Sparse Machine Learning Methods with Applications in Multivariate Signal Processing

Thomas Robert Diethe

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of the

University of London.

Department of Computer Science

University College London

2010

I, Thomas Robert Diethe, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

This thesis details theoretical and empirical work that draws from two main subject areas: Machine Learning (ML) and Digital Signal Processing (DSP). A unified general framework is given for the application of sparse machine learning methods to multivariate signal processing. In particular, methods that enforce sparsity will be employed for reasons of computational efficiency, regularisation, and compressibility. The methods presented can be seen as modular building blocks that can be applied to a variety of applications. Application specific prior knowledge can be used in various ways, resulting in a flexible and powerful set of tools. The motivation for the methods is to be able to learn and generalise from a set of multivariate signals.

In addition to testing on benchmark datasets, a series of empirical evaluations on real world datasets were carried out. These included: the classification of musical genre from polyphonic audio files; a study of how the sampling rate in a digital radar can be reduced through the use of Compressed Sensing (CS); analysis of human perception of different modulations of musical key from Electroencephalography (EEG) recordings; classification of genre of musical pieces to which a listener is attending from Magnetoencephalography (MEG) brain recordings. These applications demonstrate the efficacy of the framework and highlight interesting directions of future research.

Acknowledgements

To my parents, who have supported my education from start to finish, thank-you so much for giving me this opportunity. To my supervisor John Shawe-Taylor, whose breadth and depth of knowledge never ceases to amaze me, thank-you for your guidance.

The research leading to the results presented here has received funding from the EPSRC grant agreement EP-D063612-1, *"Learning the Structure of Music"*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Machine Learning

ML is a relatively young field that can be considered an extension of traditional statistics, with influences from optimisation, artificial intelligence, and theoretical computer science (to name but a few). One of the fundamental tenets of ML is statistical inference and decision making, with a focus on prediction performance of inferred models and exploratory data analysis. In contrast to traditional statistics, there is less focus on issues such as coverage (*i.e.* the interval for which it can be stated with a given level of confidence contains at least a specified proportion of the sample). In statistics, classical methods rely heavily on assumptions which are often not met in practice. In particular, it is often assumed that the data residuals are normally distributed, at least approximately, or that the central limit theorem can be relied on to produce normally distributed estimates. Unfortunately, when there are outliers in the data, classical (linear) methods often have very poor performance. This calls for theoretically justified non-linear methods which require fewer assumptions. This is the approach that will be taken throughout this thesis, with a focus on developing a computational methodology for efficient inference with empirical evaluation. This will be backed up through analysis drawn from statistical learning theory, which allows us to make guarantees about the generalisation performance (or other relevant properties) of particular algorithms given certain assumptions on the classes of data.

1.2 Sparsity in Machine Learning

In information theory, the concept of *redundancy* is defined as the total number of bits used to transfer a message minus the number of bits of actual information in the signal. In ML redundancy appears in data in many forms. Perhaps the most common is noise - whether this is measurement noise or system noise - but there are also often domain specific sources of redundancy due to the nature of the data itself (*i.e.* high self-similarity) or to the way in which it is collected. In the particular application domains of interest in this thesis, namely multivariate signals, we are faced with potentially high levels

of both of these type of redundancy. Whenever there is redundancy in a dataset, there is the potential for sparse representations. In its most literal form, sparsity may involve a reduction in the number of data dimensions ("dimensionality reduction"), or in the number of examples needed to represent a pattern ("sample compression"). These two types of sparsity are known as "primal" and "dual" sparsity respectively, due to the concept of duality from the optimisation community (see *e.g.* [1]). Both of these types of sparsity have attractive properties, including:

- data compression,
- subset or feature selection,
- statistical stability (in terms of the generalisation of patterns),
- robustness (*i.e.* to outliers or small departures from model assumptions),
- space efficiency, and
- faster computations (after learning).

