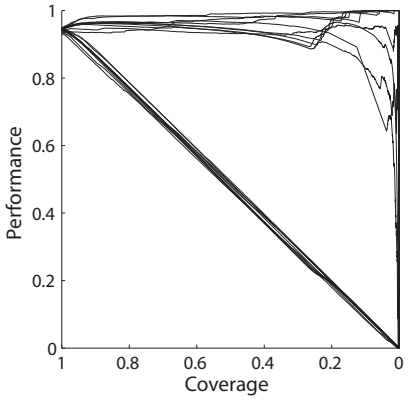
(c) Ambiguous vs. other



(d) Ambiguous vs. other

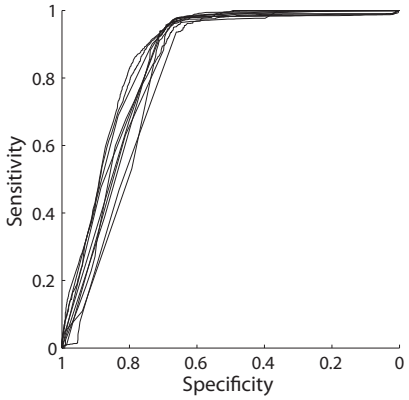


(e) Inadequate vs. other

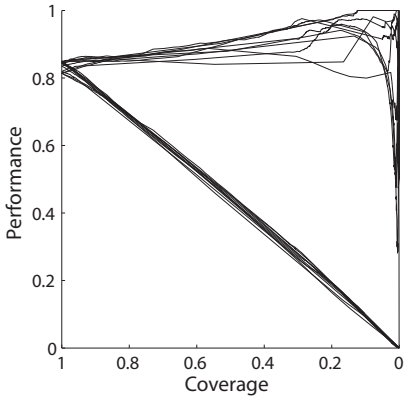


(f) Inadequate vs. other

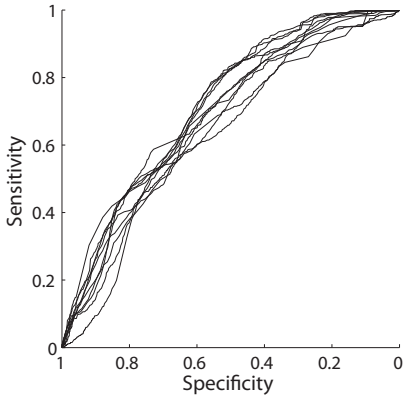
Figure 4.3: ROC curves (*left*) and C-P curves (*right*, see Equation 4.2) for the Protein Sequence Alignment Quality data set, with SVM parameters optimized using the heuristic.



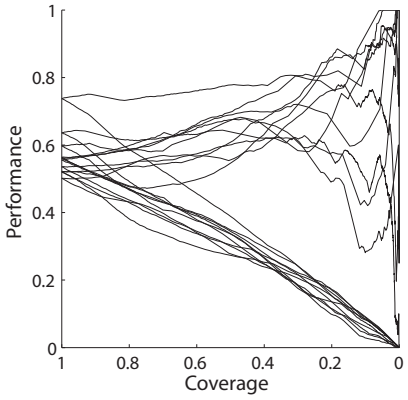
(a) Valid vs. other



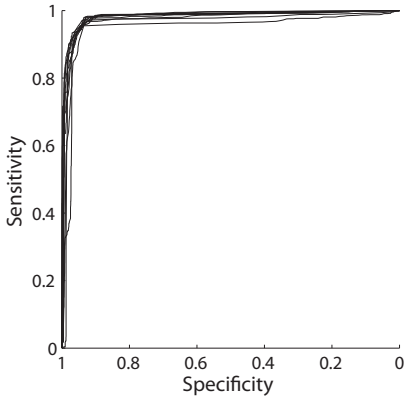
(b) Valid vs. other



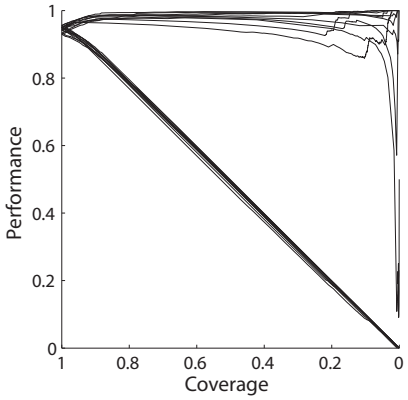
(c) Ambiguous vs. other



(d) Ambiguous vs. other



(e) Inadequate vs. other



(f) Inadequate vs. other

Figure 4.4: ROC curves (*left*) and C-P curves (*right*, see Equation 4.2) for the Protein Sequence Alignment Quality data set, with SVM parameters optimized using a grid search.

(*right*). These curves are advantageous in the case where much of the data is ambiguous, in the sense that there is a higher probability of overlap between the classes. Here, the coverage c is defined as (Trappenberg, 2005)

$$c = 1 - \frac{N_n}{N} \tag{4.2}$$

where N_n is the number of ambiguous samples, for which the decision surface returned by the classifier is below some threshold P_t, and N is the total number of samples. The lower curve is defined by (Trappenberg, 2005)

$$P_{\text{low}} = \frac{N_c}{N} \tag{4.3}$$

where N_c is the number of correctly classified samples. For this curve, unclassified (ambiguous) data are penalized as misclassifications. The upper curve is defined by (Trappenberg, 2005)

$$P_{\text{high}} = \frac{N_c}{N - N_n} \tag{4.4}$$

which does not penalize ambiguous data. These curves meet at the far left of each diagram, where $c = 1$: that is, no samples are classified as ambiguous. Notice that as the coverage decreases, the fraction of samples considered to be ambiguous increases, and the variation between each of these ten example curves also increases. As with the ROC curves, the C-P curves clearly show high variance between the ten examples for the Ambiguous vs. other classifier, whereas the other two classifiers exhibit much lower variance between example runs.

These results make intuitive sense, since the training data have been classified manually: it seems likely that an Ambiguous class will naturally have significant overlap with the Adequate or Inadequate classes. Therefore, due to the high accuracy of the Inadequate vs. other and Valid vs. other classifiers, we decided to train a fourth classifier to distinguish Inadequate vs. Valid, excluding the samples classified as Ambiguous during training. The results from these tests are also shown in Table 4.5 in the last two rows. The classifier was trained with 100 randomly-selected (but class-balanced) points, as with the other classifiers. This resulted in very high accuracy and consistency over ten runs of the fast-cooling heuristic with a different, randomly-selected set of training samples for each run. The standard deviation across these ten runs was also significantly lower than those of the first three classifiers. The fast-cooling heuristic was then trained on 1000 similarly-selected points, and achieved a 10-fold cross-validation accuracy of 99.5% with only 10 support vectors.

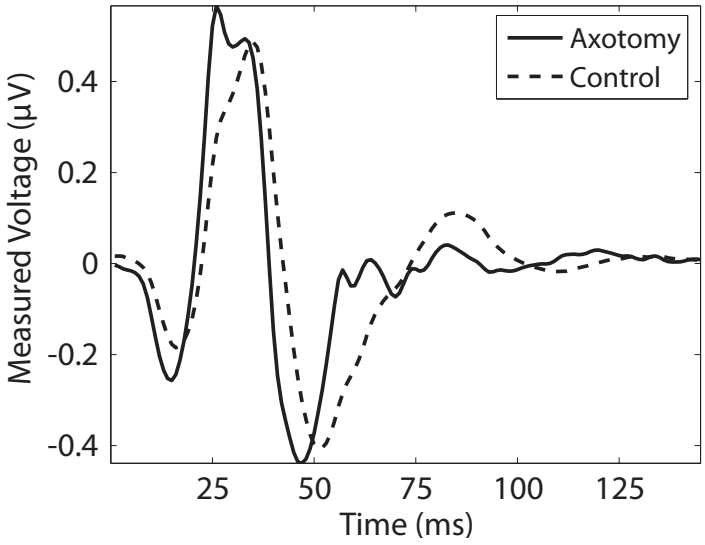


Figure 4.5: Mean pattern ERG waveforms for six axotomy and eight control subjects. The mean waveforms appear to be visually separable, but even to an expert, the actual observed waveforms may be somewhat ambiguous upon visual examination. See also Figure 7.6, in which we explore the relevance of each segment of these waveforms to binary classification.

This high accuracy, with low model complexity, indicates that with the samples classified as ambiguous removed from the training set, the classification problem becomes much more separable. This suggests an automatic method for including or excluding protein sequence alignments, in which those alignments with ambiguous quality are detected and removed from the training set — either manually or through some novelty-detection mechanism — so that a classifier is trained to perform this Inadequate vs. Valid classification, with statistical ambiguity decided from a chosen threshold as with the C-P curves in Figures 4.3 and 4.4. Such a classifier would free bioinformatics researchers to concentrate on the actual phylogenetic analyses, and may have the potential to increase the accuracy of those analyses.

4.3 Retinal Electrophysiology Data Set

Artificial neural networks (ANN) have long been used in medical research, due in part to their excellent cross-validation accuracy for the analysis of continuous variables (see for example De Roach, 1989). SVM are also primarily designed for the analysis of continuous variables, but are less frequently used in medical research despite the many advantages

such as a sparse model representation, excellent generalizability and the efficiency of calculations with comparable accuracy to ANN. An example of the use of SVM as a novelty detection mechanism, in a hybrid technique employing wavelet feature extraction for the analysis of biomedical waveforms, may be found in Strauss *et al.* (2003).

The retinal electrophysiology data set used in this thesis comes from pattern electroretinography (PERG). Axotomy procedures were performed on female domestic pigs (*Sus domesticus*), each approximately six months of age, as part of an ongoing medical research project examining the electrophysiological contributions of bipolar and ganglion cells in the retina.

A control PERG measurement under ketamine anaesthesia was first taken from each subject for comparison (isoflorane was found to adversely affect the ganglion contribution of normal subjects). The axotomy procedure then severed the optic nerve, removing the axons of all ganglion cells in a minimally-invasive procedure. After approximately six weeks, to allow phagocytosis processes to completely consume any remaining ganglion cells in the retina, another PERG measurement was taken.

Gain settings were held constant across all measurements for all subjects. Data from 103 high-contrast chequer locations displayed on a 75 Hz source were averaged using an m-sequence over approximately 2.5 minutes. This procedure results in a $103 \times 145 = 14\,935$ point vector for each observation. A mean of all chequer locations was generated to form a 145 point waveform, corresponding to 145 ms at a 1000 Hz sampling rate. In-depth discussions of similar methods may be found in Marmor *et al.* (2003); Sutter and Tran (1992).

The preliminary data used in this thesis has 14 observations in two classes: six *axotomy* and eight *control*. The mean waveforms for each class are shown in Figure 4.5. To perform the classification, the raw waveform was input as a 145-dimensional vector. Each dimension was centred and scaled independently by the mean and magnitude, such that the resulting inputs had values as Equation 4.1. To more closely approximate a balanced data set, the classifications were performed on each of the possible $\binom{8}{6} = 28$ combinations of 12 balanced subsets, and the resulting accuracy was taken as the mean across all 28 runs, giving equal weight to all experiments. Due to the low number of samples, leave-one-out cross-validation was used to assess the generalization error at each evaluated point: this is equivalent to N-fold cross-validation where $N = \ell$, and results in lower noise in the error surface as the random-partitioning is replaced by a complete, combinatorial search.

Table 4.6: Sample results from retinal electrophysiology classification, comparing a grid search to the slow-cooling heuristic. The mean over 28 combinations of 12 balanced subsets, from six axotomy and eight control subjects, are shown.

Search Method	Accuracy	n_{sv}	Evals.
Fast-Cooling Heuristic	98.5	6.2	660
Slow-Cooling Heuristic	98.5	6.2	6880
Grid Search	99.4	8.5	6603

The results of the classification are shown in Table 4.6. We have found excellent generalization performance using the heuristic with this data set: the fast-cooling heuristic found the same accuracy and complexity as the slow-cooling heuristic but with far fewer evaluations. Both found only a small reduction in cross-validation accuracy in comparison to a grid search, with approximately 25% reduction in complexity as measured by the average number of support vectors in the 28 models.

4.4 Discussion

Empirically, we have found that the noisy nature of N-fold cross-validation has made little appreciable difference in the results when using the stochastic approach, since points near the end of the search will likely be quite nearby as the temperature T decreases. However, one often-overlooked step which can have a significant effect when using SVM is to ensure the i.i.d. inputs necessary for optimal classification (Burges, 1998; Vapnik, 1995). For example, one can centre and scale the inputs such that all dimensions have zero mean and values $x_i \in [-1, +1]$, as we do for most problems in this thesis. However, when faced with noisy, volatile input data such as the sensor waveforms examined in Section 4.3, there may be peaks in future observations that were not seen in the training data, thereby breaking these arbitrary bounds on the input vector. A common alternative is to centre and scale to zero mean and unit variance, but in our tests we found this reduced the accuracy somewhat, since this assumes a Gaussian distribution for all dimensions: in waveform classification, we may often see very different distributions, such as a bimodal or Poisson distribution.

Variable selection methods may be used to limit the number of attributes available to the classifier, in order to improve computational efficiency. For example, indicative sections of the waveform may be selected by expert users, or separability measures such as the Fisher

Ratio, Pearson Correlation Coefficients or Kolomogorov-Smirnoff statistics may be used to select particular variables in the waveform. We will examine some of these methods in Chapter 7, but for the classification and regression problems addressed elsewhere in this thesis, we include all known dimensions in each data set. Modification of the input variables through normalization might be useful for the retinal electrophysiology data set, since multiple ERG machines might be used with varying gain settings. For example, we might normalize each waveform by the mean of the trailing samples. However, these particular tests were all performed on the same machine with gain settings held constant, so no normalization was needed for the experiments in this thesis.

Chapter 5

Regression Results

Introduced in Section 2.1.6, ε-SVR allows reduced sensitivity to noise by providing a small noise threshold ε, an estimate of the constant, additive noise present in the target values. Target observations that are within $\pm\varepsilon$ of the predicted target values will not be penalized, but those outside this ε-tube will be penalized according to the cost value C.

In this chapter, we extend the heuristic to univariate regression using ε-insensitive Support Vector Regression (ε-SVR). We adapt the heuristic to simultaneously optimize three parameters: the cost parameter C, the RBF width parameter γ and the noise-insensitive tube width ε. We find that when optimizing in such a three-dimensional parameter space, it is advantageous to extend the cooling schedule somewhat to evaluate a larger number of points.

5.1 Introduction

At the 2005 IEEE International Joint Conference on Neural Networks (IJCNN 2005), a special session was held on applying machine learning techniques to environmental modelling (Cherkassky *et al.*, 2005). That same year, at the first Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) Challenges Workshop, a machine learning competition was held on techniques for evaluating losses in probabilistic prediction (Candela *et al.*, 2005). These events inspired a new machine learning challenge, to be discussed at a special session in IJCNN 2006. The *Predictive Uncertainty in Environmental Modelling Competition* (Cawley, 2006) was conceived and operated by Gavin Cawley at the University of East Anglia.

The objective of the competition was to perform a regression analysis on three real-world environmental data sets, detailed below, then use this analysis to predict the target value for a number of test observations. However, rather than simply providing the best possible prediction, competitors were also to provide an estimate of the expected distribution of that prediction: in other words, competitors were to predict the uncertainty of their prediction for each target.

In the competition, this *heteroscedasticity* may be represented in three forms (Cawley,

2006):

1. The competitors may assume a Gaussian distribution, and give the mean and variance of that distribution along with each prediction.

2. The distribution may be modelled by a Gaussian Mixture Model (GMM), with any number of Gaussian distributions summing to provide an estimate of the target distribution, by providing the mean, variance and weight of each Gaussian in the mixture.

3. The distribution may be modelled by a number of quantiles (Candela *et al.*, 2005), by giving the width and cumulative probability of each quantile. Exponential tails on either side of the distribution assure that the probability distribution integrates to unity.

The set of predictions for each data set from each competitors were to be evaluated on two measures. The first was the mean squared error (MSE) of the set of n target predictions $y_i \in \mathbb{R},\ i = 1, \ldots, n$ in the test data set, as (Candela *et al.*, 2005)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{\sigma^2(y)} \tag{5.1}$$

where μ_i is the mean of the predicted conditional probability $p(y_i|x_i)$. The metric is normalized by the variance $\sigma^2(y)$ of the target values.

The second measure to evaluate performance was the mean negative log estimated predictive density (NLPD) of the predictions, as (Candela *et al.*, 2005)

$$NLPD = -\frac{1}{n} \sum_{i=1}^{n} \log\, p\,(y_i|x_i) \tag{5.2}$$

This measure penalizes predictions which are accurate in terms of MSE, but which are "under-confident" or "over-confident" (Cawley, 2006) in their estimate of the predictive uncertainty $p(y_i|x_i)$.

5.2 Data

Three "real-world" data sets were provided for analysis, with a fourth synthetic data set provided for experimentation. Each data set was split into Training, Validation and Testing

Table 5.1: Dimensions of the provided data sets in the Predictive Uncertainty in Environmental Modelling Competition. *Source:* Cawley (2006).

Data Set	Features	Training Observations	Validation Observations	Testing Observations
Precipitation	106	7031	3515	3517
Temperature	106	7117	3558	3560
Sulphur Dioxide	27	15 304	7652	7652
Synthetic	1	256	128	1024

partitions. The Testing partition remained unseen until after the competition deadline. The dimensions of these data sets are summarized in Table 5.1.

The *Precipitation* and *Temperature* data sets (Cawley, 2006) include precipitation and temperature measurements made by an environmental monitoring station. The input features are meteorological information obtained by a large-scale General Circulation Model (GCM) (see for example Russell *et al.*, 1995). From a machine learning perspective, these data sets are challenging due to the larger number of input variables. The *Sulphur Dioxide* (SO_2) data set (Cawley, 2006) contains less than a third as many input variables, but has more than twice the number of sample observations. It is based on forecasting the concentration of atmospheric SO_2, based on current SO_2 levels and other meteorological conditions.

Noise processes in such environmental data sets are thought to be non-Gaussian (Cawley, 2006), such that a model that assumes a Gaussian probability distribution for the target uncertainties may be inaccurate. It seems likely that the data sets in this competition were specifically chosen such that a solution assuming Gaussian distributions would not perform well when measured by the NLPD metric. However, we will make such an assumption here as we find this gives adequate regression performance: on the Synthetic benchmark, for example, our results compare well with the winner of this category and the overall competition, Markus Harva, who used a GMM representation of the distributions.

5.3 Methods

The optimum noise-insensitive tube width ε may be chosen *a priori* if the expected noise density for the data set is known (Smola and Schölkopf, 2004; Vapnik, 1995), or may

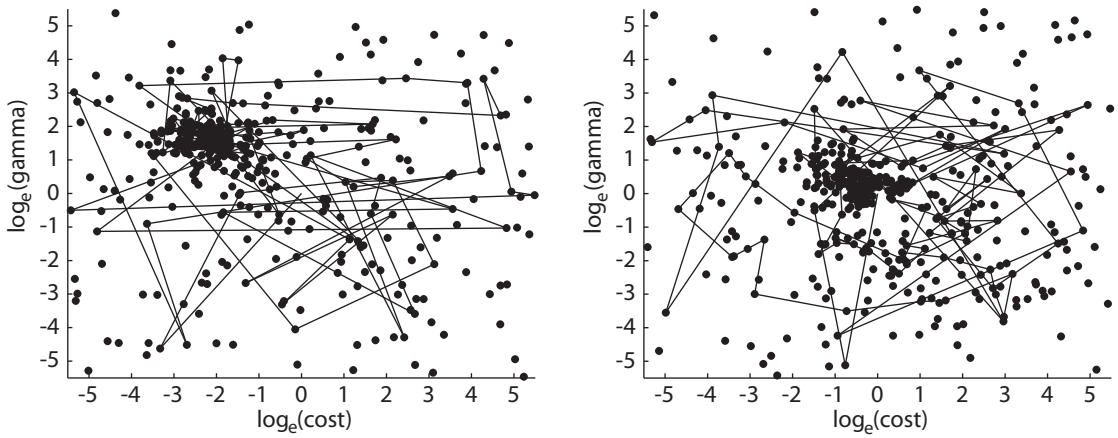


Figure 5.1: Two-dimensional projections of the stochastic path followed by the regression heuristic used in this chapter, through a three-dimensional parameter space defined by the cost parameter C, the RBF width parameter γ and the ε-insensitive tube width. The dots correspond to evaluated points in parameter space, and the thin lines represent the path taken by the heuristic. On the *left*, we show the path followed by the first part of the heuristic applied to the Synthetic data set, evaluated by MSE as detailed in the text. On the *right*, we show the path followed by the second part, evaluated by NLPD. Both paths converge to a single point in parameter space. However, the path is much more erratic than we would see with a two-dimensional path such as Figure 2.6, since here the z-axis — containing the path information for ε — is not shown.

be estimated empirically (Cherkassky and Ma, 2004). Here, we show that the optimum value of ε may be numerically estimated through a minor modification of the proposed heuristic that optimizes the cost parameter C and RBF width parameter γ in Chapter 4. The heuristic is extended to three-dimensional parameter space in order to simultaneously find the optimum noise-insensitive tube width ε. Examples of a two-dimensional projection of the resulting stochastic path through this parameter space are shown in Figure 5.1.