One of the biggest drawbacks of sparse methods tends to be in terms of computational efficiency during learning. Much of the work in this thesis will be focussed on optimisation methods for sparse learning that are computationally efficient. The most well known examples of sparse methods in statistics and ML include methods such as the Least Absolute Shrinkage and Selection Operator (LASSO) [2] and SVM [3], which are sparse in the primal and dual respectively. There are close relations between both of these methods as outlined by [4], and indeed with many other sparse methods such as LPBoost [5] and Kernel Basis Pursuit (KBP) [6]. Other classes of sparse methods include greedy methods such as Kernel Matching Pursuit (KMP) and methods based on random subsampling such as the Nyström method [7]. Chapter 2 will outline these and other methods and try to emphasise the linkage between them, whilst Chapter 3 builds on these methods to produce novel algorithms that are theoretically motivated and empirically validated.

1.3 Multivariate Signal Processing

As already alluded to, the specific class of data that will be the particular focus of this thesis is multivariate signals. The issues of redundancy and sparsity are particularly magnified within this domain, as the sensors used to gather the signals are often spatially proximal, and as a result their measurements are often highly correlated. In addition, many real-world signals are affected by a high degree of noise (which can be systemic noise or measurement noise). Finally, due to high rates of sampling and dense sensor grids, the data is often extremely high dimensional. It is therefore especially important that the methods used are capable of learning in this difficult domain.

Standard batch or online ML methods often fall short when analysing signals because the data violates one of the basic assumptions: that the data is *independently and identically distributed* (i.i.d.). There are of course a range of ML methods that deal specifically with non-i.i.d. data and in particular time series data, but the models are often highly complex and do not scale well to large datasets. In particular, these approaches often become intractable in the multivariate case - when we are dealing with

large sets of signals (as is often the case in biological applications, for example). Another approach to take is to break the signal into "chunks", perform a series of DSP operations on these chunks, and use the resulting data as examples in standard ML algorithms. Whilst the i.i.d. assumption is still violated, its impact is often softened as significant integration over time takes place. However care must be taken to avoid learning trivial relations due to this issue. The major benefit of this approach is that it means the problem of inference on signals can be "modularised", *i.e.* broken into subproblems, and subsequently highly developed methods from both DSP and ML can be applied. This approach will form the basis of the machine learning framework for multivariate signal processing that will be outlined in Chapter 3.

The links between DSP and ML run very deep, often with the same mathematical methods being used for different applications. In essence, both fields are interested in the solutions to underdetermined problems, inverse problems, and sparse estimation (see *e.g.* [8]). This means that there is fertile ground for cross-pollination of ideas; for example in Section 3.2 I will show how "greedy" methods from DSP can be used to solve ML optimisation problems, and use statistical learning theory analysis to give guarantees on the performance of the resulting algorithms.

1.4 Application Areas

1.4.1 Learning the Structure of Music

The funding and therefore main application area for this thesis was the EPSRC project entitled "Learning the Structure of Music", which encompasses three fields of science, music cognition, representation, and machine learning. The project was a collaborative effort between the Centre for Computational Statistics and Machine Learning at University College London, the Interdisciplinary Centre for Computer Music Research at the University of Plymouth, the Leibniz Institute for Neurobiology at the University of Magdeburg, and the Department of Computational Perception at the Johannes Kepler University Linz. The aims of the project were to develop models and tools that apply novel signal processing and machine learning techniques to the analysis of both musical data and brain imaging data on music cognition. The metrics of success for the project were in terms of both theoretical results and experimental results. Specifically, the goals were to deepen the understanding of the relationship between musical structure and musical performance, quantifiable by the ability to predict performer styles; to deepen the understanding of the relationship between musical structure and listening experience, quantifiable by the ability to predict patterns of brain activity; and to develop systems for generative performance and music composition, quantifiable by the ability to generate coherent musical performances and compositions.

The experimental research that falls within the scope of this thesis seeks to find common patterns between the features extracted from polyphonic music, and the representation of musical structures within the brain through the use of EEG and MEG recordings. This thesis is therefore targeted at the first two of the three goals described above. To this end, the experimental research initially naturally followed two paths, namely the understanding of polyphonic audio signal and of brain activity recordings, before integrating the two to search for common patterns. Each of these stages will be described in detail.