The data sets in this chapter have a large number of observations available for training and validation. Therefore, for each evaluated point in parameter space, rather than using the number of support vectors to perform the extrinsic regularization as with the classification experiments in Chapter 4, we train a model on the training data as before, but evaluate the model's performance on the validation partition. This prevents the model from overfitting to the training observations, as shown in Figure 2.5, since none of the training set — and therefore none of the support vectors — appear in the validation set. Generalization performance on the training set is ignored, creating a general solution without the need to calculate a model complexity measure. An example of a general model resulting from this

approach is shown in Figure 5.2 (*right*).

Making the assumption of a Gaussian probability distribution for the uncertainty of each target, the solution to this regression problem can be broken into two parts: first, we perform a typical regression analysis, optimizing the ε-SVR free parameters C, γ and ε on a model trained on the training partition in order to minimize the MSE on the validation partition. This prediction value gives us the mean of the target probability distribution at each evaluated point in the test set.

Next, we perform a second regression analysis to predict the absolute *difference* between the predictions and the training set targets, and optimize the same parameters again on a model trained on these difference in order to minimize the NLPD on the differences between the model and observation for the validation partition. The square of this prediction of the estimated difference, between the target value and the predicted value, gives the variance of the target probability distribution at each evaluated point in the test set: the predicted uncertainty of the target prediction.

This two-part process is illustrated in Figure 5.2 for the Synthetic data set. A 6th-degree polynomial regression, using this same two-part process, is shown for comparison. Both the polynomial regression (*left*) and ε-SVR (*right*) approaches result in smooth, general models with no overfitting. The models appear to be quite similar, in fact the means of these models appear to be identical. However, the ε-SVR model appears to have more precision in the estimated deviation at the extremes of the input value, where $\mathbf{x}_i \leq 0.5$ or $\mathbf{x}_i \geq 3.0$, whereas the polynomial model appears to have more precision in the centre of the distribution, where $1.5 \leq \mathbf{x}_i \leq 2.5$, reacting to the higher spread of the empirical deviations in this range to a slightly greater extent.

For the Synthetic data set, this process completes quite quickly using the same cooling schedule as with the classification results in Chapter 4. To perform this search efficiently for the larger data sets, two steps were taken to reduce computational complexity. First, the cooling schedule of the simulating annealing heuristic was evaluated from $T_0 = 100$ to $T_C = 1$ (rather than $T_C = 0.1$ as in Chapter 4) in steps of either $\delta = 0.1$ (Precipitation), $\delta = 0.05$ (SO_2) or $\delta = 0.025$ (Temperature), resulting in 440, 900 and 1820 evaluations respectively. Second, the number of points included during optimization was a random selection of 1000, 1000 and 1500 points respectively for each data set, although in all cases, the full set of training observations was used to train the final models before predictions were performed. These choices are summarized in Table 5.3.

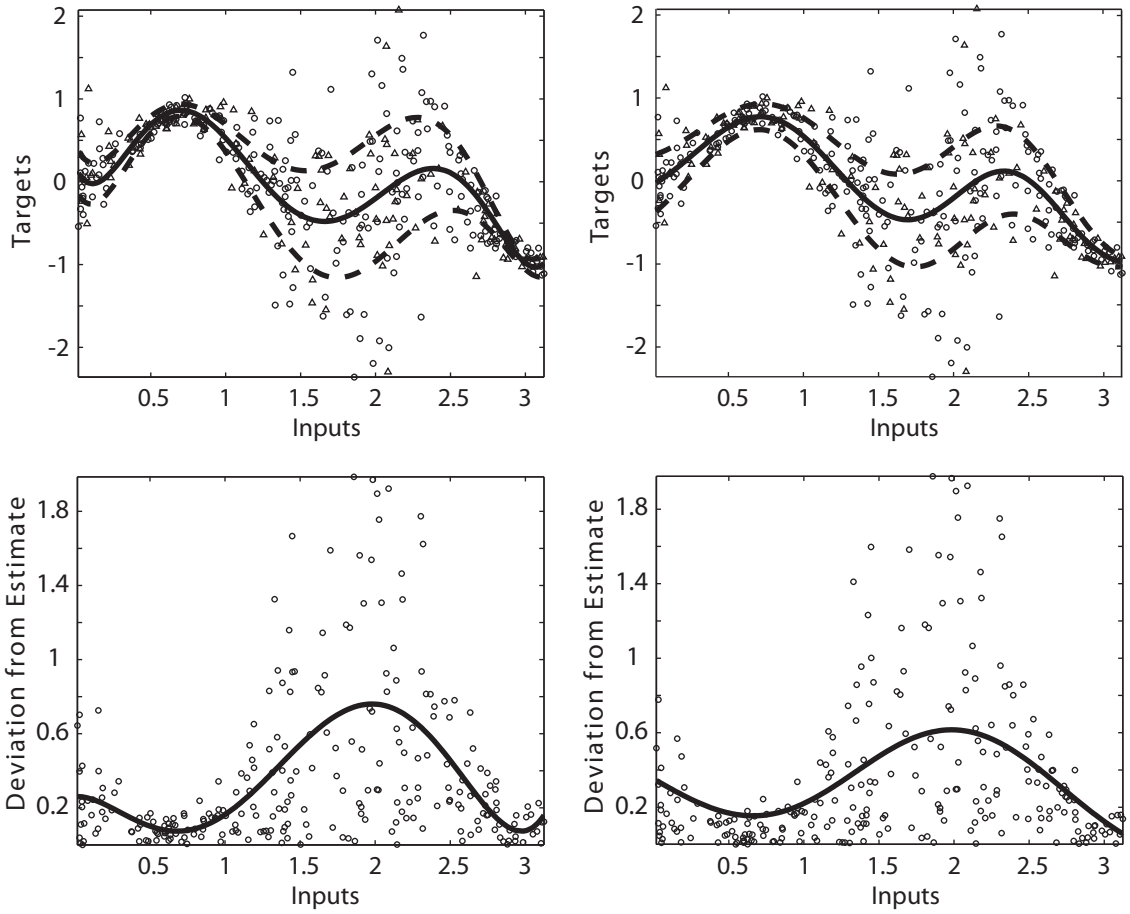


Figure 5.2: Comparing polynomial regression using a 6th degree polynomial (*left*) with ε-SVR regression (*right*) for the Synthetic data set. In the *upper* diagrams, circles ($\circ$) represent the training set, whereas triangles ($\triangle$) represent the validation set. The solid line indicates the prediction values, and the dotted lines represent the predicted standard deviation from the prediction. In the *lower* diagrams, the circles ($\circ$) represent the difference between the prediction and target values for the training set. The solid line is the resulting estimate of the standard deviation from the predicted values.

Table 5.2: Final results of the NLPD metric on unseen Test partitions for Temperature, Precipitation and Sulphur Dioxide data sets, sorted by best mean NLPD across all three data sets. The overall winner of the contest was Markus Harva, who placed just after the contest organizer Gavin Cawley. Considering that we use a Gaussian representation of the uncertainty distribution, with blind application of the heuristic and no parameter tuning for any specific data set other than varying the length of the cooling schedule, a fourth place finish seems quite reasonable: M. Harva used a GMM representation of the distributions, whereas T. Bagnall, G. Cawley and VladN all appear to have used quantile representations. *Source:* Cawley (2006).

Name	Precipitation	Sulphur Dioxide	Temperature	Mean
(G. Cawley)	-0.510	4.255	0.053	1.266
M. Harva	-0.279	4.370	0.202	1.431
VladN	1.272	4.616	0.108	1.999
T. Bagnall	1.114	4.758	0.136	2.003
M. Boardman	1.606	5.090	0.076	2.257
S. Kurogi et al.	3.098	11.01	0.059	4.721
I. Takeuchi	0.747	6.043	24.79	10.53
I. Whittley	–	–	0.625	–
E. Snelson	–	–	0.035	–

5.4 Results

The final results of the competition are summarized in Table 5.2.

We would not expect these methods to be very effective in this contest in comparison to more sophisticated models, for two main reasons. First, we are representing the model by Gaussian mean and variance which assumes a Gaussian distribution, whereas other models may represent the model with the far more flexible quantile or GMM representations. Second, we are using ε-SVR which assumes a constant noise-insensitivity threshold ε. However, even from the synthetic data set shown in Figure 5.2, it is clear that this is not an appropriate assumption for this data: it seems likely that these data sets were specifically chosen such that a Gaussian representation of the distribution would not perform well. These compromises were made to favour blind application of the existing heuristic, without radical modification.

In addition to these factors, for the three real-world data sets, we have also somewhat

compromised the parameter optimization by limiting the number of observations used during the search heuristic. We have also limited the stochastic search to a fast- or moderate-cooling schedule, rather than the slow-cooling schedule used in the Synthetic benchmark, in order to reduce computational complexity.

Despite these allowances, we find that the method returns quite favourable results. In the Synthetic data set, for example, due to the small number of observations with only a single input dimension, using the slow-cooling heuristic with 100% of the training samples was quite feasible. In this case, the model gave a mean square error only slightly higher than the "ground truth" reference by the contest authors, which was generated from the original model used to create the actual Synthetic data set.

For the real data sets, however, it was necessary to use a faster cooling rate and a smaller portion of the training data, randomly selected at each evaluate point in parameter space, in order for the calculations to be complete in a reasonable amount of time. For the real data sets, these calculations finished within one or two hours, with the exception of the Temperature data set, for which a somewhat slower cooling rate and somewhat larger portion of training data was used to extend the training time to about eight hours on our test machine, a 3.4 GHz Intel Pentium IV. Performing similarly slowed searches on the other two data sets should improve results somewhat.

In terms of MSE on the test data partition of these real data sets, the optimized ε-SVR model appears to be roughly comparable to a Bayesian-regularized multilayer perceptron (MLP) entered by Gavin Cawley: by this measure, comparable results were found for all four regression problems. However, in terms of NLPD, the results seem more comparable with VladN's submissions, which used a gradient-based optimization with details to be announced at the competition (Cawley, 2006). Both VladN and Gavin Cawley appear to have used a more complex quantile representation for the uncertainty distributions, which may partially account for the higher precision of these models.

With slower cooling and a higher percentage of the training data used, increasing computational complexity, these numbers may improve somewhat. For example, for the Sulphur Dioxide (SO_2) data set, two entries were submitted: the first used the fast cooling schedule (440 evaluations) as with the Precipitation data set, but this yielded poor results for the NLPD metric on the Validation partition. A second entry was therefore submitted with a somewhat slower cooling rate (900 evaluations), resulting in a somewhat improved

Table 5.3: Detailed results of the MSE and NLPD for each of the four data sets, for the submitted entry using the proposed two-part method in this Chapter. The winner in each category, by minimum MSE and by minimum NLPD on the Testing partition, are shown for comparison. *Source:* Cawley (2006). This table also shows the number of evaluated points in parameter space performed by the heuristic, and the number of training samples considered during parameter optimization.

		Validation		Testing			Optimization
Data Set	Name	MSE	NLPD	MSE	NLPD	Evals.	Samples
Synth.	Winner by both	0.475	0.313	0.562	0.386		
	Heuristic	0.473	0.313	0.566	0.475	4590	256/256
Precip.	Winner by MSE	0.535	0.463	0.611	0.747		
	Winner by NLPD	0.609	-0.436	0.631	-0.510		
	Heuristic	0.611	1.556	0.644	1.606	440	1000/7031
Temp.	Winner by both	0.061	-0.042	0.066	0.035		
	Heuristic	0.069	0.055	0.071	0.076	1820	1500/7117
SO_2	Winner by MSE	0.479	2.519	0.688	6.043		
	Winner by NLPD	0.776	4.292	0.799	4.255		
	Heuristic	0.836	5.167	0.840	5.090	900	1000/15304

MSE (from 0.87 to 0.84) but a greatly improved NLPD (from 12.1 to 5.1). With the fast-cooling schedule, the Temperature data set appeared to give good overall results by both measures, so a moderate-cooling schedule was employed with a higher number of training observations, in order to improve results further.

5.5 Discussion

In Table 5.3, we compare the results obtained using this two-part heuristic with those of the winner in each category, by minimum MSE and minimum NLPD. For the Synthetic data set, our results closely match the winner, even outperforming the winner on the Validation partition in terms of MSE, but fall somewhat behind in terms of NLPD: in this case the winner, Markus Harva, used a Gaussian Mixture Model to represent the uncertainty probability distribution, allowing greater precision. For the remaining data sets, the heuristic performs quite well, with comparable results for each category. On the Testing partition of the SO_2 data set, for example, the heuristic results in a better NLPD than that of the entry with the best MSE.

Further steps could be taken to improve the accuracy of this analysis. The heuristic performs very well on the Synthetic data set, and performs adequately on the real-world data sets. We have demonstrated that extending the cooling schedule, to evaluate a greater number of points in parameter space, is advantageous for both metrics. We might also combine the training and validation sets, then randomly partition this superset of observations into training and validation partitions at each evaluated point in parameter space, thereby allowing many more points to be available for assessing cross-validation performance. A more advanced analysis might represent the uncertainty probability distributions as quantiles or GMM, rather than assuming a simple Gaussian distribution: this would add a number of additional free parameters which might be searched by a straightforward modification of the simulated annealing heuristic. Finally, as mentioned in Section 5.3, here the extrinsic regularization is performed by evaluating performance on a separate validation data partition, rather than a separate complexity measure: taking both factors into account may well result in a more general solution.

Chapter 6

Multivariate Regression Results

In this chapter, we extend the heuristic to multivariate regression problems — those with a multidimensional output as well as a multidimensional input — and apply the heuristic to the detection of periodic gene expression in DNA microarray experiments.

Noise levels and cross-study variation present in gene-expression data from DNA microarray experiments create obstacles for genomic researchers. A reliable method for modelling such data is required in order to impute missing observations, and to improve the signal to noise ratio in time-variant or cross-study experimental results. In this chapter, we combine multivariate support vector regression and non-linear, periodic curve-fitting methods to model differential gene-expression in periodic microarray data. Support vector regression makes no assumptions about the distribution of the underlying data model, seeking a general, regularized solution for data sets with a low number of samples but with high dimensionality. The cleaned microarray data from this model may then be analyzed further through other methods: for example, in this chapter, we apply an additive, periodic model to detect regular periodicity, such as the transcription of mitotic genes during a cell's reproductive cycle.

We apply these methods to a recent study of cell-cycle synchronization methods in fission yeast, *Schizosaccharomyces pombe*, and evaluate the models in comparison to univariate polynomial and linear regression approaches common for microarray data imputation. In this study, our goals are to impute missing data points in a periodically meaningful context, to determine which genes exhibit high periodicity that is strongly synchronized with the cell-cycle and to determine the most-likely activation point of each gene within the cycle.

6.1 Introduction

DNA microarrays measure the expression levels of active genes in an organism, by detecting and quantifying particular strands of messenger RNA (mRNA) which are transcribed from a cell's DNA during protein synthesis (Orengo *et al.*, 2003). In this chapter, we examine the modelling of differentially-expressed microarray data without regard to the

underlying cause of errors. Our approach is based on two data models: first a multivariate, kernel-based, non-linear regression using Support Vector Machines (SVM); and second a univariate, maximum-likelihood model using non-linear curve-fitting, with both a linear and non-linear, periodic component. Our goal is to create models which can be used to impute missing observations, for data correction and normalization, or which can in themselves be used for further analysis.

Noise and errors arise in DNA microarray hybridization experiments from a variety of factors, including gene-specific dye bias (Martin-Magniette *et al.*, 2005), probe and experiment design (Smyth *et al.*, 2003), culture heterogeneity (Gilks *et al.*, 2005), variations in slide quality and manufacturing processes which can create surface abnormalities or allow slide-movement within the microarray scanner (Agilent, 2005) and the normal deterioration of mRNA post transcription (Orengo *et al.*, 2003). A comprehensive overview of many of these sources of experimental and analytical errors, and common statistical techniques used to overcome them, can be found in Smyth *et al.* (2003).

Many other statistical approaches have been applied to microarray data analysis, such as univariate or multivariate Analysis of Variances models (ANOVA or MANOVA) (Gilks *et al.*, 2005; Kerr *et al.*, 2000) commonly used for normalization prior to further analysis (Smyth *et al.*, 2003). Independent Components Analysis (ICA) (Martoglio *et al.*, 2002) appears to have significant potential for automatic artefact isolation and removal, by maximizing the statistical independence of the resulting signals.

Univariate or multivariate regression approaches are common in microarray analysis (Tsai *et al.*, 2004; Wu, 2005). The majority of these are univariate, in which a single output variable is targeted (although there may be one or many input dimensions). In addition to reducing input noise and sources of error, univariate regression models have been used for classification and prediction (Choi *et al.*, 2003; Wu, 2005). More recently, multivariate regression approaches, in which the output is a vector rather than a single value, have also been suggested. Gilks *et al.* (2005) proposed a multivariate, linear regression technique based on a controlled design matrix to fuse data from multiple, similar experiments, in which the output is a fused, cleaned, time-variant microarray experiment for periodic data. Choi *et al.* (2003) also fuse multiple, time-variant data sets using a covariance measure for Bayesian meta-analysis and apply their algorithm to the problem of cancer profiling. Johansson *et al.* (2003) also used a multivariate approach, applying an algorithm based

on Partial Least Squares (PLS) to fuse time-variant data sets for budding yeast, *Saccharomyces cerevisiae*: the authors of this study note that an advantage of PLS is to obtain models with high generalization performance for data sets with few observations but high dimensionality, which as we note in Section 7.1, is also a significant advantage for SVM. Tai and Speed (2004) have proposed a Bayesian approach of similar form, but which uses a Bayesian statistic to approximate this design matrix automatically and from which the goal is to create a vector of expression values for each individual gene probe, similar to the approach taken here.

Additive data models have also been proposed, such as Tsai *et al.* (2004) in which a linear and non-linear model are combined, an approach similar to the periodic model we apply in this chapter, but using a sequential normalization algorithm rather than a non-linear maximum likelihood model. A non-linear maximum likelihood model was proposed in Huber *et al.* (2002), in order to normalize data prior to more complex analysis, however such non-linear transformations will negate the assumption of an additive error or residual component (Gilks *et al.*, 2005).

Data imputation methods attempt to find the most likely value of missing observations and minimize noise levels in the gene expression data to identify the most-likely underlying signals. However, in some cases it may not be necessary or warranted to identify each individual gene-expression signal in the data: we may simply wish to measure the goodness-of-fit to an additive, periodic model in order to isolate a particular component of the underlying signal relevant for a particular biological analysis, or to impute missing pieces of source data in a periodically meaningful way rather than employing a statistical averaging technique such as K-Nearest Neighbors (KNN) (Gilks *et al.*, 2005) which is known to perform badly in data imputation in terms of RMSE (Troyanskaya *et al.*, 2001; Wang *et al.*, 2006). A comparison of several methods for data imputation is provided in Jörnsten *et al.* (2005), who provide a new method based on convex linear combination of several current methods, and in Troyanskaya *et al.* (2001), who compare SVD and KNN with a row-oriented mean for several real data sets.

Further analysis of the cleaned data set to identify periodically expressed genes during cell processes to identify mitotic genes has been performed via clustering (Rustici *et al.*, 2004) and Singular Value Decomposition (SVD) (Gilks *et al.*, 2005) or Principal Components Analysis (PCA) (Johansson *et al.*, 2003) which find signals with maximum variance, although a more common approach is to use a statistical ranking techniques such as the

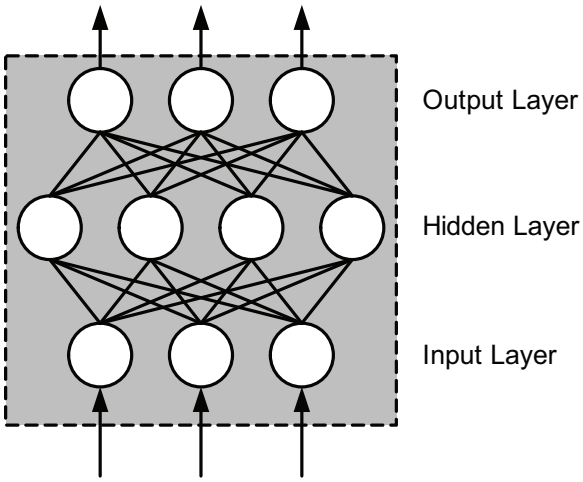
commonly used t-statistic (Smyth *et al.*, 2003). Here we will use the mean squared error (MSE), normalized by the variance as with Equation 5.1, as a goodness-of-fit statistic.

6.1.1 Multivariate Support Vector Regression

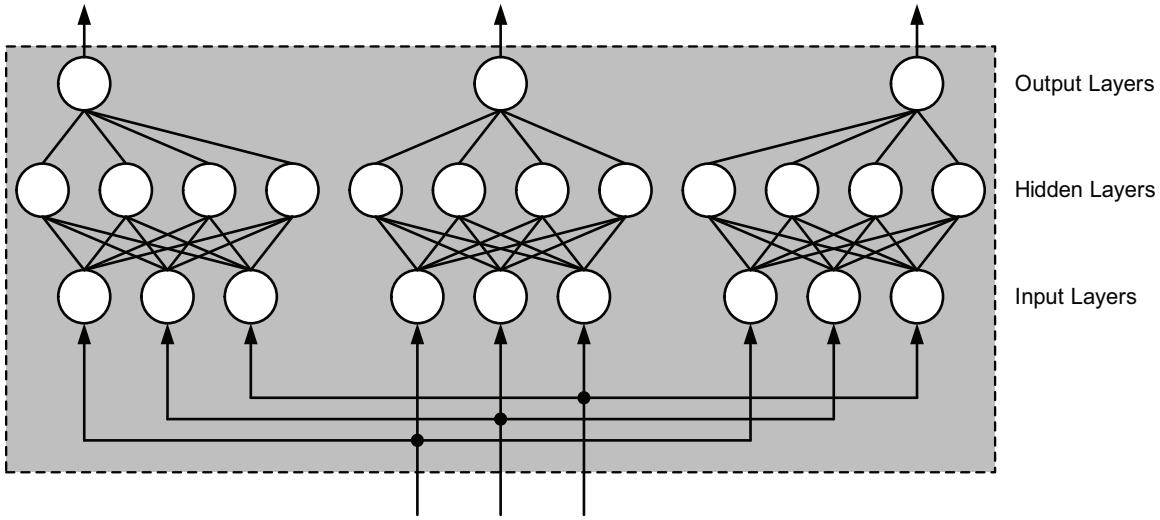
Supervised machine-learning techniques such as Artificial Neural Networks (ANN) have been used for classification in microarray research (see for example Brown *et al.*, 2000). Regression models based on SVM are less common, as the technology is comparatively recent (Drucker *et al.*, 1997; Smola and Schölkopf, 2004; Vapnik *et al.*, 1997).

True multivariate SVM regression has been proposed and implemented (Tsochantaridis *et al.*, 2004). However, its use is not widespread and the theory behind truly multivariate SVM regression is still being developed. The approach we take here might be more properly called *multiple* regression, since we build a series of univariate regression models, each of which is trained by the same vector of input genes but target a different output gene. Each model also uses the same cost parameter C, RBF width parameter γ and ε-insensitive tube width. The difference between these philosophies is illustrated in Figure 6.1: interdependencies between the outputs are only available in a truly multivariate approach. We will refer to this technique as multivariate regression in this work as a convenience, in the sense that we generate a "black box" model with multivariate inputs and outputs.

In Wang *et al.* (2006), Support Vector Regression was shown to be superior to K-Nearest Neighbors (KNN), Bayesian Principal Components Analysis (BPCA) and Local Least Squares (LLS) for data imputation on microarray data sets, in terms of mean squared error normalized by individual gene variance. As in our analysis here, in Wang *et al.* (2006) multiple univariate regression models are employed. The free parameters ε, γ, C for the ε-SVR implementation in this study were determined through a grid search, whereas here we adapt the simulated annealing heuristic from Chapter 3. Since the number of training observations for each gene is small, we use a model complexity measure to perform extrinsic regularization as in Chapter 4, but include optimization of the ε parameter as in Chapter 5. These optimum parameters were determined individually for each column (time point) in Wang *et al.* (2006), whereas here we use a representative sample of genes in order to further enhance generalization and reduce the computational burden. Finally, the column-wise orthogonal input coding scheme to flag missing values in Wang *et al.* (2006) was used to alleviate a restriction that only a single missing value could be estimated for each row (gene)



(a) True multivariate model.



(b) Multiple univariate models.

Figure 6.1: A conceptual comparison between true multivariate regression and regression using multiple univariate models. Both "black box" neural network models have three inputs and three outputs, which might be considered multivariate. However, the implementation in (a) uses true multivariate outputs, where the outputs are interdependent. The network in (b) treats the outputs as three entirely-separate models. This is the approach we take in this chapter as a convenience, and which is also performed in Wang *et al.* (2006).

of the data in their implementation, whereas the multivariate ε-SVR approach used in this chapter allows any number of missing values so long as at least one observation is present. Some extreme examples of this are presented in Figure 6.5(c-d). In practice, it would of course be unwise to rely on data modelled on only two or three observations. Fusing the results from this experiment with the other experiments with different cell-cycle synchronization techniques, or by repeating the particular experiment to obtain more data, would allow us to determine if these predictions are accurate.

In terms of a true representation of the data rather than purely mean squared error, we would expect a model based on ε-SVR to create the best possible representation of noisy, inconsistent microarray data — even in comparison to neural network approaches — since the advanced regularization capability of these kernel-based methods allows for a highly-accurate model with a relatively small number of observations. The efficiency of the ε-SVR training algorithm and a sparse model representation allows for much higher dimensionality in the input observations and output targets than neural network approaches (Haykin, 1999; Nabney, 2002), which typically require two weight matrices: the size of the input weight matrix is determined by the number of inputs times the number of hidden nodes, and similarly the output weight matrix is determined by the number of outputs times the number of hidden nodes, as illustrated by the thin lines connecting the nodes in Figure 6.1(a). For an MLP with thousands of inputs, such weight matrices may well exceed available memory limitations. The importance of generalization performance, as a tradeoff to mean squared error, is illustrated in Figure 2.5.

6.2 Data and Methods

In this chapter, we will primarily follow the notation of Gilks *et al.* (2005). We define $\mathbf{D}$ as an $N \times m$ observed data matrix of m gene probes taken at N time points. As is generally the case, here $m \gg N$ since for the *elutriation2* data set, $m = 5038$ and $N = 20$. We refer to the individual elements of $\mathbf{D}$ such that for a particular gene i we have we have observations $\mathbf{x_i}$ and targets y_i as in Equation 2.1, corresponding to row i of $\mathbf{D}$ where in this case $N = \ell$.

In Rustici *et al.* (2004), nine different cell-cycle synchronization techniques were applied to *S. pombe* in comparison to unsynchronized cell cultures, including elutriation to isolate fine cells from heavier cells in the culture, or selective blocking and releasing of

particular proteins known to control the cell cycle through temperature variation. The raw data from these experiments were made available, under accession numbers E-MEXP-54 to -64, through ArrayExpress (`http://www.ebi.ac.uk/arrayexpress`).

Each experiment in this data set shows the normalized, unitless signal ratio of the experimental culture at each time point for each gene, in comparison to that of an unsynchronized, control culture of the same organism, for $> 99.5\%$ (Rustici *et al.*, 2004) of all identified genes in the *S. pombe* genome through several successive hybridizations taken every 15 minutes. A value of one would indicate that the gene has equal levels of expression in the synchronized and unsynchronized cultures, whereas a value higher than one indicates that the synchronized culture exhibits a proportionately higher gene expression level than the unsynchronized control culture. In this study, through a clustering algorithm, 407 genes were identified as periodic and 136 of these were identified as strongly periodic, defined as those whose maximum difference (from peak to trough) was greater than 2. The data in this study was later analyzed using a multivariate linear regression and SVD in Gilks *et al.* (2005).

There are many missing data points in these data sets. For example, in the *elutriation2* data set, 286 of the 5038 genes evaluated on each microarray slide had no data in any of the observed hybridizations, and a further 819 genes had fewer than 75% of the data points available. A total of 16.9% of the observations are missing. It is self-evident that the fewer observations available for any particular gene probe, the less accurate any attempt to model the underlying data will be. However, from the analysis performed in Gilks *et al.* (2005), the *elutriation2* data set appears to exhibit excellent periodicity through nearly two cell cycles, so in this preliminary work, we initially concentrate on this one particular experiment.

6.2.1 Non-linear Multivariate Regression

We first aim to impute these missing observations, and reduce noise levels throughout the data set, using SVM. Since both the input and output for the SVM is the full, time-course experiment, in this case 5038 genes over 20 time points, our SVM model naturally falls into the category of multivariate regression. However, as discussed in Section 6.1.1, here we use a series of univariate regressions: an ε-SVR model for each gene with multidimensional input (the values of the all remaining genes at a specific time point) and unidimensional

output (the value of the target gene for a specific time point). 3515 of the 5038 genes were identified as each having 100% of the available data points available. These were used as the training input for the model, and were normalized such that

$$\mathbf{x}_{ij} \in [-1, +1], \quad i = 1, \ldots, m, \quad j = 1, \ldots, N \tag{6.1}$$

Importantly, the time vector itself $t \in \{0, 15, 30, \ldots, 285\}$ was *not* provided as an input vector for this model; only the gene expression data was used. This allows the SVM model to seek a generalized solution in 3515-dimensional input space purely on the basis of exploring the relationships between individual genes. This removes any assumption of the time-variant nature of the gene expression data, and allows us to impute missing observations even if a significant number of observations is missing: for example, see Figure 6.5 (b), in which data for the second cycle is imputed based on observations made during the first cycle. Naturally, the target gene itself is also excluded from the training data for each model.

An important consideration in the practical application of SVM is the determination of appropriate values for the free parameters, or hyperparameters, of the SVM during training: we wish to determine optimum values for the ε-tube width, RBF γ-radius and C cost free parameters. Here we use the heuristic proposed in Chapter 3, adapted to ε-SVR by adjusting the scoring function as follows.

As in Chapter 5, the goal of this heuristic is to balance generalization performance with model complexity. Through the course of the three-dimensional, stochastic parameter search, we minimize the cost functional $\mathscr{E}(\varepsilon, \gamma, C)$ defined as Equation 3.1. Here, $\mathscr{E}_s(\varepsilon, \gamma, C)$ is the 10-fold cross-validation mean squared error (MSE) resulting from a model trained using parameters defined by this point in three-dimensional parameter space, normalized by the inverse of the standard deviation of the target observations $\sigma(y_i)$, as

$$\mathscr{E}_s = \frac{1}{N\sigma(y_i)} \sum_{j=1}^{N} \left| y_{ij} - f(\mathbf{x}_{ij}) \right|^2 \tag{6.2}$$

where $f(\mathbf{x}_{ij})$ is the predicted value of y_{ij} from the regression model, as Equation 2.36. Normalizing in this manner allows those genes with small expression to more significantly affect the mean squared error in comparison to those with relatively high expression. $\mathscr{E}_c(\varepsilon, \gamma, C)$ is a model-complexity penalty defined by

$$\mathscr{E}_c = \left(\frac{n_{sv}}{\ell} \right)^{\Gamma} \tag{6.3}$$

where $n_{sv}(\varepsilon, \gamma, C)$ is the number of support vectors in the resulting model representation and ℓ is the total number of training observations. The importance of including a model complexity measure in regression problems is illustrated in Figure 2.5.

The regularization parameter λ balances the tradeoff between $\mathscr{E}_s$ and $\mathscr{E}_c$, and the square $\Gamma = 2$ introduces a non-linearity to more sharply penalize those models which obtain a low mean squared error at the expense of high model complexity. We found that a value of $\lambda = 10$ brought the two terms to the same orders of magnitude, giving roughly equal weight to each.

To balance computational complexity with model accuracy, rather than calculating the optimum free parameters for each of the 5038 models, we minimize the sum of $\mathscr{E}$ taken over ten representative genes at each evaluated point in three-dimensional free parameter space. Each of these genes were identified as periodic by the clustering methodology in Rustici *et al.* (2004). Some had small expression throughout the time course of the data, others had large expression, indicating the need for normalization in the MSE term as shown above. The selected genes were *slp1*, *cdc20*, *h4.2:hhf2*, *sly1*, *h3.1:hht1*, *h3.3:hht3*, *h4.3:hhf3*, *h3.2:hht2*, *h4.1:hhf1* and *bgs4*. The optimum parameters found from the analysis of these ten genes were then used to train the ε-SVR for all genes.

The simulated annealing heuristic was employed with a moderate cooling schedule, such that the objective function was evaluated at a total of 4590 points in parameter space. This is contrast to the classification approach in Chapter 4, which searched a two-dimensional parameter space for the RBF γ and C regularization parameters for the purpose of binary classification, using either a fast-cooling schedule with 440 evaluations or a slow-cooling schedule with 6880 evaluations.

We define the matrix of observations cleaned by this multivariate regression as $\mathbf{S}$, of the same dimensions as $\mathbf{D}$.

6.2.2 Periodic Additive Model

The output from this multivariate analysis is then used as an input to a periodic, additive model. For each gene i, we define the true underlying signal to be the time-variant function $C_i(t)$. The estimate of this signal obtained through the periodic model is defined to be $\hat{C}_i(t)$. The difference between these signals is then the residual $\xi_i(t)$:

$$C_i(t) = \hat{C}_i(t) + \xi_i(t) \tag{6.4}$$

We propose that the model of each gene contain both a periodic component and a linear component:

$$\hat{C}_i(t) = \rho_i \sin\left(\frac{2\pi t}{\Lambda} + \phi_i\right) + (t\alpha_i + \beta_i) \tag{6.5}$$

where the parameters α_i and β_i define the linear component of the model for each gene i, the magnitude and phase of the periodic component is defined by ρ_i and ϕ_i for each gene i, and the cell-cycle length is expressed by Λ as a constant for all genes within the particular experiment (not to be confused with the λ regularization parameter in Equation 3.1).

In order to determine the model parameters for each gene, we use a non-linear, least-squares curve-fitting procedure provided by the MATLAB Curve Fitting Toolbox (Mathworks). Specifically, we use the robust implementation of the trust-region reflective Newtonian algorithm, which allows us to impose logical bounds on each model parameter. To shake the optimization procedure from local minima, two starting points were used with opposite phase $\phi = \{0, \pi\}$. A purely linear model was also applied. Of these three resulting models, we take that with the lowest mean squared error. For additional detail on the specific implementation of similar curve-fitting procedures, see for example Press *et al.* (1992, § 15.5–15.7).

We first apply the proposed periodic model (above) to each gene for a particular set of hybridizations, using non-linear, least-squares curve fitting, to find the most likely model parameters for each gene allowing Λ to vary independently for each gene. For this step, we consider only the 407 genes identified as periodic by Rustici *et al.* (2004). Some of these genes were identified as linear by our model, and these were naturally excluded. The median Λ of the remaining genes was taken to be the "true" cell-cycle length.

We then reapply the curve-fitting procedure for all genes, taking this Λ value as fixed, to generate the final periodic model for each gene.

6.2.3 Goodness-of-Fit Statistics

As a goodness-of-fit statistic, here we normalize the commonly-used mean squared error by the variance (Wang *et al.*, 2006), to define the normalized root mean squared error (NRMSE) of the observations compared to the SVM model as

$$\mathcal{G}F_{SVM}(\mathbf{D}_i, \mathbf{S}_i) = \sqrt{\frac{1}{N\sigma^2(\mathbf{D}_i)} \sum_{j=1}^{N} |d_{ij} - s_{ij}|^2} \tag{6.6}$$

Table 6.1: Comparing mean squared error (MSE) $\times 1000$ resulting from periodic expression models for several genes, from *elutriation2* data set. The data from gene *C222.06* was used to find optimum ε-SVR model parameters for this univariate analysis.

Method	*slp1*	*hhf2*	*bgs4*	*cdc20*
Polynomial Regression (Degree 6)	35.0	42.4	11.0	17.3
Univariate Support Vector Regression†	3.3	13.6	4.3	20.9
Linear Regression	385.1	658.1	37.1	66.5
Periodic Model	24.9	53.6	14.3	35.4

where d_{ij} is the j-th observation of the i-th gene, s_{ij} is the j-th prediction of the i-th gene by the ε-SVR model, and $\sigma^2(\mathbf{D}_i)$ is the variance of the observations for gene i.

Similarly, to determine how well the periodic, additive model fits the SVM model, we define the NRMSE of the SVM model compared to the periodic model as

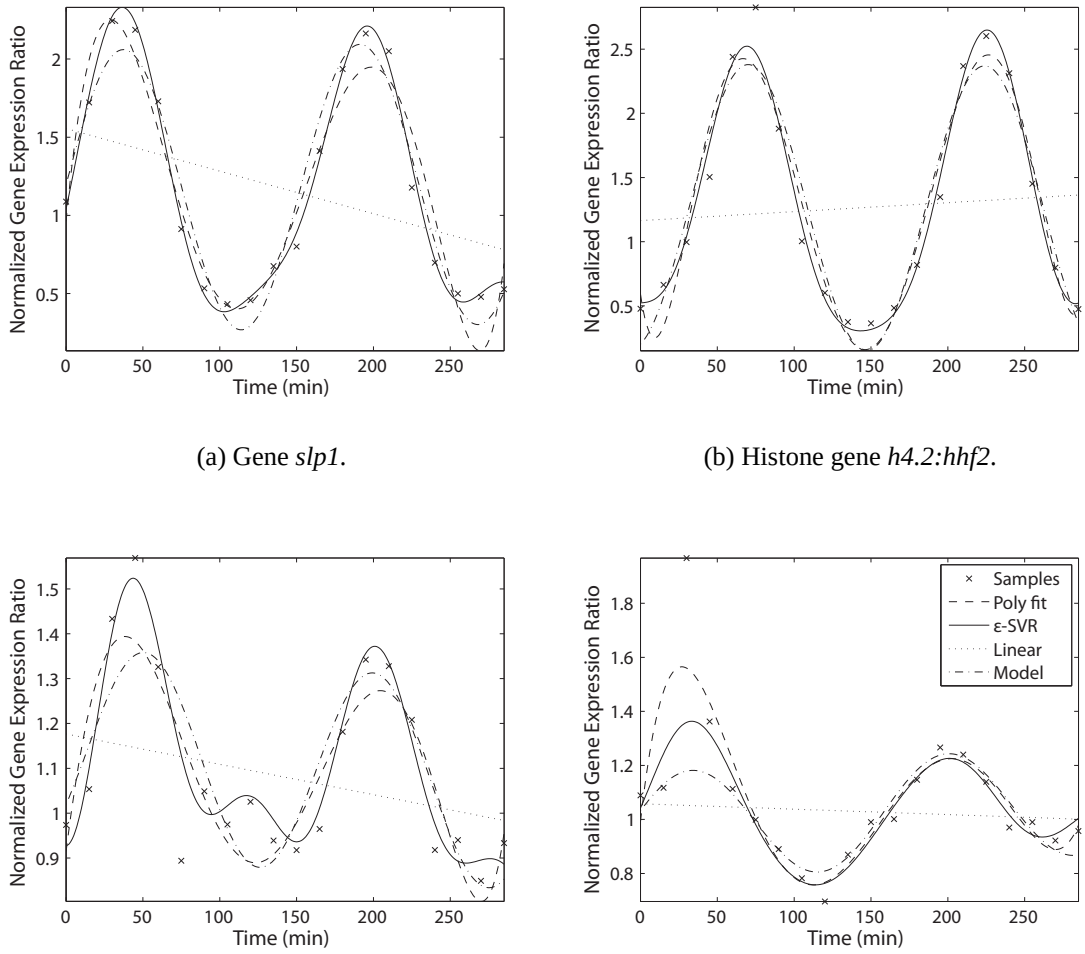
$$\mathcal{G}F_{Model}(\mathbf{S}_i, \hat{C}_i) = \sqrt{\frac{1}{N\sigma^2(\mathbf{S}_i)} \sum_{j=1}^{N} |s_{ij} - \hat{c}_{ij}|^2} \tag{6.7}$$

where $\hat{c}_{ij}$ is the periodic model's expected value for the j-th observation of the i-th gene. and $\sigma^2(\mathbf{S}_i)$ is the variance of the ε-SVR model's predictions for gene i.

6.3 Results

We first obtained some preliminary results based on applying a univariate ε-SVR (considering only the time vector as input) and the periodic model to each gene independently in the *elutriation2* data set. Figure 6.2 shows a comparison of the observed data for four specific genes, identified as typical examples in Gilks *et al.* (2005), to four different signal estimation techniques: linear regression, least-squares polynomial regression with six degrees of freedom, ε-SVR and the proposed additive, periodic model. We found that although the ε-SVR curves precisely approximate all four genes with the least mean squared error, the proposed periodic model also aligns well these four genes and appears to be a good fit in all four cases. Table 6.1 shows the mean squared error for each of the four genes obtained from each estimation technique.