1.4.2 Music Information Retrieval

In the first part of the research, the goal was to investigate techniques for extracting features from music in two forms: score-based representations (*e.g.* Musical Instrument Digital Interface (MIDI)), and polyphonic music (*e.g.* Waveform Audio File Format (WAVE) audio). As most musical pieces are not available in the former of these representations, and the signal processing required to extract information from polyphonic audio is much more complicated, the research focussed on polyphonic audio. When available, however, score-based representations provide a rich source of information and this led to their use in later experiments involving human subjects. A broad range of audio features were considered, including musical structure, melody, harmony, chord sequences, or more general spectral or timbral characteristics. An initial survey of the field identified that classification of musical genre from audio files, as a fairly well researched area of music research, provided a good starting point. What would appear on the surface to be a relatively trivial task, is in reality difficult for a number of reasons, not least that the concept of a genre is rather subjective and amorphous. However despite these shortcomings, useful progress has been made in this area, including insights into the types of features that are appropriate for this kind of task and the types of algorithm best suited to the classification problem. Chapter 4 describes research into this area, and includes a description of the novel approach taken, as well as a discussion of the complications unearthed by this research.

1.4.3 Automatic analysis of Brain Signals

Neuroscience, like many other areas of science, is experiencing a data explosion, driven both by improvements in existing recording technologies, such as EEG, MEG, Positron Emission Tomography (PET), and functional Magnetic Resonance Imaging (fMRI). The improvements increase the quantity of data through these technologies have had a significant impact on basic and clinical neuroscience research. An analysis bottleneck is inevitable as the collection of data using these techniques now outpaces the development of new methods appropriate for analysis of the data, and the dimensionality of the data increases as the sensors improve in spatial and temporal resolution.

1.4.4 Additional Application Areas

Traditional processing of digital radar relies on sampling at the Nyquist frequency - *i.e.* twice the frequency of the highest part of the bandwidth required. This requires extremely fast and expensive Analogue to Digital Conversion (ADC) equipment, often operating at rates of up to 1 GHz. Methods that can reduce the frequency at which the ADC operates, or alternatively increase the signal bandwidth whilst operating at the same frequency, would be of great benefit to the radar community. A form of Compressed Sensing (CS) known Analogue to Information Conversion (AIC) [9, 10] that reduces the sampling frequency from the traditional Nyquist rate by sampling at the information rate, rather than the rate required to accurately reproduce the baseband signal, will be applied to real radar data in 4.

1.4.5 Published Works

The following publications have resulted from this work, and will be referenced where appropriate in the text.

Peer reviewed technical reports

Diethe, T., & Shawe-Taylor, J. (2007). Linear Programming Boosting for the Classification of Musical Genre. Technical Report Presented at the NIPS 2007 workshop Music, Brain & Cognition. [11]

Diethe, T., Durrant, S., Shawe-Taylor, J., & Neubauer, H. (2008). Semantic Dimensionality Reduction for the Classification of EEG according to Musical Tonality. Technical Report Presented at the NIPS 2008 workshop Learning from Multiple Sources. [12]

Diethe, T., Hardoon, D.R., & Shawe-Taylor, J. (2008). Multiview Fisher Discriminant Analysis. Technical Report Presented at the NIPS 2008 workshop Learning from Multiple Sources. [13]

Peer reviewed conference papers

Diethe, T., Durrant, S., Shawe-Taylor, J., & Neubauer, H. (2009). Detection of Changes in Patterns of Brain Activity According to Musical Tonality. Proceedings of IASTED Artificial Intelligence and Applications. [14]

Diethe, T., Hussain, Z., Hardoon, D.R., & Shawe-Taylor, J. (2009). Matching Pursuit Kernel Fisher Discriminant Analysis. Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, 5, 121-128. [15]

Diethe, T., Teodoru, G., Furl, N., & Shawe-Taylor, J. (2009). Sparse Multiview Methods for Classification of Musical Genre from Magnetoencephalography Recordings. Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009) Jyvskyl, Finland, online at `http://urn.fi/URN:NBN:fi:jyu-2009411242`. [16]