We then extend this technique to multivariate ε-SVR, and estimate the cell-cycle length for the *elutriation2* data set. Only the 407 genes found to be periodic using a clustering algorithm in Rustici *et al.* (2004) were included. We find that the most likely cell-cycle



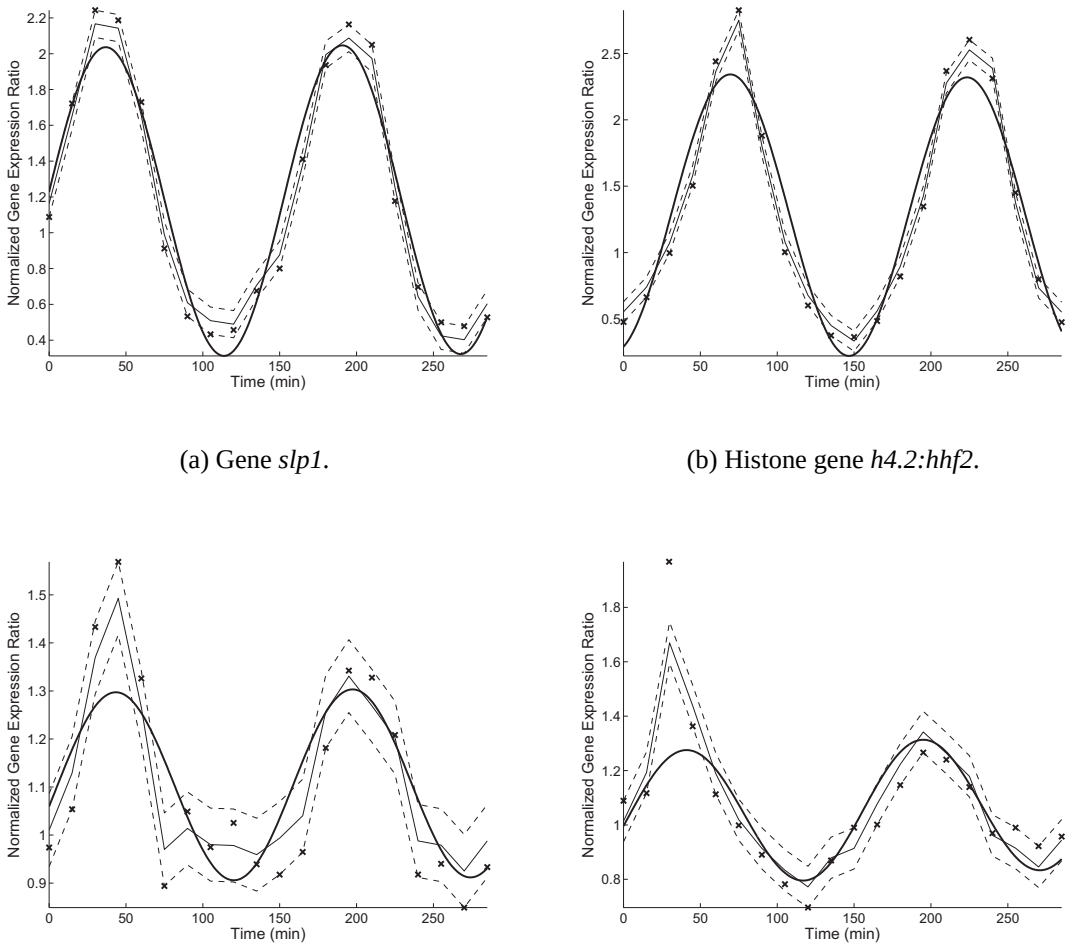
(a) Gene *slp1*.

(b) Histone gene *h4.2:hhf2*.

(c) Gene *bgs4*.

(d) Gene *cdc20*.

Figure 6.2: Modelling periodic gene expression levels for several genes as Gilks *et al.* (2005), comparing sixth-degree polynomial regression, ε-SVR and linear regression to the periodic model proposed in this chapter, from the *elutriation2* data set. The legend for all four figures is as (d) (not shown in other figures for clarity). These results agree with the findings in Gilks *et al.* (2005), with *slp1* and *hhf2* exhibiting highly periodic behaviour but finding much less periodicity with *bgs4* and *cdc20*. However, although the ε-SVR model finds some evidence of a second peak in *bgs4*, found to be slightly biperiodic in Gilks *et al.* (2005), the other models ignore this secondary peak as experimental noise. Also, *cdc20* was thought to be non-periodic in Gilks *et al.* (2005), however these results do find some evidence of weak cell-cycle periodicity.



(a) Gene *slp1*.

(b) Histone gene *h4.2:hhf2*.

(c) Gene *bgs4*.

(d) Gene *cdc20*.

Figure 6.3: Modelling periodic gene expression levels for several genes as Gilks *et al.* (2005), comparing multivariate ε-SVR (thin line) within ε noise-insensitivity tube (thin dashed lines) and the periodic model with cell-cycle length fixed at $\Lambda = 153.9$ minutes (thick line) with the original observed data ($\times$). The ε-SVR appears more jagged than in the univariate comparison (Figure 6.2) since with the multivariate model only those points defined by the remaining genes may be used, whereas in the univariate model (which considers only the time vector as input) the curve may be evaluated at much finer resolution. It is important to note that in the multivariate case, the time vector itself is *not* used as an input: only those 3515 genes which have 100% of the data points (and which are not the target gene!) are used.

Table 6.2: Comparing model parameters found from proposed periodic maximum likelihood model for several genes, from the *elutriation2* data set cleaned through multivariate ε-SVR, with a fixed cell-cycle length of $\Lambda = 153.9$ minutes.

Model Parameter	*slp1*	*hhf2*	*bgs4*	*cdc20*
ϕ_i : Phase (radians)	0.068	-1.269	-0.198	-0.080
ρ_i : Periodic Amplitude	0.865	1.055	0.197	0.249
α_i : Linear Decay ($\times 1000$)	0.064	-0.144	0.039	0.246
β_i : Linear Offset	1.170	1.297	1.098	1.016
$\mathcal{GF}_{SVM}$: NRMSE of SVM Model	0.100	0.089	0.277	0.335
$\mathcal{GF}_{Model}$: NRMSE of Periodic Model	0.202	0.269	0.464	0.469

length for this data set is 153.9 minutes, indicating that the 285 minutes in this experiment cover $\sim$1.85 cell-cycles. This appears to match well with the result found in Gilks *et al.* (2005) obtained through SVD. Figure 6.4(a) shows a histogram of the most likely cell-cycle length obtained from the model for each gene in this data set: the estimated cell-cycle length is taken as the median value. Figure 6.4(c–d) show histograms of the NRMSE of the models. A museum of interesting genes, showing examples of what is possible with this multivariate approach, is presented in Figure 6.5.

Finally, we run the curve-fitting procedure again, holding the cell-cycle length $\Lambda = 153.9$ minutes. The results for the same four genes are compared in Figure 6.3. Table 6.2 shows the periodic model parameters obtained for each of these four genes. Models for the full set of 407 genes identified as periodic by Rustici *et al.* (2004) are available online at the author's web site, for both fixed and unfixed Λ.

In contrast to the conclusion in Rustici *et al.* (2004), in which a clustering algorithm determined that there were 407 periodic genes and 136 were strongly expressed, our analysis shows that 1252 of the 5038 genes show some statistically-significant levels of periodic activity, higher than the empirical noise level ε within the *elutriation2* experiment. 332 of these show periodic activity with a magnitude twice as high or greater than the estimated noise level. The noise level, which the multivariate ε-SVR model determines through the ε-tube width free parameter, was used as a significance measure in comparison to the periodic amplitude ρ_i determined by the periodic model's curve-fitting algorithm, in order to remove any potentially biased or arbitrary assumption. The complete list of 332 strongly-periodic genes may also be found on the author's web site.

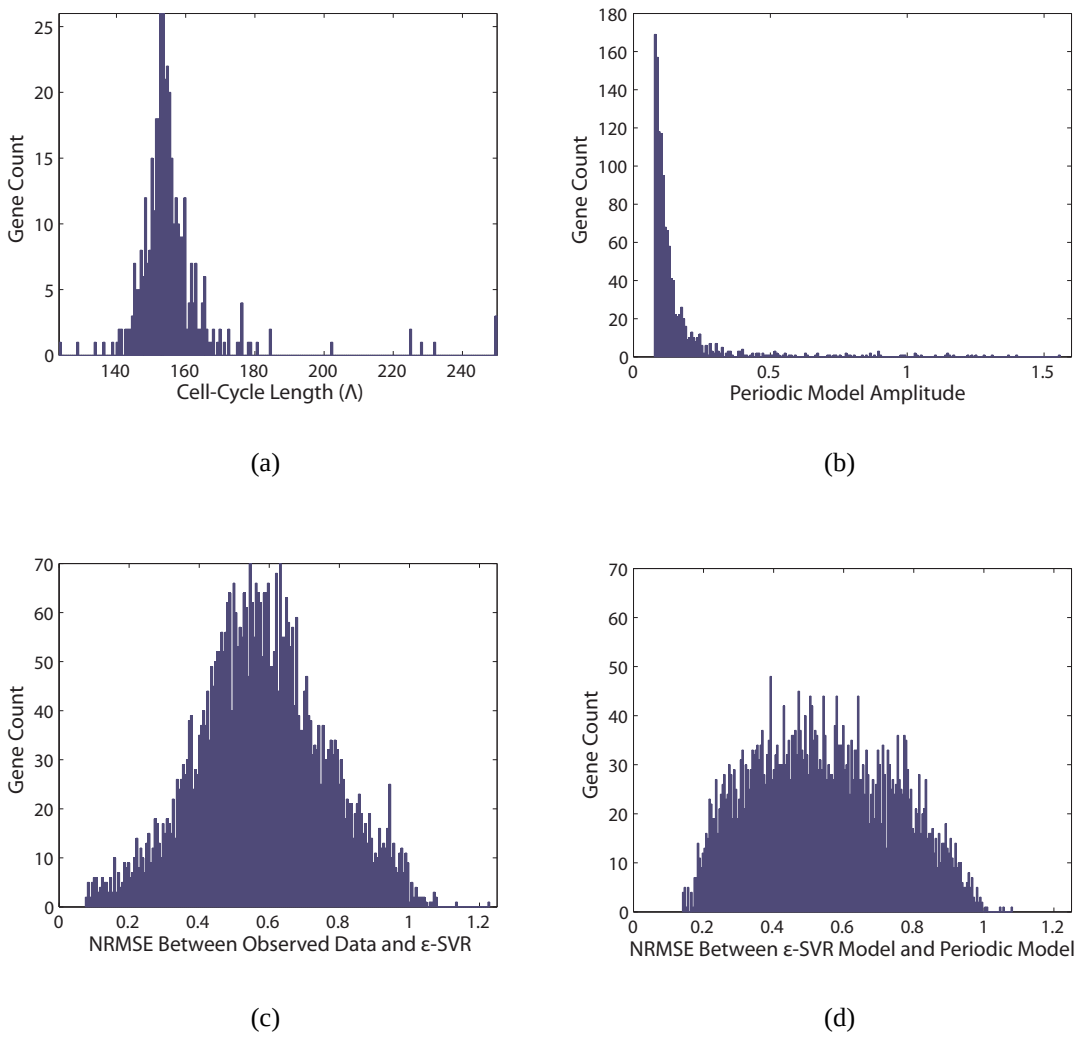
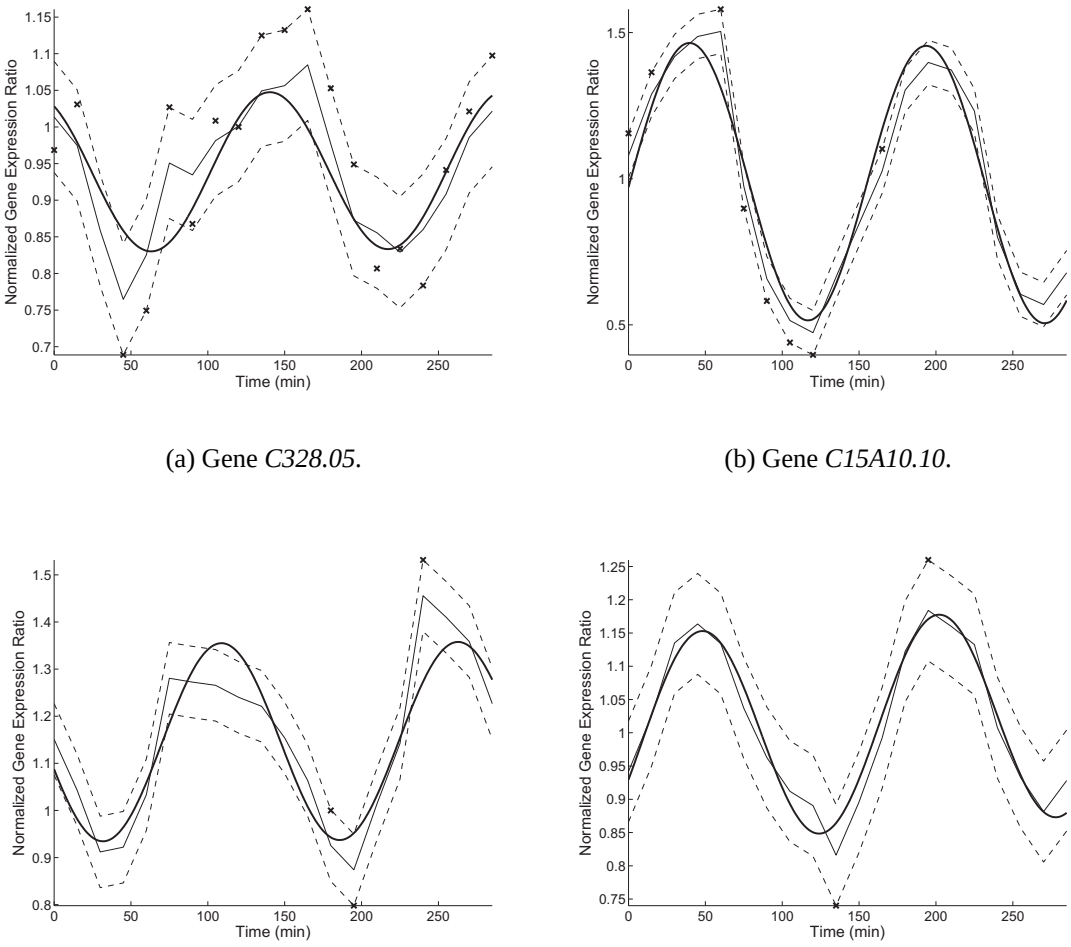


Figure 6.4: (a) Distribution of cell-cycle length from the *elutriation2* data set for each gene, found by the periodic model using the cleaned multivariate ε-SVR data set as input. The 407 genes identified as periodic by Rustici *et al.* (2004) were used to find the median cell-cycle length of 153.9 minutes. The sharp distribution, with few outliers, allows high confidence in the resulting cell-cycle estimate. (b) Distribution of the amplitude of the periodic component of the additive model, for those genes with a statistically significant periodic amplitude, that is, greater than the empirical noise estimate ε determined by the heuristic. The majority of the genes exhibit low periodicity, but a significant number are strongly expressed with amplitude $\rho > 2\,\varepsilon = 0.152$. (c) Distribution of the $\mathcal{G}F_{SVM}$ NRMSE statistic for all genes, between the original data and the data cleaned by multivariate ε-SVR. The median NRMSE is 0.5725. (d) Distribution of the $\mathcal{G}F_{Model}$ NRMSE statistic for all genes, between the data cleaned by multivariate ε-SVR and the predictions of the periodic model. The median NRMSE is 0.5322. Note the difference in the overall shape of the distributions in (c) and (d), although both have similar median values.



(a) Gene *C328.05*.

(b) Gene *C15A10.10*.

(c) Gene *C1105.13c*.

(d) Gene *C8E4.06*.

Figure 6.5: A museum of interesting genes found through the course of this analysis, showing the potential power of the multivariate ε-SVR algorithm. Since one of the strengths of SVM is to generalize the solution to a problem from a low number of observed samples, we would expect superior imputation performance for a data set such as this. It is important to note that the time vector is not used as an input for the ε-SVR: when imputing observations for a target gene, only that gene's relations to the remaining genes are considered. (a) Imputation of a single observation (at $t = 30$ min) with a high level of noise and small expression levels. (b) Data for the first cycle is present, but the second cycle is entirely imputed. KNN, for example, would simply assume a flat curve for the second cycle, but the relationships between genes in this multivariate approach allows the true expression of this gene to be seen. (c) Imputing periodicity from only three observations, and (d) from only two observations: these are possible since the observations are not closely spaced, so that the differences in other genes allow for prediction even in these seemingly impossible scenarios.

As a visualization of the result, we then compare the magnitude and phase model parameters of the periodic component of the model obtained for each gene to determine the relative periodicity in relation to cell-cycle length for each active gene. This results in a plot similar to the "peppered fried egg plot" in Gilks *et al.* (2005, Figure 4) which was obtained from the first two eigenvectors of an SVD for each gene. Rather than reproduce the actual "egg yolk" (loess curve of radius of gene expressions through the cell cycle) and "egg white" (loess curve of the density of genes through the cell cycle) individually, we combine these into a single curve as an indication of the average total gene expression through the cell cycle, and fit the curve using univariate ε-SVR rather than a loess curve: since the free parameters for ε-SVR are obtained through the same simulated annealing heuristic in Chapter 3, no smoothing assumption is necessary — a "span," or square window, is needed for the loess curve method to determine a breadth of values to be used for smoothing. This visualization is presented in Figure 6.6 for the 407 genes identified as periodic by Rustici *et al.* (2004). In this plot, all genes in this subset that are identified as periodic by the model are shown, regardless of the NRMSE statistic.

However, can we obtain the same result by applying logical thresholds to the results from the full set of all 5038 genes? In Figure 6.7, we show the rotational plot of the cell-cycle for all genes identified as significantly periodic by the additive model, — those with a periodic amplitude $\rho_i > 2\,\varepsilon$ — and for which the NRMSE statistics are below $\mathcal{G}F_{SVM} < 0.4$ and $\mathcal{G}F_{Model} < 0.6$. Those genes with a large decay constant $\alpha_i \times 1000 > 3$ are also removed from the plot, as a large decay constant indicates that the curve-fitting procedure considers the linear component of the additive model to be too highly significant and was not able to properly converge. It seems likely in these cases that the slowly changing gene expression levels described by the linear component are due to some unobserved, external factor — such as changing ambient light levels, for example — rather than sensor drift in the microarray reader, which typically require daily recalibration (Agilent, 2005). This resulted in the selection of a total of 274 genes, shown in Figure 6.7.

6.4 Discussion

Comparing the visualization of the 407 periodic genes in Figure 6.6 to those of Gilks *et al.* (2005)(Figure 4), it is clear that the combined multivariate ε-SVR and non-linear curve-fitting approach has been successfully applied to this data set. There are many similarities

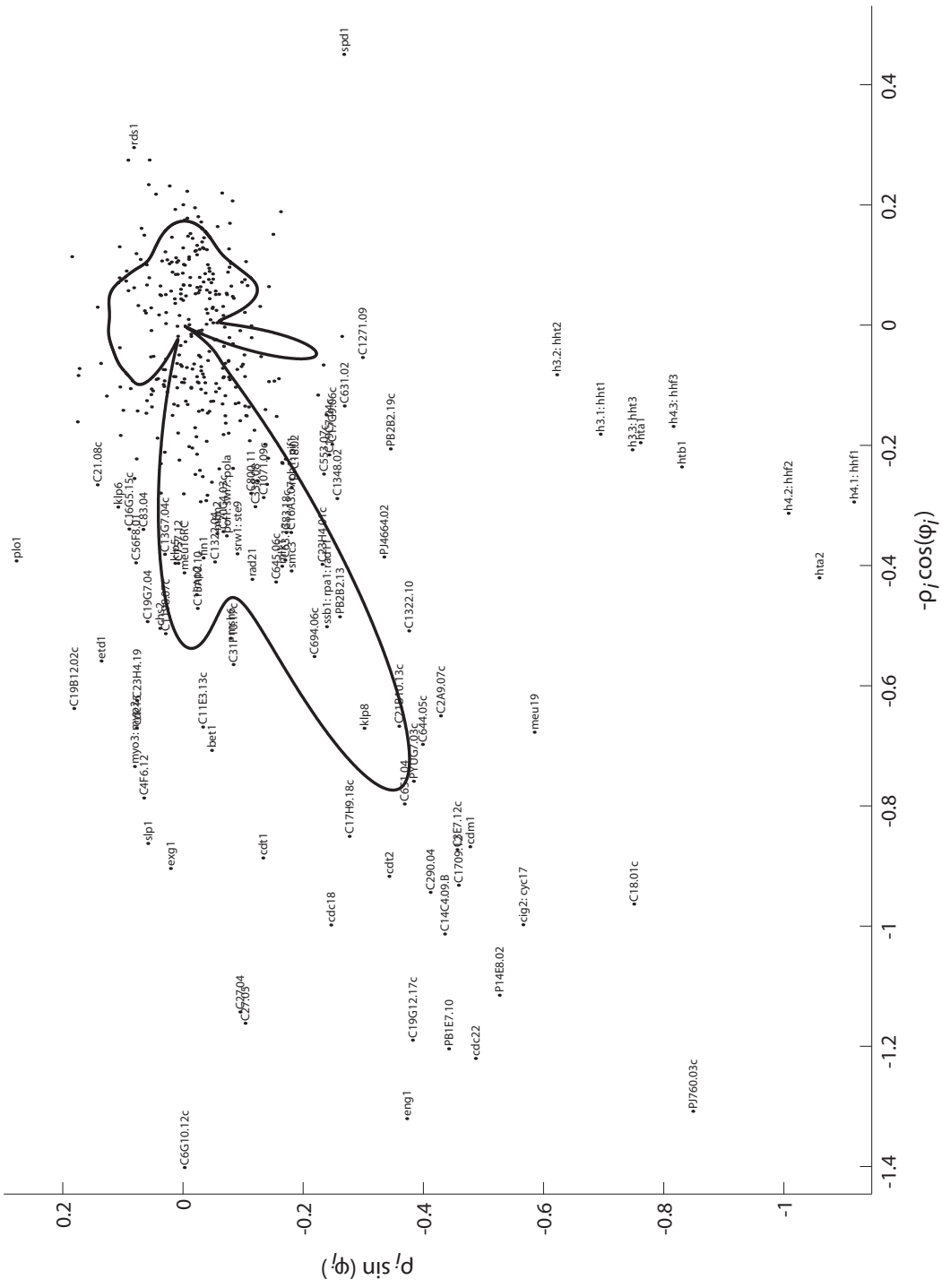


Figure 6.6: A rotational plot of the periodic magnitude and phase, as captured by the periodic additive model created by the data set cleaned by multivariate ε-SVR, of each of the 407 genes identified as periodic by Rustici *et al.* (2004). The dots indicate the magnitude and phase of the periodic expression of individual genes, with some extrema labelled. The curve represents the average total gene expression throughout the cycle, smoothed by a univariate ε-SVR based on a histogram with 100 bins.