Diethe, T., & Hussain, Z. (2009). Kernel Polytope Faces Pursuit. Proceedings of ECML PKDD 2009, Part I, LNAI 5781, 290-301. [17]

Smith, G.E., Diethe, T., Hussain, Z., Shawe-Taylor, J., & Hardoon, D.R. (2010). Compressed Sampling For Pulse Doppler Radar. Proceedings of RADAR 2010. [18]

1.5 Structure of this thesis

The work in this thesis draws from several disparate areas of research, including digital signal processing, machine learning, statistical learning theory, psychology, and neuroscience. The next Chapter (2) will introduce some concepts from signal processing and machine learning that underly the theoretical and algorithmic developments, which are linked together into a coherent framework in Chapter 3. The following two Chapters, 4 and 5, will describe the experimental work described above in more detail, with

a focus on univariate and multivariate signal processing respectively. The final Chapter (6) concludes by giving some philosophical insights and discussion of intended future directions.

Chapter 2

Background

Abstract

Space and Time. *In this chapter I will provide background information for the two main subject areas that form the basis of the thesis: Machine Learning and Signal Processing. Machine Learning is a field that has grown from other fields such as Artificial Intelligence, Statistics, Pattern Recognition, Optimisation, and Theoretical Computer Science. The core goal of the field is to find methods that learn statistical patterns within data that are generalisable to unseen data using methods that are efficient and mathematically grounded. Signal processing is broader in the sense that there are multiple goals, such as control, data compression, data transmission, denoising, filtering, smoothing, reconstruction, identification etc., but narrower in the sense that it (generally) focusses on time-series data (which can be continuous or discrete, real or complex, univariate or multivariate). Where these fields intersect interesting challenges can be found that drive development in both fields.*

2.1 Machine Learning

An important feature of most developments in the field of ML that is derived directly from a computer science background is the notion of modularity in algorithm design. Modular programming (also known as 'Divide-and-Conquer') is a general approach to algorithm design which has several obvious advantages: when a problem is divided into sub-problems, different teams/programmers/research groups can work in parallel, reducing programme development time; programming, debugging, testing and maintenance are facilitated; the size of modules can be reduced to a humanly comprehensible and manageable level; individual modules can be modified to run on other platforms; modules can be re-used within a programme and across programmes. In the context of ML, modularity exists due to the existence of so called *kernel functions* (which will be explained below), which allow the problem of learning to be decomposed into the following stages: preprocessing, feature extraction, kernel creation (or alternatively weak-learner generation - see Section 2.1.11), and learning. This flow is depicted in Figure

2.1. Common to both ML and DSP is a desire not only to find solutions to problems, but also to do so

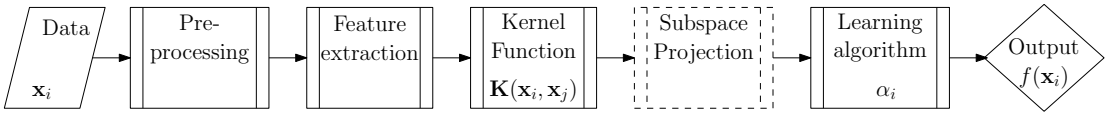


Figure 2.1: *Modularity of kernel methods*

efficiently. Drawing from optimisation theory, much work revolves around trying to find more efficient methods for solving problems that are exactly correct or approximately correct. The choice of optimisation method often comes down to a trade-off between computation time and memory requirements, or alternatively between accuracy of solutions and the time it takes to achieve them. Much of the focus of the next Chapter will be on different optimisation methods to achieve sparse solutions in computationally efficient ways. These methods include convex optimisation, iterative "greedy" methods, and methods that involve random subsampling or random projections. Examples of each of these methods will be introduced later in this Chapter.

ML deals with a wide variety of problems, from ranking of web-pages to learning of trading rules in financial markets. However the present focus will be on the more fundamental problems of classification, regression (function fitting and extrapolation), subspace learning and outlier detection. Many more complex tasks can be decomposed into these fundamental tasks, so it is important to focus on the foundations before building up to more complex scenarios. However common to all of the tasks is a focus on the generalisation ability of learnt models, so this will be the key metric upon which the empirical validation is grounded.

The first part of the Chapter will introduce some of the basic concepts mentioned above, firstly ML methods: regression, classification, regularisation, margin maximisation, boosting, subspace learning, and MVL; following from this will be DSP concepts such as dictionaries, bases, sparse representations, multivariate signal processing, and compressed sensing. Theoretical insights from Statistical Learning Theory (SLT) will be used to justify the methods as they are introduced.

2.1.1 Reproducing Kernel Hilbert Spaces

Outside of ML, the Reproducing Kernel Hilbert Spaces (RKHS) method provides a rigorous and effective framework for smooth multivariate interpolation of arbitrarily scattered data and for accurate approximation of general multidimensional functions. Given a Hilbert space $\mathcal{H}$ and an example $\mathbf{x}_i$, the reproducing property can be stated as follows,

$$f(\mathbf{x}_i) = \langle f, \kappa(\mathbf{x}_i, \cdot) \rangle_{\mathcal{H}} \tag{2.1}$$

of the reproducing kernel κ for every function $f(\mathbf{x}_i)$ belonging to $\mathcal{H}$. This property allows us to work in the implicit feature space defined only with the inner products, and is the key to kernel methods for ML.

This allows inner products between *nonlinear* mappings $\phi : \mathbf{x}_i \to \phi(\mathbf{x}_i) \in \mathcal{F}$ of $\mathbf{x}_i$ into a *feature space* $\mathcal{F}$, as long as the inner product $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ can be evaluated efficiently. In many

cases, this inner product or *kernel function* (denoted by κ) can be evaluated much more efficiently than the feature vector itself, which can even be infinite dimensional in principle. A commonly used kernel function for which this is the case is the Radial Basis Function (RBF) kernel, which is defined as:

$$\kappa_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right). \tag{2.2}$$

2.1.2 Regression

Given a sample S containing examples $\mathbf{x} \in \mathbb{R}^n$ and labels $y \in \mathbb{R}$. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)'$ be the input vectors stored in matrix $\mathbf{X}$ as row vectors, where $'$ denote the transpose of vectors or matrices.

Table A.1 in Appendix A is included as reference for some of the more commonly used mathematical symbols.

The following assumptions will be made in order to aid presentation: Data is centered (or alternatively a column of ones can be added as an extra feature, which will function as the intercept); the data is generated i.i.d. according to an unknown but fixed distribution $\mathcal{D}$. Furthermore, a Gaussian noise model with zero mean is assumed.

2.1.3 Loss functions for regression

Before going on to give specific examples of learning algorithms for regression, it is worth introducing the different loss functions that are commonly used for regression, along with their relation to the noise model.

Defining the square loss as

$$\mathcal{L}_{\{2\}} = \|f(\mathbf{x}) - y\|_2^2, \tag{2.3}$$

where $\hat{y} = f(\mathbf{x})$ is the estimate of the outputs y. This is also known as Gaussian loss as minimising this loss is the Maximum Likelihood solution if a Gaussian noise model is assumed. Alternatively we can denote the vector of slack variables $\boldsymbol{\xi} = |\mathbf{y} - \hat{\mathbf{y}}|$ as the differences between the true and estimated labels, and we divide by a half to make algebra easier, giving

$$\mathcal{L}_{\{2\}} = \frac{1}{2}\|\boldsymbol{\xi}\|_2^2. \tag{2.4}$$

The ℓ_1 loss is similarly defined as,

$$\mathcal{L}_{\{1\}} = \|\boldsymbol{\xi}\|_1, \tag{2.5}$$

whose minimisation leads to the Maximum Likelihood solution under a Laplacian noise model. Defining