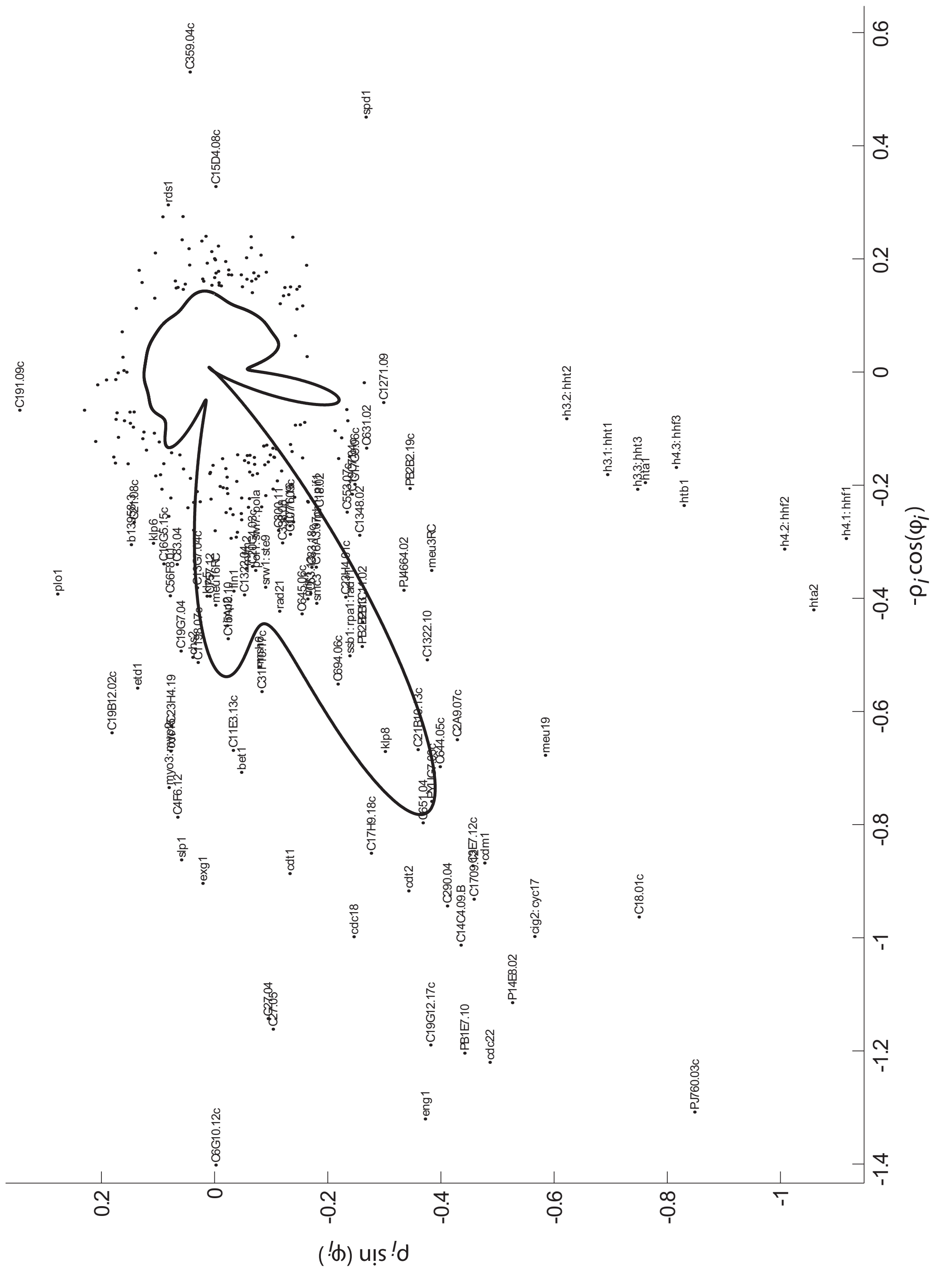


Figure 6.7: A rotational plot as Figure 6.6, but including only the 274 genes determined to be strongly periodic by our analysis. Although this plot contains fewer genes, the average total gene expression curve strongly resembles that of Figure 6.6, indicating that differences between the plots will most likely be found in genes with small expression.

between these plots, for example we see nine of the histone genes (those starting with *h*) occurring at the same point in the cycle in both plots, with corresponding bumps in the gene expression curve. The *slp1*, *plo1* and *spd1* genes appear to match as well.

There are also some differences, most likely since we only consider the *elutriation2* experimental data in this analysis, rather than the fusion of all nine experiments. For example, the *rds1* gene appears to be active earlier in the cycle. Several genes have stronger expression in this plot, for example the *meu19*, *exg1* and *etd1* genes appear in Figure 6.6 but are not labelled as outliers in Gilks *et al.* (2005)(Figure 4). There also appears to be a greater deviation between those genes with small expression and those with large expression. This may be partly due to the loss of information in plotting only the first two dimensions of the SVD, which in Gilks *et al.* (2005) contained 83% of the SVD information.

In the visualization plot in Figure 6.7, of all genes found to be strongly periodic by our model, we see more examples of genes with strong enough expression to be labelled as outliers. For example, the *C191.09c*, *C15D4.08c* and *C359.04c* genes appear to be strongly periodic. Fusion of this data with the remaining synchronization experiments could determine if these are truly additional periodic that were not originally discovered in the analyses of Gilks *et al.* (2005); Rustici *et al.* (2004). It is clear that overall, however, the genes identified in these two plots match fairly closely: the curve describing the average total gene expression throughout the cell-cycle is nearly identical in both plots. The thresholds for selecting strongly periodic genes were selected arbitrarily, based on the empirical distributions of each parameter, however it would be better to set these thresholds based on sound biological reasoning: for the moment, we leave this as future work. The average total gene expression curve in Figure 6.7 strongly resembles that of Figure 6.6, indicating that differences between the plots will most likely be found in genes with small expression.

This data fusion would be the next logical step in this analysis, and could be implemented as follows. We would first perform the same cleaning and data imputation procedure for each of the nine cell-cycle synchronization experiments performed by Rustici *et al.* (2004). From the analysis in Gilks *et al.* (2005), we know that the cell-cycle length and phase both will significantly vary between experiments due to the differences in synchronization procedures: the use of a periodic model allows us to estimate the cell-cycle length for each experiment individually. The phase for each experiment could then be estimated using a representative set of genes, such as the nine histone genes, using a best-fit optimization. The resulting cell-cycle length and phase for each experiment could then be

used to create a time vector for each observation, for each gene, for each experiment, using an idealized cell-cycle stretching from 0 to $2\,\pi$. Univariate ε-SVR could then be applied to each gene individually to determine the most likely curve within this idealized cycle, and our curve fitting procedure could then determine the magnitude and phase of each gene as performed above. The results of this non-linear fusion might be interesting to compare with the above data, for example to determine how accurately the single *elutriation2* data set shows the periodic expression of each gene compared to the fusion of all sets. However, unfortunately, this is beyond the scope of the analysis performed in this chapter. We therefore leave this for future work.

Another avenue for future work may be to determine the sensitivity of each gene to the values of the remaining genes using a modified Monte Carlo approach. For example, the formation of the SVM allows us to set all input genes to zero, the average expression level of the normalized input genes, then vary the values of each independently to determine which genes affect the outcome of the regression, and to quantify the extent of this effect and rank the input genes accordingly. This in turn would show us which groups of genes are closely related, and could perhaps lead to an advanced clustering technique. For example, one could progressively form a dendrogram based solely on these sensitivity relationships.

The full set of genes identified as strongly periodic in this way can be found at `http://www.cs.dal.ca/~tt`, with rotational plots for both the 407 periodic genes identified by Rustici *et al.* (2004) and the 274 identified here, including both the average total gene expression curve and the separated radius and density curves for comparison as was done in Gilks *et al.* (2005). A simple web interface allows users to browse through the original observations, cleaned data set and generated models for the 407 periodic genes used in these experiments, for both fixed and unfixed cell-cycle length, and to see the genes identified here as strongly periodic. These comparisons are available in both PNG and PDF formats, suitable for on-screen viewing and printing respectively.

From our analysis of the full *elutriation2* data set visualized in Figure 6.7, we conclude that our combined technique of data imputation and noise reduction by multivariate ε-SVR, and non-linear curve-fitting by a periodic, additive model, is well-suited for the identification of periodic genes from DNA microarray data.

Chapter 7

Input Variable Relevance

We have established that the heuristic proposed in this thesis performs well for several real-world classification and regression problems. In this section, we will develop a methodology to determine which input variables are the most relevant in terms of classification accuracy, using the retinal electrophysiology data set from Chapter 4 as a test case. We will show several standard variable-relevance measures, including the Fisher Ratio, Pearson Correlation Coefficients, the Kolmogorov-Smirnov Test, and Mutual Information. We will also introduce two measures using SVM: the first a linear measure, based on taking each variable independently as input to a linear SVM, and the second a Monte Carlo-style approach to the sensitivity of each input variable considering the true non-linear SVM classification model. We find that the sensitivity approach works well for this data set, and compare the two variations of this sensitivity, smoothed over a small square window, with the original waveforms.

7.1 Introduction

Feature selection for a retinal electrophysiological waveform is useful for medical researchers, for example, who may wish to determine which parts of a particular waveform differ the most between two classes of subjects for a particular experiment. Such analysis may help with medical diagnosis in patients, or aid in understanding the physical mechanisms and biological processes involved. It is also useful from a computational efficiency point of view: if we can demonstrate experimentally that certain variables are irrelevant or even detrimental to cross-validation classification accuracy, those variables may be safely eliminated from the input vector, thereby increasing computational efficiency and potentially increasing the accuracy of the final classifier.

Modern problem domains may employ many thousands of input variables (Guyon and Elisseeff, 2003), leading to the so-called *curse of dimensionality*: a term originally coined by Richard Bellman, the father of modern dynamic programming, to describe the rapid increase in problem complexity as new dimensions are added (Bellman, 1961). The computational efficiency and sparse representation in SVM encourages such large input spaces

(Vapnik, 1995), allowing the SVM to "defy the curse of dimensionality" (Guyon and Elisseeff, 2003). Indeed the feature space itself may have an infinite number of dimensions, such as with the RBF kernel (Vapnik, 1995). However, it has been demonstrated experimentally that removing irrelevant features will improve SVM classification accuracy (Weston *et al.*, 2000).

Variable selection methods are typically divided into two main categories (Guyon and Elisseeff, 2003; Weston *et al.*, 2000): *filter* methods, or variable ranking methods, as a pre-processing step to analyze each variable independently and determine some measure of its relevance (Weston *et al.*, 2000), and *wrapper* methods, to evaluate subsets of several input variables to find a subset that optimizes some functional chosen *a priori* (Guyon and Elisseeff, 2003), such as cross-validation accuracy or mean squared error.

Some common filter methods for assessing variable relevance are the Fisher Ratio, which measures the class-separability of each variable (Chapelle *et al.*, 2002; Weston *et al.*, 2000); Pearson Correlation Coefficients, which measure the linear correlation between each variable and the classification target (Press *et al.*, 1992); and the Kolmogorov-Smirnov Test, which determines the maximum difference between the cumulative distributions of each class, estimated empirically from the training observations (Press *et al.*, 1992; Weston *et al.*, 2000). Mutual Information (see for example Carlson, 1986) is another filter method, based on empirical estimation of the joint probability distribution between the inputs and outputs. One problem with any filter method when applied to SVM, is that the SVM kernel's mapping of the inputs into feature space may create situations where some input variables are irrelevant to the SVM predictor, but in feature space these variables combine to create one or many very relevant features (Guyon and Elisseeff, 2003).

Wrapper methods are less commonly used, as they typically increase the computational burden as each possible subset is evaluated, becoming a combinatorial problem (Weston *et al.*, 2000). Wrapper methods generally come in two specific categories (Guyon and Elisseeff, 2003): *forward selection*, in which a variables are progressively added to an initially empty subset, and *backwards elimination*, in which variables are progressively eliminated from the full set of inputs. Genetic algorithms (see for example Mitchell, 1996) have been applied to the problem of feature selection for automated polyp detection in colonography using SVM with forward selection (Miller *et al.*, 2003). However, in our own experiments with such genetic algorithms for the retinal electrophysiology data set, we found that the

resulting SVM tended to capitalize on a small number of highly-separable input dimensions to quickly achieve high accuracy, rather than improving generalization performance. In Weston *et al.* (2000), a wrapper method is proposed which considers many possible subsets of $n < \ell$ input variables, but uses a gradient descent approach to improve computational efficiency while minimizing bounds on the leave-one-out generalization error.

In Chapelle *et al.* (2002), a novel variable selection method was proposed based on adjusting the RBF kernel parameters for each observation, by modifying the kernel as (Chapelle *et al.*, 2002)

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(- \sum_{i=1}^{d} \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\,\sigma_i^2} \right) \tag{7.1}$$

where d is the number of input dimensions and σ_i^2 is the variance of the Gaussian kernel for a particular dimension i, rather than using a single parameter $\gamma = 1/2\sigma^2$ to give all dimensions equal weight, as we do here. Chapelle *et al.* (2002) found that variable selection based on the Fisher and Pearson tests did little to improve performance for non-linear data sets, but had more success with the Kolmogorov-Smirnov Test, which found comparable accuracy to their modified kernel.

In Rueda *et al.* (2004), the idea of *variable sensitivity* was applied to the economic problem of currency crises aftermaths. In this article, sensitivity is defined as the maximum change in the magnitude of an SVM's prediction when the value of a particular input variable is varied over its allowable range, while all other input variables are held constant at their mean value. SVM models were trained using the pattern search algorithm in Momma and Bennett (2002) to optimize parameters, and an iterative Monte Carlo approach was used to measure the sensitivity of the SVM model to each input variable. Here, we will simplify a similar approach in Section 7.3, so that only a single training session is required, using the heuristic from Chapter 3 without modification. Some further variations of such sensitivity analyses are summarized in Guyon and Elisseeff (2003).

For SVM, the issue of variable selection for the sake of reducing computational complexity may well be irrelevant, since the SVM training algorithm is efficient even in cases with hundreds of thousands of input variables (Guyon and Elisseeff, 2003). One might argue that since the SVM training algorithm effectively performs its own variable selection in determining a feature space, which may have billions of features or may indeed have infinite dimensionality (Vapnik, 1995), the number of original input variables is no longer

an important issue in terms of computational complexity. Similarly, it might be argued that the removal of multiple redundant variables (see for example Yu and Liu, 2003) may also be considered unnecessary: in a finite data set, especially with a low number of observed samples, measures of redundancy may not be accurate.

However, in Rueda *et al.* (2004), a method was proposed for input variable selection with SVM by using backwards elimination in an iterative approach: a random variable was inserted at each step in the iteration, and those input variables with a sensitivity less that the random variable were eliminated. When all the remaining inputs variables have a sensitivity greater than that of the random input variable, the process is complete. An iterative approach such as this would be necessary for SVM, since if even a single variable is removed, the sensitivities of the new model to each of the remaining inputs are likely to be quite different than those of the first model, as the variables are combined in an unknown, but different, feature space. This creates a *model dependent* method for variable selection.

The similar problem of dimensionality reduction may use methods such as Principal Components Analysis (Wolf and Bileschi, 2005), or Independent Components Analysis (Trappenberg *et al.*, 2006), to combine the inputs into a new set of dimensions which contain the same information. The usefulness of these approaches when applied to SVM may again be debated, since the SVM kernel function performs a similar function.

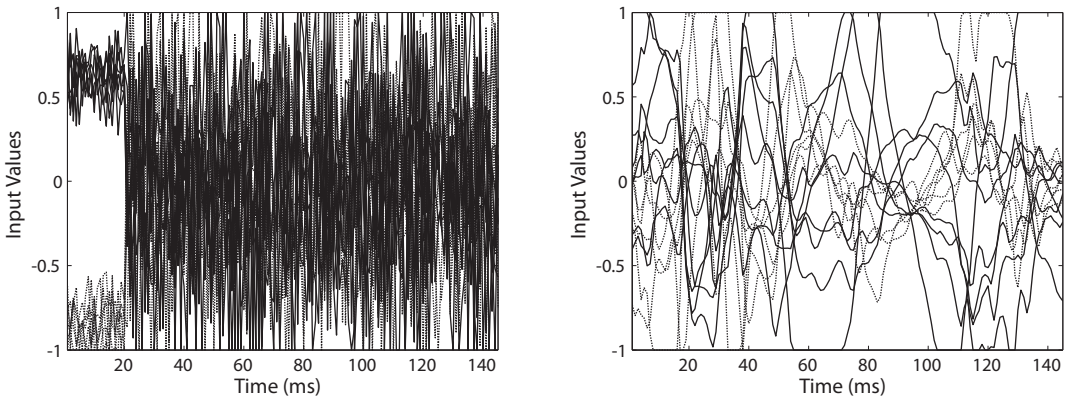
7.2 Variable Relevance Measures

First, we will examine several commonly-used filter methods for variable selection. We will consider each input variable, that is each point on the waveform, independently. Each measure will be examined for the retinal electrophysiology data set, as an example of a data set with high dimensionality but a low number of training observations, and to a toy data set of the same size for comparison.

The first 20 variables in the toy data set are linearly separable, but the rest are entirely Gaussian noise, such that

$$\mathbf{x}_i = \begin{cases} 5\,y_i + 0.1\,\eta_i & \text{if} \quad i \leq 20 \\ 0.1\,\eta_i & \text{otherwise.} \end{cases} \tag{7.2}$$

where η_i are random Gaussian variables, with a mean of zero and a variance of one. To match the retinal electrophysiology data set, six positive and eight negative samples were



(a) Toy data set.

(b) Electroretinography data set.

Figure 7.1: When the inputs are independently scaled to approximate i.i.d. data according to Equation 7.3, the first 20 variables of the toy data set are clearly separable with no overlap, whereas the remaining variables are simply scaled Gaussian noise. No obvious class separability is present in the retinal electrophysiology data set, for which significant overlap between the classes appears to be present in each dimension. Both data sets contain six positive observations (solid lines) and eight negative observations (dotted lines).

generated from this distribution. The toy data set was generated once, and the same set was used for each of the following measures.

Both the toy data set and the retinal electrophysiology data set were scaled by the mean and magnitude of each variable taken independently, as in Chapter 4, such that

$$\mathbf{x}_i \in [-1, +1], \quad i = 1, \ldots, \ell \tag{7.3}$$

and all variables have zero mean, to approximate i.i.d. data. This avoids any numerical effects from the relative scales of each variable in the input data.

Since we have six axotomy observations and eight control observations, it is advantageous for classification accuracy to balance the classes in a combinatorial fashion. We therefore run each experiment $\binom{8}{6} = 28$ times, as was done in Section 4.3. In Figures 7.2 – 7.6, we show the mean of these 28 runs for each variable as a thick, solid line. To show the variability between each run, we also include quartile bars: the top of each quartile bar represents the 75 percentile, whereas the bottom of the bar represents the 25 percentile. For clarity, extreme values beyond these quartiles are not shown.

To visualize the class separability within these two data sets, the inputs scaled to approximate i.i.d. data are shown in Figure 7.1. This diagram serves only to illustrate that the

first 20 input variables of the toy data set are clearly separable, with no overlap between the two classes, and the remaining variables are purely Gaussian noise but are scaled according to Equation 7.3. In the retinal electrophysiology data set, no obviously separable inputs appear, however the data appears to be less random than the Gaussian noise in the latter variables of the toy data set. Quantifying the class separability of each dimension is the first of the filter methods we address below.

7.2.1 Fisher Ratio

The Fisher Ratio is a linear measure of the separability between classes for each variable. It is taken as the ratio of the absolute distance between the means of each class, to the sum of the standard deviation of each class. That is, for a particular input variable $\mathbf{x}_i$ (Chapelle *et al.*, 2002; Weston *et al.*, 2000)

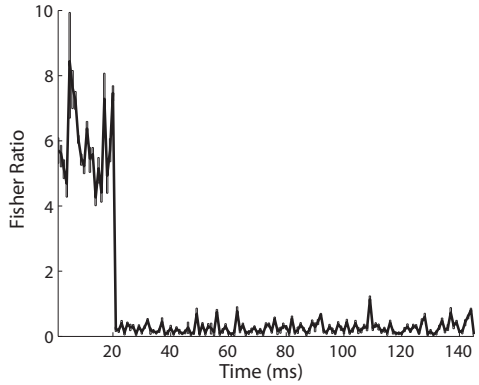
$$F(\mathbf{x}_i) = \left| \frac{\mu_i^+ - \mu_i^-}{\sigma_i^+ + \sigma_i^-} \right| \tag{7.4}$$

where $\mu_i^{\pm}$ are the means of variable $\mathbf{x}_i$ for the positive and negative classes, and $\sigma_i^{\pm}$ are the standard deviations of variable $\mathbf{x}_i$ for each class. A high Fisher Ratio for a particular variable indicates that for that variable, the two classes are easily separable from one another.