Loss	functional $\mathcal{L}(\boldsymbol{\xi})$	density model $p(\boldsymbol{\xi})$
ϵ-insensitive	$\|\boldsymbol{\xi}\|_\epsilon$	$\frac{1}{2(1+\epsilon)} \exp\left(-\|\boldsymbol{\xi}\|_\epsilon\right)$
Laplacian	$\|\boldsymbol{\xi}\|_1$	$\frac{1}{2} \exp\left(-\|\boldsymbol{\xi}\|_1\right)$
Gaussian	$\frac{1}{2}\|\boldsymbol{\xi}\|_2^2$	$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\|\boldsymbol{\xi}\|_2^2}{2}\right)$
Huber's robust loss	$\begin{cases} \frac{1}{2\sigma}\|\boldsymbol{\xi}\|_2^2 & \text{if } \|\boldsymbol{\xi}\| \leq \sigma \\ \|\boldsymbol{\xi}\| - \frac{\sigma}{2} & \text{otherwise} \end{cases}$	$\propto \begin{cases} \exp\left(-\frac{\boldsymbol{\xi}^2}{2\sigma}\right) & \text{if } \|\boldsymbol{\xi}\| \leq \sigma \\ \exp\left(\frac{\sigma}{2} - \|\boldsymbol{\xi}\|\right) & \text{otherwise} \end{cases}$
Polynomial	$\frac{1}{d}\|\boldsymbol{\xi}\|^d$	$\frac{d}{2\Gamma(1/d)} \exp\left(-\|\boldsymbol{\xi}\|^d\right)$

Table 2.1: *Common loss functions and corresponding density models, adapted from [19]*

a region of width ϵ around zero within which deviations are not penalised leads to the ϵ-insensitive loss,

$$\mathcal{L}_{\{\epsilon,1\}} = \max\left(\|\boldsymbol{\xi}\|_1 - \epsilon, 0\right) \doteq \|\boldsymbol{\xi}\|_{\epsilon,1}, \quad \text{for the } \ell_1 \text{ noise model, and} \tag{2.6}$$

$$\mathcal{L}_{\{\epsilon,2\}} = \max\left(\|\boldsymbol{\xi}\|_2 - \epsilon, 0\right) \doteq \|\boldsymbol{\xi}\|_{\epsilon,2}, \quad \text{for the } \ell_2 \text{ noise model.} \tag{2.7}$$

Some loss functions and their equivalent noise models are given in Table 2.1. For simplicity, the rest of this Section will use the square loss of Equation 2.3. However any of the loss functions given (or other loss functions not given due to space constraints) can be substituted to give different optimisation criteria. This approach is known as the General Linear Model (GLM). In all of the cases outlined here, the loss function is convex which leads to exact optimisation problems. However, non-differentiable loss functions such as the linear loss or the ϵ-insensitive loss are typically harder to solve.

2.1.4 Linear regression in a feature space

Assume that data is generated according to a linear regression model,

$$y_i = \mathbf{x}_i \mathbf{w} + n_i, \tag{2.8}$$

where n is assumed to be an i.i.d. random variable (noise) with mean 0 and variance σ^2. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)'$ be the input vectors stored in matrix $\mathbf{X}$ as row vectors, and $\mathbf{y} = (y_1, \ldots, \mathbf{y}_m)'$ be a vector of outputs. Assume the square loss as defined in Equation 2.3, as this is the Maximum Likelihood solution to the linear regression problem of Equation 2.8. Intuitively it makes sense as the squaring of the errors places emphasis on larger errors whilst ignoring the sign. The formulation for linear regression that minimises this loss is then given by,

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{X}, \mathbf{y}, \mathbf{w}) \tag{2.9}$$

$$= \min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2. \tag{2.10}$$

By differentiating with respect to $\mathbf{w}$, equating to zero and rearranging, it can be seen that there is a closed form solution for $\mathbf{w}^*$,

$$\mathbf{w}^* = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}, \tag{2.11}$$

provided that the matrix $\mathbf{X}'\mathbf{X}$ is invertible. The dual of this optimisation is formed as follows,

$$\min_{\boldsymbol{\alpha}} \|\mathbf{X}\mathbf{X}'\boldsymbol{\alpha} - \mathbf{y}\|_2^2 \tag{2.12}$$