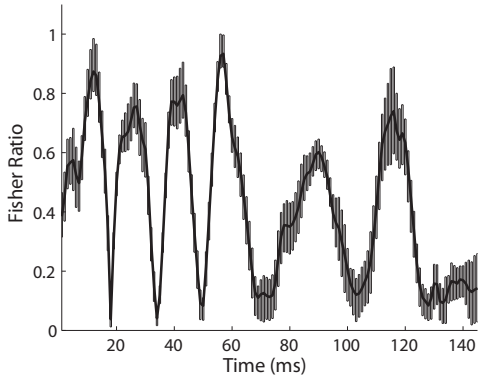
In Figure 7.2(a), the Fisher Ratios for each variable in the toy data set are what we would expect from Figure 7.1: the first 20 variables, which were chosen to be highly separable, have high values whereas the remaining variables are relatively low. However, there is a significant degree of random fluctuations between the different variables; this is most likely due to the small sample size. When we investigate the Fisher Ratios for each of the variables in the retinal electrophysiology data set in Figure 7.2(b), we see values in the same range $[0, 1]$ as many of the random variables in the toy data set. Peaks at about 10 and 60 ms have values approaching one, which could be interpreted as being equivalent to two Gaussian distributions of unit variance, with means separated by a distance of one: such distributions would have significant overlap.

7.2.2 Pearson Correlation Coefficients

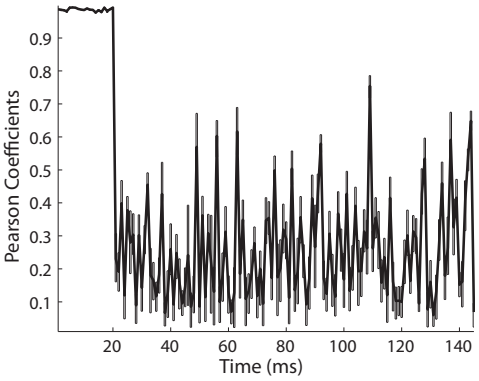
Pearson Correlation Coefficients quantify the linear correlation between the output variable and each variable in the input: if the two variables are strongly correlated, we would



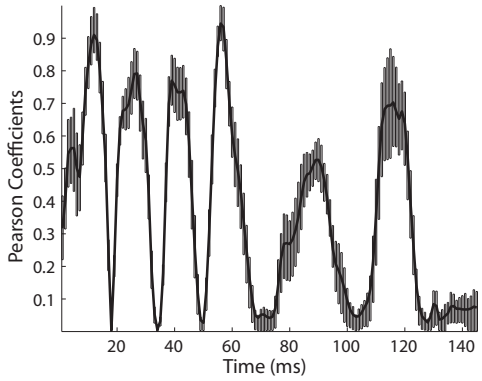
(a) Fisher ratio (toy).



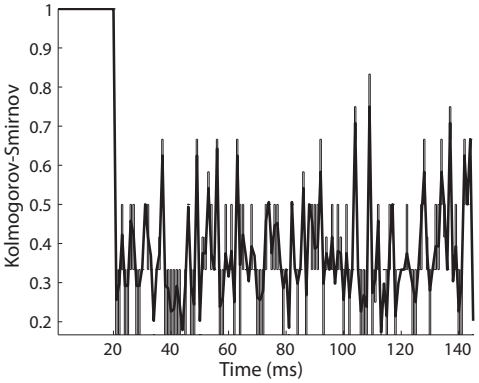
(b) Fisher ratio (retina).



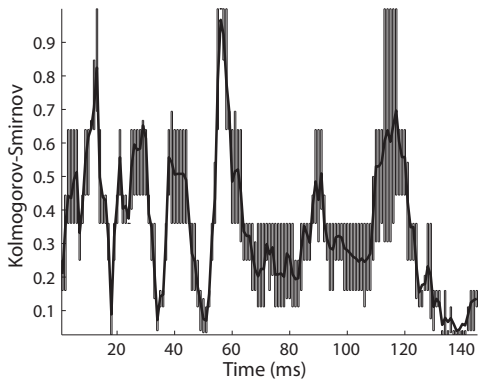
(c) Pearson coefficients (toy).



(d) Pearson coefficients (retina).



(e) Kolmogorov-Smirnov (toy).



(f) Kolmogorov-Smirnov (retina).

Figure 7.2: Comparing variable relevance measures for the toy and retinal electrophysiology data sets. Thick lines correspond with the mean, and quartile bars to the 25 and 75 percentiles, through each of the 28 possible class-balanced combinations of observations.

see a Pearson Correlation Coefficient approaching one. If the two variables are weakly correlated, we would see a coefficient approaching zero. If the two variables are inversely correlated, we would see negative values.

For a particular variable $\mathbf{x}_i$, the Pearson Correlation Coefficient is (Guyon and Elisseeff, 2003; Press *et al.*, 1992)

$$P(\mathbf{x}_i, y) = \frac{\sum_{j=1}^{\ell}(\mathbf{x}_{ij} - \mu_i)(y_j - \mu_y)}{\sqrt{\sum_{j=1}^{\ell}(\mathbf{x}_{ij} - \mu_i)^2}\sqrt{\sum_{j=1}^{\ell}(y_j - \mu_y)^2}} \tag{7.5}$$

where μ_i is the mean of $\mathbf{x}_i$, and μ_y is the mean of y.

The resulting coefficients will be $P_i(\mathbf{x}) \in [-1, +1]$. Here we will consider only the absolute value of the Pearson Correlation Coefficients, since we are interested mainly in determining whether a relationship between the variables exists: to the SVM there would be no difference, in terms of classification accuracy, between strongly positive or strongly negative correlation.

In Figure 7.2(c), the coefficients for the first 20 points approximate unity as we would expect. For the remaining points we would expect the correlation to be much smaller, however this is not the case: simply by chance, some of the variables do correlate quite strongly with the output variable. For example, we see a spike at about 110 ms with a coefficient of nearly 0.8. When we examine the coefficients for the retinal electrophysiology data set in Figure 7.2(d), we see some variables that correlate strongly with the output, with values approaching unity, and others that do not, with values approaching zero. Interestingly, the overall shape of the coefficients in Figure 7.2(d) seems to match closely with that of the Fisher Ratio in Figure 7.2(d): both measures appear to be indicating that the same areas of the waveform are significant, with approximately the same variability between the 28 different test runs.

7.2.3 Kolmogorov-Smirnov Test

Both the Fisher Ratio and Pearson Correlation Coefficients are linear, and therefore cannot model non-linear dependencies between the input and output variables (Chapelle *et al.*, 2002; Weston *et al.*, 2000). However, the Kolmogorov-Smirnov Test (Massey Jr., 1951) allows for non-linear dependencies, as it is based on the maximum absolute difference

between the empirical, cumulative distributions of each class. The statistic $KS(\mathbf{x}_i)$ may be written (Press *et al.*, 1992; Weston *et al.*, 2000)

$$KS(\mathbf{x}_i) = \max_{-\infty \leq x \leq \infty} \left| S_i^+(x) - S_i^-(x) \right| \tag{7.6}$$

where $S_i^{\pm}(x)$ are the unbiased, cumulative distribution functions for the positive and negative classes. These may be estimated from the observations $\mathbf{x}_i$ by the step functions (Heckert, 2003)

$$S_i^{\pm}(x) = n_i^{\pm}(x)/\ell \tag{7.7}$$

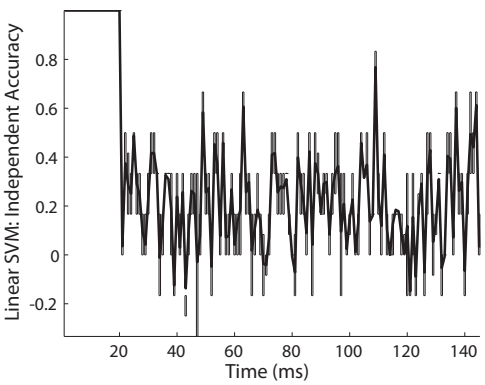
where $n_i^{\pm}(x)$ are a count of the observations of each class whose value for the i-th variable is less than or equal to x.

We wish to retain those input variables for which the empirical distributions for the positive and negative classes are the most different. As a necessary computational consideration, rather than varying $-\infty \leq x \leq \infty$, we allow $\min(\mathbf{x}_i) \leq x \leq \max(\mathbf{x}_i)$.

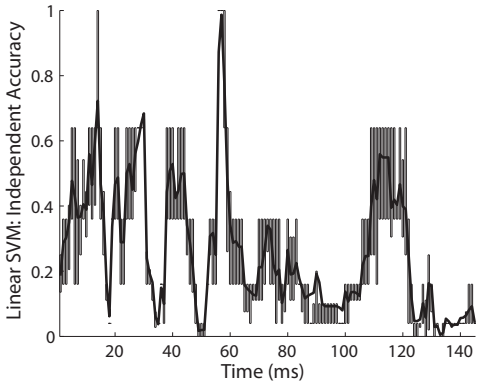
For a small number of observations ℓ, the resulting measures will appear to have low resolution, since the cumulative distributions change by exactly $1/\ell$ each time a new observation is added to the count. In Figure 7.2(e), the difference between the positive and negative distributions is maximized for the first 20 variables of the toy data set, as we would expect. However, for the remaining variables, there are significant variations from near zero (such that the empirical distributions are nearly identical) to more than 0.8. This lack of discrimination for the small sample size makes the Kolmogorov-Smirnov Test impractical for our purposes of assessing variable relevance. However, the parts of the waveform that this test determines to be significant in Figure 7.2(e) matches well with those of the Fisher Ratio and Pearson Correlation Coefficients, with similar peaks at about 10, 60 and 115 ms. We also see high variation between the 28 different combinations, for all input variables, at its worst near the tail of the waveform at about 120 ms.

7.2.4 Linear SVM

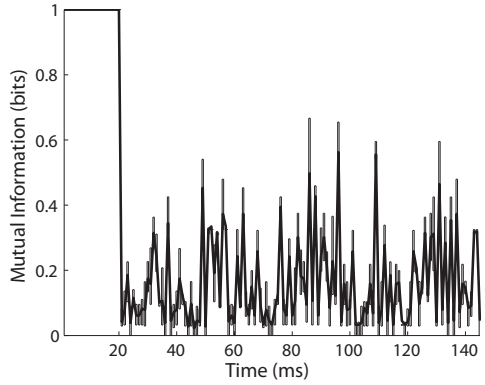
As a new proposed significance measure, we train a series of linear SVMs, one for each of the input variables taken independently, and measure the resulting accuracy that can be achieved using only that variable as the input. The cost parameter is left at the default $C = 1$ to provide a common basis for comparison. The resulting accuracy for each SVM is scaled such that a value of one indicates the SVM achieves 100% accuracy, while a value



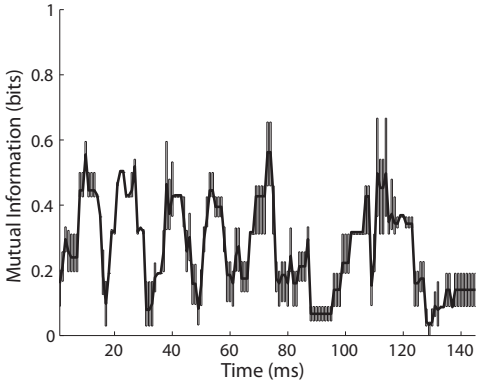
(a) Linear SVM (toy).



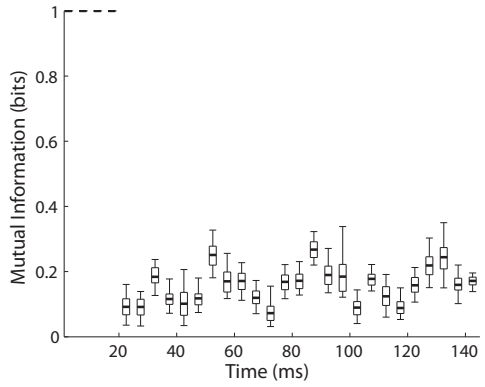
(b) Linear SVM (retina).



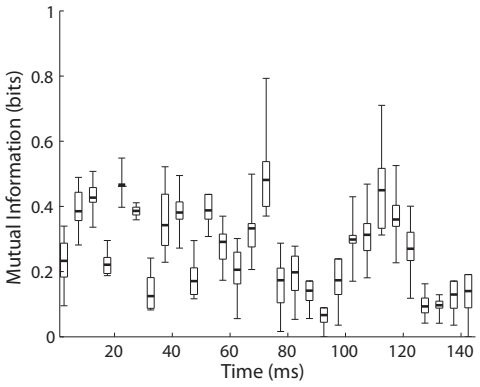
(c) Mutual information (toy).



(d) Mutual information (retina).



(e) Mutual information (toy): 5 ms bins.



(f) Mutual information (retina): 5 ms bins.

Figure 7.3: Comparing variable relevance measures for the toy and retinal electrophysiology data sets. Thick lines correspond with the mean, and quartile bars to the 25 and 75 percentiles, through each of the 28 possible class-balanced combinations of observations.

of zero indicates the SVM achieves 50% accuracy (no better than pure chance). We then have

$$\textit{LSVM}(\mathbf{x}_i, y) = 2\left(\frac{1}{2\ell}\sum_{j=1}^{\ell}|y_j - \text{sign}(f_i(\mathbf{x}_{ij}))|\right) - 1 \tag{7.8}$$

where sign$(f_i(\mathbf{x}_{ij}))$ is the output for $\mathbf{x}_{ij}$ of a linear SVM trained on variable $\mathbf{x}_i$ and targets y. This will allow some negative values, when the classifier achieves less than 50% accuracy.

The results from this measure applied to the toy data set are shown in Figure 7.3(a). As we would expect, each of the first 20 variables, taken as the only input to a linear SVM, is able to achieve 100% classification accuracy. Since the remaining variables are purely random, some of these variables are able to achieve fairly high accuracy, simply by chance: at about 110 ms there is a significant spike to 0.8, as was seen in the Pearson Correlation Coefficients and Kolmogorov-Smirnov tests. The variability in the results for these random input variables is too great for our purposes, however when the test is performed for the retinal electrophysiology data set in Figure 7.3(b), we see the same general shape as with the other tests, with a significant spike occurring at about 60 ms.

7.2.5 Mutual Information

The mutual information between each of the input variables and the output variables was estimated using software from Moddemeijer (1989), which uses a coarse histogram to estimate the joint and prior probabilities in (Carlson, 1986; Guyon and Elisseeff, 2003)

$$I(\mathbf{x}_i, y) = \sum_{j=1}^{\ell}\sum_{k=1}^{\ell} P(\mathbf{x}_{ij}, y_k)\log\frac{P(\mathbf{x}_{ij}, y_k)}{P(\mathbf{x}_{ij})\,P(y_k)} \tag{7.9}$$

where $P(\mathbf{x}_{ij}, y_k)$ is the joint probability distribution of $\mathbf{x}_i$ and y, $P(\mathbf{x}_{ij})$ is the prior probability of the input variable $\mathbf{x}_i$ and $P(y_k)$ is the prior probability of the outputs y. The resulting measure is returned in bits. Since we have a binary decision to be based on only a single input, $I(\mathbf{x}_i, y) \in [0, 1]$.

In Figure 7.3(c), we see that any of the first 20 input variables is enough to fully determine the output variable in the toy data set, with an information value of 1 bit (that is, any one of the first 20 variables is sufficient to make a single binary decision). Since we have a small sample size, there is still a substantial amount of information in the remaining random variables. In Figure 7.3(d), the same procedure for the retinal electrophysiology data set gives results in approximately the same range of values.

However, many of the variables that are near one another have similar information content. To determine whether we can use this to improve the result, the input variables were accumulated into 5 ms bins in Figures 7.3(e) and (f) for each data set. In these box plots, the minimum and maximum values for each variable are shown as well as the 25% and 75% quartiles. The thick line in the centre of each box is the mean information content of that variable across the 28 balanced combinations of observations. For the toy data set, this has the effect of reducing the significance of the random variables, which is desirable. Although much detail is lost in the peaks for the real data set, we find that the two peaks at about 75 and 115 ms are more prominent in (f) than in (d).

In comparison to the linear SVM in Figure 7.3(b), the strong peak at about 60 ms has disappeared completely in both the binned and unbinned mutual information figures. This strong peak had indicated that the linear SVM was able to completely determine the output class based solely on this single input value: the fact that this peak is missing from the mutual information measure indicates that mutual information will not give a true indication of the significance of these variables in the context of SVM classification, although the remaining peaks are roughly comparable with the previous measures. A windowed measure such as those examined for the sensitivity measures later in this chapter, rather than the binned measure we evaluate here, may improve these results. For now, however, we wish only to compare mutual information with the remaining filter methods.

7.3 Variable Sensitivity

We have examined the separability, output correlation and information content of the input variables. The Kolmogorov-Smirnov Test compares the empirical distributions of the classes, and the Linear SVM Test shows the classification accuracy that can be achieved taking each variable independently. Next we shall measure how sensitive the actual SVM model used to perform the classification is, to each of the input variables taken independently and to groups of input variables in a sliding window.

7.3.1 Method

Here we simplify the approach of Rueda *et al.* (2004), to find the sensitivity of the SVM model to each of the input variables. As in Rueda *et al.* (2004), we define sensitivity as the absolute change in the output variables relative to the total change in input variables, that

is $S(i) = |\Delta f / \Delta x|$. However, in this work the intention of the authors was to combine sensitivity with backwards elimination, iteratively removing those input dimensions with a sensitivity less than a random variable inserted as an extra dimension during training, requiring many training sessions.

Here, we adapt a simpler approach to quantify the sensitivity of the classifier to each of the input variables, relative to each other, but require only a single training session using the heuristic proposed in Chapter 3. For each of the $\binom{8}{6} = 28$ combinations of class-balanced observations:

1. Scale inputs to approximate i.i.d. data such that $\mathbf{x}_i \in [-1, +1]$, $i = 1, \ldots, \ell$.

2. Optimize free SVM parameters C and γ by any method, and train an SVM classification model.

3. Determine $f_\varnothing$, the value of the SVM decision surface when all inputs are zero.

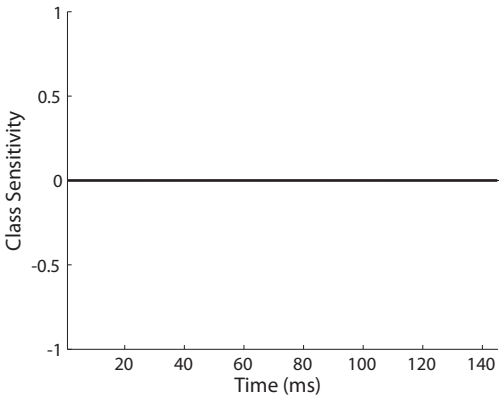
4. Vary each input variable independently to $\{-1, +1\}$, holding all remaining inputs at zero, and measure the value of the resulting decision surface $f_i^{\pm}$.

5. The sensitivity of each variable i is then the maximum absolute change in the value of the decision surface, as

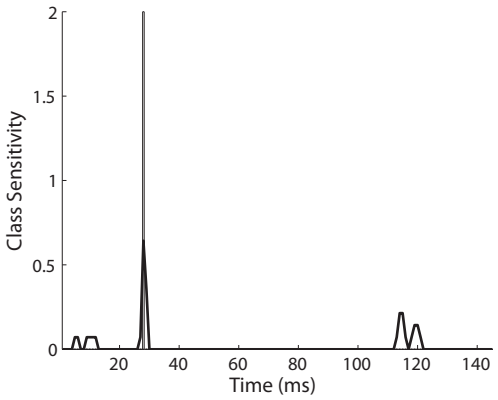
$$S(i) = \max \left| \frac{\Delta f}{\Delta x} \right| = \max \left(\left| f_i^+ - f_\varnothing \right|, \left| f_i^- - f_\varnothing \right| \right) \tag{7.10}$$

   since $|\Delta x| = 1$ for both trials of all input variables.

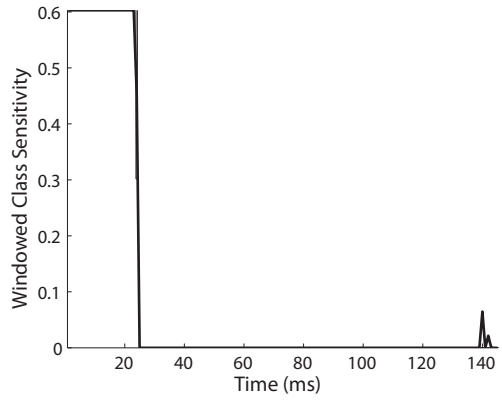
Here, LIBSVM (Chang and Lin, 2001) was used to optimize the parameters due to its computational efficiency, but SVM^{light} (Joachims, 1999) was used to measure the values of the decision surface in steps 3 and 4, as it provides open access to the value of the decision surface rather than just the sign of the decision surface. Both LIBSVM and SVM^{light} give the same model for the same training observations and free parameter values, the difference between the two is primarily in the different quadratic optimization algorithms employed. Another approach may be to repeatedly vary the inputs in step 4 by a random variable $\in [-1, +1]$ rather than the discrete values $\{-1, +1\}$, however for our purposes here we wish this process to be deterministic.



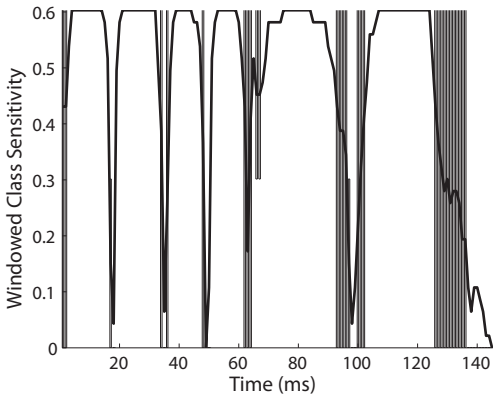
(a) Class sensitivity (toy).



(b) Class sensitivity (retina).



(c) Windowed class sensitivity (toy).



(d) Windowed class sensitivity (retina).

Figure 7.4: Comparing variable class-sensitivity for the toy and retinal electrophysiology data sets. Thick lines correspond with the mean, and quartile bars to the 25 and 75 percentiles, through each of the 28 possible class-balanced combinations of observations. The class sensitivity is the maximum absolute change in sign$(f(\mathbf{x}))$ that results from changing a single input variable from the mean of zero to either extreme $\in \{-1, +1\}$. The sliding window in the lower diagrams has a width of 11 ms, such that the variables changed for any variable i are $i - 5, \ldots, i + 5$ within the bounds $i = 1, \ldots, \ell$.

7.3.2 Class Sensitivity

Two variations on this procedure were tried. We first measure the class sensitivity rather than the surface sensitivity, that is

$$S(i) = \max\left(\left|\text{sign}(f_i^+) - \text{sign}(f_\varnothing)\right|, \left|\text{sign}(f_i^-) - \text{sign}(f_\varnothing)\right| \right) \tag{7.11}$$

This results in Figures 7.4(a) for the toy data set, and (b) for the retinal electrophysiology data set. In (a), we see that changing the value of any one variable is not enough to change the class determination: the sensitivity value for all variables is zero. In (b), we see a similar situation for most of the 28 class-balanced combinations, with some variables being sufficiently sensitive to change the class value for only a few particular combinations of observations.

A windowed approach was therefore used, to aggregate the values through a moving, square window with a 11 ms width. That is, such that eleven input variables would be changed at once to the same value $\in \{-1, +1\}$, rather than just a single variable, as

$$S(i) = \max\left(\left|\text{sign}(f^+_{i-5,\ldots,i+5}) - \text{sign}(f_\varnothing)\right|, \left|\text{sign}(f^-_{i-5,\ldots,i+5}) - \text{sign}(f_\varnothing)\right| \right) \tag{7.12}$$

This scheme will allow edge effects, where for $i \leq 5$ and $i > (\ell - 5)$ there will be fewer than eleven variables selected. The scale will be retained, since for each 11 ms window we take the value of the decision surface rather than an aggregate value. However, now we will need to adjust the attained sensitivity values by

$$|\Delta x| = \sqrt{\sum_{i-5,\ldots,i+5} (\pm 1)^2} = \sqrt{11} \approx 3.3 \tag{7.13}$$

This results in Figures 7.4(c) for the toy data set, and (d) for the retinal electrophysiology data set. In (c), we see that windows centred on the first 20 variables now do allow changes in class values. There is also a small peak at about 140 ms, indicating that a window centred on this location is enough to change the class values for a few of the 28 combinations of observations, but not all combinations. In (d), many windows of variables allow the class value to change. From about 120 ms, there is great variability between different combinations of observations, and we see a small edge effect after 140 ms as we would expect. Using a smaller 5 ms window helps to avoid the "clipped" appearance of this diagram, however there is still great variability between the different combinations of

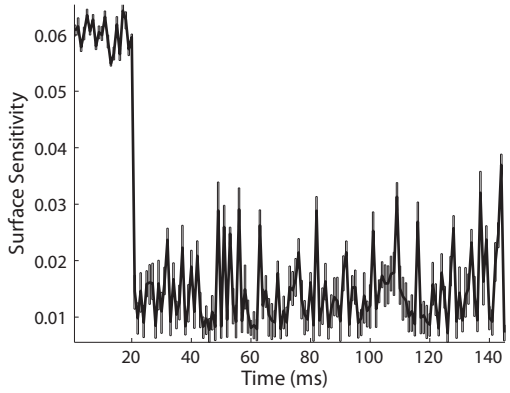
observations. The maximum sensitivity value that can be attained within the window is $|\Delta y/\Delta x| = |(\pm 1) - (\mp 1)|/\sqrt{11} \approx 0.6$.

From this figure, we have determined that the variables for which an 11 ms window centred on that variable achieved a change in class value for all combinations of observations are 4–14 ms, 21–32 ms, 39–44 ms, 53–59 ms, 75–84 ms and 107–124 ms. However, this does not tell us anything about the relative contribution of each of these peaks: for example, we cannot determine whether the range from 4–14 ms is any more or less significant than the range from 21–32 ms.

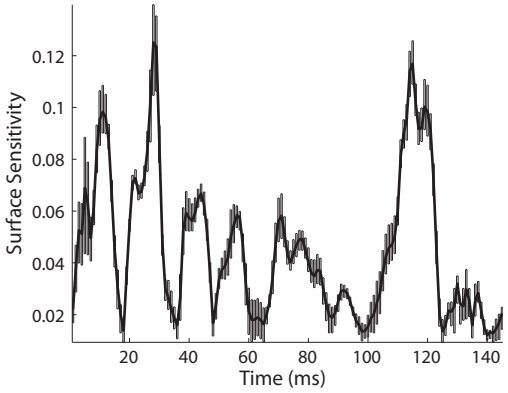
7.3.3 Surface Sensitivity

For a more fine-grained sensitivity analysis, we examine the sensitivity of the decision surface $f(x)$ itself, rather than the resulting class sign$(f(x))$. This results in Figures 7.5(a) for the toy data set, and (b) for the retinal electrophysiology data set. In (a), there is some variability between the first 20 input variables, indicating that this method is somewhat susceptible to input noise. This variability is increased in the remaining variables. However, when we compare with (b), we see that the magnitude of the change in the decision surface is actually quite small for any of the first 20 variables: there are many variables in (b) that can, independent of any other variable, affect the decision surface to the same degree. In fact, we see variables at about 30 ms and 110 ms that have more than twice the impact (~ 0.12) of any of the first 20 perfectly-separable variables in the toy data set (~ 0.06). This shows that in the SVM model, it truly is the combination of multiple variables that significantly affects the outcome: even a large change in only a single variable has little impact, improving the robustness of the classifier.

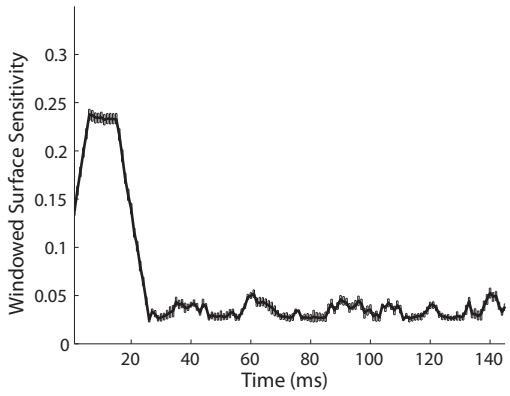
To reduce this noise sensitivity, we examine a windowed approach to surface sensitivity in Figures 7.5(c) and (d), as we did with class sensitivity in Section 7.3.2. In (c) for the toy data set, we now see a much smoother curve with significantly decreased noise between the different input variables. Changes in the first 20 variables have more than four times the impact (~ 0.24) than with the single-variable surface sensitivity, even though we normalize these windowed diagrams by $|\Delta x| = \sqrt{11}$ as mentioned above. There are edge effects at the beginning and end of the first 20 significant variables, as we would expect, however the remaining variables — which have no class relevance in the underlying distribution — have a dramatically reduced effect.



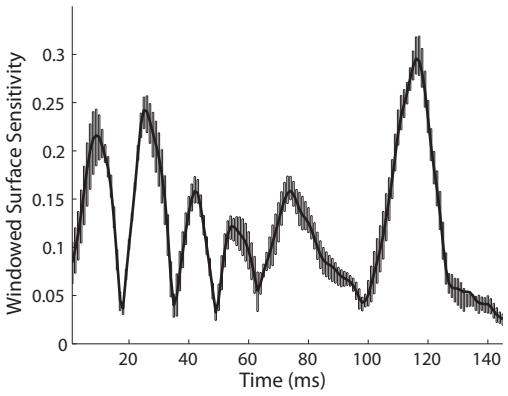
(a) Surface sensitivity (toy).



(b) Surface sensitivity (retina).



(c) Windowed surface sensitivity (toy).



(d) Windowed surface sensitivity (retina).

Figure 7.5: Comparing variable surface-sensitivity for the toy and retinal electrophysiology data sets. Thick lines correspond with the mean, and quartile bars to the 25 and 75 percentiles, through each of the 28 possible class-balanced combinations of observations. The surface sensitivity is the maximum absolute change in $f(\mathbf{x})$ that results from changing a single input variable from the mean of zero to either extreme $\in \{-1, +1\}$. The sliding window in the lower diagrams has a width of 11 ms, such that the variables changed for any variable i are $i - 5, \ldots, i + 5$ within the bounds $i = 1, \ldots, \ell$.

In (d), the peaks are now also smoother, and we can clearly see the differences in variability between the 28 different combinations of observations. The classifier is at least as sensitive to the first two peaks as to the first 20 variables in the toy data set, however we also see that it is more sensitive to the tail of the waveform at about 115 ms. Since we scale all variables independently to approximate i.i.d. data, such that $\mathbf{x}_i \in [-1, +1]$, it seems likely that within this region, small differences between the class distributions are amplified since the scaling factor is larger.

7.4 Discussion

We desire a measure which will emphasize both the importance of the first 20 variables in the toy data set and the insignificance of the remaining variables, with low variability within these two groups of input variables and low variability between the 28 different combinations of observations.

The Fisher Ratio differentiates well between the two groups of input variables, but is susceptible to large variations between the variables within each group. The Pearson Correlation Coefficients, Kolmogorov-Smirnov Test, Linear SVM and Mutual Information do not suffer from this variation, but do not differentiate well between the two groups. With the latter two measures, no single point on the waveform seems to be more relevant than the random variables in the toy data set. This effect is reduced by binning the mutual information results over 5 ms in Figure 7.3(e) and (f), however the results for the retinal electrophysiology data set are somewhat difficult to visualize. Despite these shortcomings, the measures agree to a large extent on which parts of the waveform are the most significant.

The variable sensitivity measures do not suffer from this lack of differentiation between groups. Changing a single variable at a time has little effect on either the class or the decision surface, but employing an 11 ms window results in much more significant changes. In particular, the windowed surface sensitivity measure gives us the result we desire from a variable selection measure: in the toy data set, we now have a method which shows a smooth curve, with low variability between the two groups of input variables and low variability between the 28 different runs, and which emphasizes the importance of the first 20 variables and reduces the impact of the remaining variables. We can also intuitively have more confidence in the results from these sensitivity measures, since they employ the true SVM models we use to perform the actual classification, with the same free parameters

chosen by the heuristic from Chapter 3.

In Figure 7.6, we compare the input variables chosen as significant by these methods to the original waveform. Each section of the waveform is labelled according to { N1, P1, N2, P2 }, standard designations for the field of retinal electrophysiology (see for example Marmor *et al.*, 2003), corresponding to negative (N) and positive (P) waves in order of occurrence. The input variables selected by windowed *class* sensitivity are shown as shaded regions, since all have equal weight. The windowed *surface* sensitivity is shown by the mean and quartile boxes as in Figure 7.5(d), with each peak indicated by a thin, dotted line extending between the two diagrams. The two methods agree for the most part, except for a small shift on the variables centred at about 80 ms.

The classifier appears to be most sensitive to the leading and trailing edges of each wave, in particular the leading edges of the N1 and P1 waves centred at about 10 and 25 ms respectively, and to the trailing part of the waveform centred at about 115 ms. Further measurements to be made during the course of this ongoing medical research at Dalhousie University's Retina and Optic Nerve Research Laboratory (RONRL) will help to determine if this effect is real, or simply a numerical coincidence due to variable scaling as mentioned in the previous section.

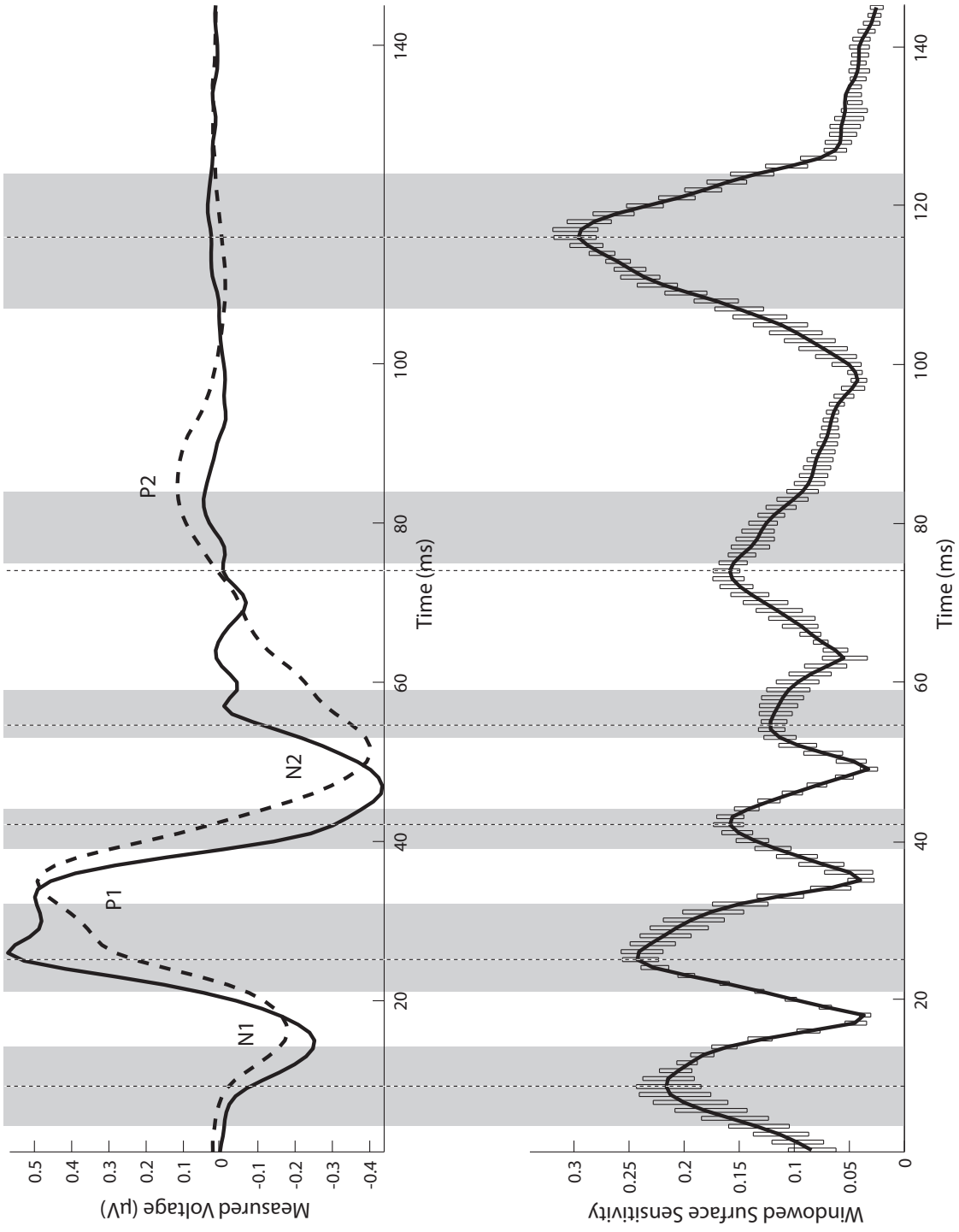


Figure 7.6: Comparison of the sensitivity measures to the mean PERG waveforms from Figure 4.5. On the *left*, the thick solid line is the mean waveform of the Axotomy class, whereas the dashed line is that of the Control class. On the *right*, the thick line and quartile bars correspond with the windowed *surface* sensitivity in Figure 7.5(d), and the thin, dotted lines mark each peak in the windowed surface sensitivity. The shaded regions are those identified by the windowed *class* sensitivity in Figure 7.4(d).

Chapter 8

Discussion

In this thesis, we have examined the generalization error surface resulting from examining a $\log_e$ range of values for the C cost parameter and the γ width parameter of the RBF kernel. We have proposed a heuristic to traverse this volatile error surface, including three primary elements: the use of a model complexity penalty to regularize the solution to prevent overfitting; the use of a guided, stochastic search using simulated annealing rather than a classical grid search; and the calculation of an intensity-weighted centre of mass to find the final optimum set of parameters.

Blind application of this annealing scheme was found to give good results with several classic and real-world classification problems. We have found that when optimizing free parameters, including a model complexity penalty enhances the generalizability of the final solution. We have found results with comparable accuracy to those of a grid-search, but with lower model complexity and greater search efficiency. In comparison with a grid search, we find that the guided, stochastic search concentrates the evaluated points to the area of interest to a much greater extent. In our tests, this has also had the unexpected yet desirable effect of improving the speed of convergence for the SVM trained at each point: speaking generally, very small or very large values of the free parameters will prevent the SVM algorithm from quickly converging to a solution. By concentrating on the area of interest, in which this convergence generally occurs more quickly, and by including a small origin bias in our move selection, this is largely prevented.

This technique can easily be extended to take further free parameters into account. For example, we have shown how the heuristic can be used to discover an optimal solution in a three-dimensional parameter space defined by C, γ and the width of the ε-tube used in the soft margin loss function for regression or function estimation problems using ε-SVR. We have successfully applied this to three real-world environmental modelling problems, and to noise reduction and the imputation of missing data for mitotic gene-expression in DNA microarray data.

We have kept the simulated annealing scheme quite simple, and have not tuned the annealing scheme for any particular problem other than to increase the length of the annealing

schedule when more evaluations significantly improved results, such as with the regression problems in Chapter 5. The slow cooling schedule allows a more extensive search of the parameter space and may be useful for finding global, narrow extrema, whereas the fast cooling schedule may be used to find a good solution more quickly. Although we have added several new parameters to control the annealing schedule, in practice the algorithm appears to be sensitive only to the δ parameter, controlling the length of the annealing scheme and therefore the number of evaluations performed, and to the λ parameter, controlling the tradeoff between empirical loss and model complexity.

We have investigated several input variable selection and sensitivity techniques to discover the most significant parts of the retinal electrophysiology waveform, including several classical filter techniques and a model-specific sensitivity analysis. We find that for this data set, an SVM classification model is most sensitive to the leading edges of the N1 and P1 waves, and perhaps to the tail of these waves during cell recovery although this is likely to simply be numerical coincidence due to input variable scaling.

Feature construction methods create alternative representations of the data (Guyon and Elisseeff, 2003). Frequency transformations, such as Fast Fourier Transforms (FFT), or wavelet transformations, such as the Harr, Daubechies or Symlet mother wavelets (Graps, 1995), might be used as inputs for waveform data such as the retinal electrophysiology data set we have explored in this thesis. In our early experiments with the retinal electrophysiology data set, we found some indications that a frequency representation using an FFT, or Harr, Symlet-1 or Symlet-3 wavelet representations with a decomposition level of four or higher, might achieve higher classification performance. However, we have found that such methods did not help nor hinder resulting accuracy to any statistically-significant level given the number of samples that were available to us: for example, for there to be a 95% probability that any differences in the results are not due to chance alone, a statistical-significance level of $p = 0.05$, we would typically require 38 observations (Norman and Streiner, 2003). But since this work is preliminary, we have only 14 samples available at present. Likewise, using second-order (slope) information as the input appeared to make no statistically-significant difference.

In terms of feature construction, it appears that a better choice is to provide all available inputs to the SVM, adjusting the distributions of the inputs to achieve i.i.d. data, and allow the training algorithm to decide which combinations of which inputs conflate to provide a feature space with the greatest separation between classes, allowing a maximal margin

width. Other representations of the data will likely either remove some information (for example, a slope representation of n inputs will reduce the number of inputs by one) or provide a larger number of dimensions unnecessarily (for example, an FFT representation may double the number of inputs since the transformation provides negative frequency intensities as well as positive frequencies).

In comparison to ANN, SVM have many advantages, such as intrinsic regularization, a sparse model representation, fewer free parameters and an efficient, convex training algorithm. However, SVM also have implicit limitations. SVM have high precision and high accuracy for continuous variables, but discrete variables, such as a traffic light signal which can take values "red," "amber" or "green" or a genomic sequence with values A, T, G or C, must be encoded to a continuous representation (Hsu *et al.*, 2003), thereby increasing the number of input variables. However, as we discuss in Chapter 7, increasing the number of input variables has little effect on the efficiency of the SVM training algorithm. SVM also do not currently have the capability for structured or hierarchical outputs, or truly multivariate outputs as we discuss in Chapter 6. However, such an SVM is currently being evaluated for specific applications (Tsochantaridis *et al.*, 2004). Multiple two-class classifiers (with binary outputs or with real-valued outputs) may be combined in a one-against-all or one-against-one fashion to estimate multi-class posterior probabilities (see for example Milgram *et al.*, 2005; Wu *et al.*, 2004). True multi-class SVM have also been proposed (Weston and Watkins, 1998), by changing the nature of an SVM from a quadratic optimization, as we discuss in Chapter 2, to a multi-class generalization of the underlying Lagrangian optimization problem.

Another limitation of "black box" classifiers such as SVM is that they have limited exposure to the decision rules indicated by the underlying model: to form an SVM classifier, for example, one need only determine the support vectors, the weighting α_i of each support vector and the hyperplane offset b (see Equation 2.26). The model itself contains no human-readable rules per se, other than the selection of representative support vectors. Rule extraction methods have therefore been proposed for SVM. For example, in Núñez *et al.* (2002), a clustering algorithm is used to determine representative *prototype* vectors for each class, and in Fung *et al.* (2005), linear SVM were used to extract non-overlapping, human-readable rules. In our sensitivity analysis in Chapter 7, we have addressed a similar problem by determining the sensitivity of the SVM model to each part of the input waveform, to quantify the relevance of each section in terms of classification performance with

the goal of understanding the biological mechanisms involved.

In conclusion, support vector machines include robust intrinsic regularization, but naïve choices of the free parameters will often result in unacceptable generalization error. Appropriate selection of free parameters is essential to achieving high performance. By including extrinsic regularization in the optimization of free parameters, we have proposed an approach that balances model complexity with classification or regression error. By traversing the generalization error using a simple simulated annealing algorithm, we reduce the number of function evaluations that must be performed. By including a centre-of-mass operation, we both reduce solution volatility and improve generalization error, moving suggested points in parameter space away from regions with sharp drops in accuracy. This approach is especially advantageous where there are few observed samples with high dimensionality ($\ell \ll d$). We have shown experimentally that such an approach will achieve high generalization performance with reduced computational complexity, for real-world classification and regression problems in electroretinography, environmental modelling and bioinformatics.

Bibliography

Agilent Technologies. Agilent SureScan technology. Technical Report 5988-7365EN, 2005. http://www.chem.agilent.com.

Richard E. Bellman, Ed. *Adaptive Control Processes*. Princeton University Press, 1961.

Kristen P. Bennett and Colin Campbell. Support vector machines: Hype or hallelujah? *SIGKDD Explorations*, Vol. 2, pp. 1–13, 2000.

Matthew D. Boardman and Thomas P. Trappenberg. A heuristic for free parameter optimization with support vector machines (in press). *Proceedings of the 2006 IEEE International Joint Conference on Neural Networks*, Vancouver, BC, July 2006.

Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pp. 144–152, Pittsburgh, PA, July 1992.

Michael P. S. Brown, William N. Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terrence S. Furey, Manuel Ares, Jr. and David Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences*, Vol. 97, No. 1, pp. 262–267, 2000.

Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, pp. 121–167, 1998.

Joaquin Q. Candela, Carl E. Rasmussen and Yoshua Bengio. Evaluating predictive uncertainty challenge (regression losses). In *Proceedings of the PASCAL Challenges Workshop*, Southampton, UK, April 2005. http://predict.kyb.tuebingen.mpg.de.

A. Bruce Carlson. *Communications Systems: An Introduction to Signals and Noise in Electrical Communication*, 3rd edition. McGraw-Hill, New York, NY, pp. 574–577, 1986.

Gavin Cawley. Predictive uncertainty in environmental modelling competition. Special session to be discussed at the *2006 IEEE International Joint Conference on Neural Networks*, Vancouver, BC, July 2006. Results available at http://theoval.cmp.uea.ac.uk/~gcc/competition.

CBS Corporation. *Numb3rs*. Epsiode 32: Dark matter, Aired: April 7, 2006. http://www.CBS.com/primetime/numb3rs.

Chih-C. Chang and Chih-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available from http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Olivier Chapelle, Vladimir N. Vapnik, Olivier Bousquet and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, Vol. 46, No. 1–3, pp. 131–159, 2002.

Olivier Chapelle and Vladimir N. Vapnik. Model selection for support vector machines. In S. Solla, T. Leen and K.-R. Müller, Eds., *Advances in Neural Information Processing Systems*, Vol. 12. MIT Press, Cambridge, MA, pp. 230–236, 1999.

Vladimir Cherkassky and Yunqian Ma. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, Vol. 17, No. 1, pp. 113–226, 2004.

Vladimir Cherkassky, Julio Valdes, Vladimir Krasnopolsky and Dimitri Solomatine. Applications of Learning and Data-Driven Methods to Earth Sciences and Climate Modeling, a special session held at the *2005 IEEE International Joint Conference on Neural Networks*. Montreal, Quebec, July 2005.

Jung K. Choi, Ungsik Yu, Sangsoo Kim and Ook J. Yoo. Combining multiple microarray studies and modeling interstudy variation, *Bioinformatics*, Vol. 19, pp. i84–i90, 2003.

Corinna Cortes and Vladimir N. Vapnik. Support-vector networks. *Machine Learning*, Vol. 20, No. 3, pp. 273–297, 1995.

Sven Degroeve, Koen Tanghe, Bernard De Baets, Marc Leman and Jean-Pierre Martens. A simulated annealing optimization of audio features for drum classification. In *Proceedings of the 6th International Conference on Music Information Retrieval*, pp. 482–487, London, UK, September 2005.

J. N. De Roach. Neural networks: An artificial intelligence approach to the analysis of clinical data. *Australasian Physical and Engineering Sciences in Medicine*, Vol. 12, No. 2, pp. 100–106, 1989.

Harris Drucker, Christopher J. C. Burges, Linda Kaufman, Alexander J. Smola and Vladimir N. Vapnik. Support vector regression machines. In M. C. Mozer, M. I. Jordan and T. Petsche, Eds., *Advances in Neural Information Processing Systems*, Vol. 9. MIT Press, Cambridge, MA, pp. 155–161, 1997.

Rong-E. Fan, Pai-H. Chen and Chih-J. Lin. Working set selection using the second order information for training support vector machines. *Journal of Machine Learning Research*, Vol. 6, pp. 1889–1918, 2005.

Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, Vol. 7, No. 2, pp. 179–188, 1936.

Frauke Friedrichs and Christian Igel. Evolutionary tuning of multiple SVM parameters. *Proceedings of the 12th European Symposium on Artificial Neural Networks*, pp. 519–524, Bruges, Belgium, April 2004.

Rule extraction from linear support vector machines. *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 32–40, Chicago, IL, August 2005.

Audrey P. Gasch, Paul T. Spellman, Camilla M. Kao, Orna Carmel-Harel, Michael B. Eisen, Gisela Storz, David Botstein and Patrick O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, Vol. 11, No. 12, pp. 4241–4257, 2000.

Walter R. Gilks, Brian D. M. Tom and Alvis Brazma. Fusing microarray experiments with multivariate regression. *Bioinformatics*, Vol. 21 (Supplement 2), pp. ii137–ii143, 2005.

Amara Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, Vol. 2, No. 2, pp. 50–61, 1995.

Isabelle M. Guyon. *SVM application list*, 1999–2006. Available at `http://www.clopinet.com/isabelle/Projects/SVM/applist.html`.

Isabelle M. Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, Vol. 3, No. 7–8, pp. 1157–1182, 2003.

Trevor Hastie, Robert Tibshirani and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, New York, NY, 2001.

Simon Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd edition. Prentice-Hall, Upper Saddle River, NJ, pp. 267–277, 1999.

David Heckerman. *A tutorial on learning with Bayesian networks*. MIT Press, Cambridge, MA, pp. 301–354, 1998.

Alan Heckert. Kolmogorov-Smirnov goodness of fit test. *National Instutute of Standards and Technology: Dataplot*, Vol. 1, Ch. 15, 2003. Available at `http://www.itl.nist.gov`.

Chih-W. Hsu, Chih-C. Chang and Chih-J. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 2003. Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Wolfgang Huber, Anja von Heydebreck, Holger Sültmann, Annemarie Poustka and Martin Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, Vol. 18 (Supplement 1), pp. S96–S104, 2002.

F. Imbault and K. Lebart. A stochastic optimization approach for parameter tuning of support vector machines. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, Vol. 4, pp. 597–600, Cambridge, UK, August 2004.

Thorsten Joachims. Making large-scale SVM learning practical. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, pp. 169–184, 1999. Software available at `http://svmlight.joachims.org`.

Daniel Johansson, Petter Lindgren and Anders Berglund. A multivariate approach applied to microarray data for identification of genes with cell cycle-coupled transcription. *Bioinformatics*, Vol. 19, pp. 467–473, 2003.

Rebecka Jörnsten, Hui-Y. Wang, William J. Welsh and Ming Ouyang. DNA microarray data imputation and significance analysis of differential expression. *Bioinformatics*, Vol. 21, No. 22, pp. 4155–4161, 2005.

M. Kathleen Kerr, Mitchell Martin and Gary A. Churchill. Analysis of variance for gene expression microarray data. *Journal of Computational Biology*, Vol. 7, No. 6, pp. 819–837, 2000.

S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi. Optimization by simulated annealing. *Science*, Vol. 220, No. 4598, pp. 671–680, 1983.

Olvi L. Mangasarian and William H. Wolberg. Cancer diagnosis via linear programming. *Society for Industrial and Applied Mathematics News*, Vol. 23, No. 5, pp. 1–18, 1990.

Michael F. Marmor, Donald C. Hood, David Keating, Mineo Kondo, Mathias W. Seeliger and Yozo Miyake. Guidelines for basic multifocal electroretinography (mfERG). *Documenta Ophthalmologica*, Vol. 106, No. 2, pp. 105–115, 2003.

Marie-L. Martin-Magniette and Julie Aubert and Eric Cabannes and Jean-J. Daudin. Evaluation of the gene-specific dye bias in cDNA microarray experiments. *Bioinformatics*, Vol. 21, No. 9, pp. 1995–2000, 2005.

Ann-M. Martoglio and James W. Miskin and Stephen K. Smith and David J. C. MacKay. A decomposition model to track gene expression signatures: preview on observer-independent classification of ovarian cancer. *Bioinformatics*, Vol. 18, No. 12, pp. 1617–1624, 2002.

Frank J. Massey, Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, Vol. 46, No. 253, pp. 68–78, 1951.

Boriana L. Milenova, Joseph S. Yarmus and Marcos M. Campos. SVM in Oracle Database 10g: Removing the barriers to widespread adoption of support vector machines. *Proceedings of the 31st International Conference on Very Large Data Bases*, pp. 1152–1163, Trondheim, Norway, August 2005.

Estimating accurate multi-class probabilities with support vector machines. *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, Vol. 3, pp. 1906–1911, Montreal, Quebec, July 2005.

Meghan T. Miller, Anna K. Jerebko, James D. Malley and Ronald M. Summers. Feature selection for computer-aided polyp detection using genetic algorithms. In A. V. Clough and A. A. Amini, Eds., *Medical Imaging 2003: Physiology and Function: Methods, Systems and Applications*, *Proceedings of the International Society for Optical Engineering*, Vol. 5031, pp. 102–110, 2003.

Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.

Rudy Moddemeijer. On Estimation of Entropy and Mutual Information of Continuous Distributions. *Signal Processing*, Vol. 16, No. 3, pp. 233–246, 1989. Software available from `http://www.cs.rug.nl/~rudy/matlab`, 2001.

Michinari Momma and Kristin P. Bennett. A pattern search method for model selection of support vector regression. In *Proceedings of the 2nd Society for Industrial and Applied Mathematics International Conference on Data Mining*, Philadelphia, PA, April 2002.

Klaus-R. Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda and Bernhard Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, Vol. 12, No. 2, pp. 181–202, 2001.

Ian T. Nabney. *Netlab: Algorithms for Pattern Recognition*. Springer-Verlag, New York, NY, 2002. Software available at `http://www.ncrg.aston.ac.uk/netlab`.

Julia Neumann, Christoph Schnörr and Gabriele Steidl. SVM-based feature selection by direct objective minimisation. *Proceedings of the 26th Deutsche Arbeitsgemeinschaft für Mustererkennung (German Symposium on Pattern Recognition)*, Vol. 3175, pp. 212–219, Tübingen, Germany, August 2004.

David J. Newman, S. Hettich, C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. Available at `http://www.ics.uci.edu/~mlearn`.

Geoffrey R. Norman and David L. Streiner. *PDQ (Pretty Darned Quick) Statistics*, 3rd edition. BC Decker, Hamilton, Ontario, 2003.

Haydemar Núñez, Cecilio Angulo and Andreu Català. Rule extraction from support vector machines. *Proceedings of the 10th European Symposium On Artificial Neural Networks*, pp. 107–112, Bruges, Belgium, April 2002.

Christine A. Orengo, David T. Jones and Janet M. Thornton. *Bioinformatics: Genes, Proteins & Computers*. Springer-Verlag, New York, NY, 2003.

John C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges and A. J. Smola, Eds., *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, pp. 185–208, 1999.

William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd edition. Cambridge University Press, 1992.

Royal Holloway, University of London Press Office. Highest professional distinction awarded to Professor Vladimir Vapnik. *College News*, March 8, 2006. `http://www.rhul.ac.uk`.

Ismael E. A. Rueda, Fabio A. Arciniegas and Mark J. Embrechts. SVM sensitivity analysis: An application to currency crises aftermaths. *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans*, Vol. 34, No. 3, pp. 387–398, 2004.

Gary L. Russell, James R. Miller and David Rind. A coupled atmosphere-ocean model for transient climate change studies. *Atmosphere–Ocean*, Vol. 33, No. 4, pp. 683–730, 1995.

Gabriella Rustici, Juan Mata, Katja Kivinen, Pietro Lió, Christopher J. Penkett, Gavin Burns, Jacqueline Hayles, Alvis Brazma, Paul Nurse and Jürg Bähler. Periodic gene expression program of the fission yeast cell cycle. *Nature Genetics*, Vol. 36, No. 8, pp. 809–817, 2004.

Bernhard Schölkopf. *Support vector learning* (PhD dissertation). Technische Universität Berlin, 1997.

Bernhard Schölkopf, Christopher J. C. Burges and Alexander Smola. Introduction to Support Vector Learning. In B. Schölkopf, C. J. C. Burges and A. J. Smola, Eds., *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, pp. 1–16, 1999a.

Bernhard Schölkopf, Alexander J. Smola and Klaus-R. Müller. Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges and A. J. Smola, Eds., *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, MA, pp. 327–352, 1999b.

Bernhard Schölkopf, Alexander J. Smola, Robert C. Williamson and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, Vol. 12, No. 5, pp. 1207–1245, 2000.

Bernhard Schölkopf, Kah-Kay Sung, Christopher J. C. Burges, Frederico Girosi, Partha Niyogi, Tomaso Poggio and Vladimir N. Vapnik. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, pp. 2758–2765, 1997.

Mark R. Segal, Kam D. Dahlquist and Bruce R. Conklin. Regression approaches for microarray data analysis. *Journal of Computational Biology*, Vol. 10, No. 6, pp. 961–980, 2003.

Yunfeng Shan, Evangelos E. Milios, Andrew J. Roger, Christian Blouin and Edward Susko. Automatic recognition of regions of intrinsically poor multiple alignment using machine learning. In *Proceedings of the 2003 IEEE Computational Systems Bioinformatics Conference*, pp. 482–483, Stanford, CA, August 2003.

Alexander J. Smola. *Regression estimation with support vector learning machines* (Master's thesis). Technische Universität München, 1996.

Alexander J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, Vol. 14, No. 3, pp. 199–222, 2004.

Gordon K. Smyth, Yee H. Yang and Terry Speed. Statistical issues in cDNA microarray data analysis. In M. J. Brownstein and A. B. Khodursky, Eds., *Functional Genomics: Methods and Protocols, Methods in Molecular Biology*, Vol. 224. Humana Press, Totowa, NJ, pp. 111–136, 2003.

Carl Staelin. Parameter selection for support vector machines. Technical Report HPL-2002-354 (R.1), HP Laboratories Israel, 2003.

E. H. K. Stelzer. Contrast, resolution, pixelation, dynamic range and signal-to-noise ratio: Fundamental limits to resolution in fluorescence light microscopy. *Journal of Microscopy*, Vol. 189, No. 1, pp. 15–24, 1998.

Daniel J. Strauss, Wolfgang Delb, Peter K. Plinkert and Jens Jung. Hybrid wavelet-kernel based classifiers and novelty detectors in biosignal processing. *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vol. 3, pp. 2865–2868, Cancun, Mexico, September 2003.

Erich E. Sutter and D. Tran. The field topography of ERG components in man—I. The photopic luminance response. *Vision Research*, Vol. 32, No. 3, pp. 433–446, 1992.

Y. C. Tai and T. P. Speed. A multivariate empirical Bayes statistic for replicated microarray time course data. Technical Report 667, University of California, Berkeley, 2004.

Jeffrey G. Thomas, James M. Olson, Stephen J. Tapscott and Lue Ping Zhao. An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. *Genome Research*, Vol. 11, No. 7, pp. 1227–1236, 2001.

Andrei N. Tikhonov. The regularization of ill-posed problems (in Russian). *Doklady Akademii Nauk USSR*, Vol. SSR 153, No. 1, pp. 49–52, 1963.

Thomas Trappenberg. Coverage-performance estimation for classification with ambiguous data. In Michel Verlysen, Ed., *Proceedings of the 13th European Symposium On Artificial Neural Networks*, pp. 411-416, Bruges, Belgium, April 2005.

Thomas Trappenberg, Jie Ouyang and Andrew Back. Input variable selection: Mutual information and linear mixing measures. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 18, No. 1, pp. 37–46.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein and Russ B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, Vol. 17, No. 6, pp. 520–525, 2001.

Chen-A. Tsai, Huey-M. Hsueh and James J. Chen. A generalized additive model for microarray gene expression data analysis. *Journal of Biopharmaceutical Statistics*, Vol. 14, No. 3, pp. 553–573, 2004.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. *Proceedings of*

the 21st International Conference on Machine Learning, pp. 104–111, Banff, Alberta, July 2004. Software available at `http://svmlight.joachims.org`.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, 1995.

Vladimir N. Vapnik and Alexey Ja. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities (in Russian). *Doklady Akademii Nauk USSR*, Vol. 181, No. 4, pp. 781–784, 1968.

Vladimir N. Vapnik and Alexey J. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of probabilities and its applications*, Vol. 16, pp. 264–280, 1971, English translation by *Soviet Mathematical Reports*.

Vladimir N. Vapnik and Alexey Ja. Chervonenkis. *Theory of Pattern Recognition* (in Russian). Nauka, Moscow, USSR, 1974.

Vladimir N. Vapnik, Steven E. Golowich and Alexander J. Smola. Support vector method for function approximation, regression estimation and signal processing. In M. C. Mozer, M. I. Jordan and T. Petsche, Eds., *Advances in Neural Information Processing Systems*, Vol. 9. MIT Press, Cambridge, MA, pp. 281–287, 1997.

H.-Q. Wang, D.-S. Huang and B. Wang. Optimisation of radial basis function classifiers using simulated annealing algorithm for cancer classification. *Electronics Letters*, Vol. 41, No. 11, 2005.

Xian Wang, Ao Li, Zhaohui Jiang and Huanqing Feng. Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme, *BMC Bioinformatics*, Vol. 7, No. 32, 2006.

Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio and Vladimir N. Vapnik. Feature selection for SVMs. In T. K. Leen, T. G. Dietterich and V. Tresp, Eds., *Advances in Neural Information Processing Systems*, Vol. 13. MIT Press, Cambridge, MA, pp. 668–674, 2000.

Jason Weston and Chris Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Royal Holloway, University of London, 1998.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edition. Morgan Kaufmann, San Francisco, CA, 2005. Software available at `http://www.cs.waikato.ac.nz/ml/weka`.

Lior Wolf and Stanley M. Bileschi. Combining variable selection with dimensionality reduction. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 801–806, 2005.

Baolin Wu. Differential gene expression detection using penalized linear regression models: the improved SAM statistics. *Bioinformatics*, Vol. 21, No. 8, pp. 1565–1571, 2005.

Ting-F. Wu, Chih-J. Lin and Ruby C. Weng. Probability estimates for multi-class classification by pairwise coupling. *The Journal of Machine Learning Research*, Vol. 5, pp. 975–1005, 2004.

Lei Yu and Huan Liu. Efficiently handling feature redundancy in high-dimensional data. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 685–690, Washington, DC, August 2003.