$$= \min_{\boldsymbol{\alpha}} \|\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\|_2^2, \tag{2.13}$$

which in turn has a closed form solution,

$$\boldsymbol{\alpha}^* = (\mathbf{X}\mathbf{X}')^{-1}\mathbf{y}, \tag{2.14}$$

$$= \mathbf{K}^{-1}\mathbf{y}, \tag{2.15}$$

again provided that the matrix $\mathbf{X}\mathbf{X}'$ is invertible. The function to test this model on a new data point is given by,

$$f(\mathbf{x}_i) = \mathbf{K}(i, \cdot)\boldsymbol{\alpha}^*. \tag{2.16}$$

This kernel trick is based on the reproducing property introduced in Section 2.1.1, with the observation that in the equation to compute $\boldsymbol{\alpha}^*$ (2.24) as well as in the equation to evaluate the regression function (2.16), all that is needed are the vectors $\mathbf{x}_i$ in inner products with each other. It is therefore sufficient to know these inner products only, instead of the actual vectors $\mathbf{x}_i$. Observe that the kernel regression form of Equation 2.15 when used with an RBF kernel has higher *capacity* than the linear regression form of Equation 2.11, *i.e.* it allows for a richer class of functions to be learnt than by the standard linear model. Whilst this increase in capacity may be desirable if the data is not in fact linear, in the presence of noise this can cause problems due to the ability of the model to fit the noise (*overfitting*). In this situation, some form of *capacity control* is required.

2.1.5 Stability of Regression

In statistics, this capacity control can be seen through what is known as the *bias variance trade-off* [20]. Typically, a model with low capacity such as the linear model of Equation 2.11, will have high bias as it will fit only a very restricted class of data, whilst the variance is low as perturbing some of the data points will have little effect. In contrast, if a high capacity model is used such as Equation 2.15 with the RBF kernel as defined in (2.2), the function can fit the data exactly (low bias) but if even a single data point is perturbed the function will change drastically (high variance). Hence it would be desirable to optimise the trade-off between these two in order to generate models with predictive power on new data. This is closely related to the concepts of overfitting and regularisation that will be discussed in Section

2.1.6.

McDiarmid's inequality [21], which is a generalization of Hoeffding's inequality [22], is a result in probability theory that gives an upper bound on the probability for the value of a function depending on multiple independent random variables to deviate from its expected value. This is a result that comes from the law of large numbers by Chernoff in relation to the convergence of Bernoulli trials [23]. The risk associated with a function f is defined as the expectation of the loss function,

$$\mathcal{R} = \mathbb{E}_{\mathbf{x}, y \in \{\mathcal{X} \times \mathcal{Y}\}}[\mathcal{L}(f(\mathbf{x}, y))], \tag{2.17}$$

and the empirical risk as the expectation of a particular sample S,

$$\begin{aligned} \hat{\mathcal{R}} &= \mathbb{E}_{\mathbf{x}, y \in S}[\mathcal{L}(f(\mathbf{x})] \\ &= \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(f(\mathbf{x}_i, y_i)). \end{aligned} \tag{2.18}$$

Given random variables $\mathbf{x}_i$ lying in the range $[a_i, b_i]$, the probability that the expected empirical risk $\hat{\mathcal{R}}$ differs from the true risk (or error) $\mathcal{R}$ by a value ϵ can be bounded as follows,

$$\Pr\left(|\hat{\mathcal{R}} - \mathcal{R}| \geq \epsilon\right) \leq 2 \exp\left(-\frac{2m\epsilon^2}{(b-a)^2}\right), \tag{2.19}$$

This shows that there is an exponential decay of the difference in the probabilities as the sample size increases. This gives us a clue that to learn well, the best thing that one can do is to increase the amount of data available. However, if this is not possible, the only other option is to control the capacity of the the learning algorithm.

Another viewpoint introduced by Vapnik and Chervonenkis is the notion of Structural Risk Minimisation (SRM) [24, 25, 26]. The real error $\mathcal{R}$ is upper bounded by the empirical error $\hat{\mathcal{R}}$ and another value called the *structural risk* $\mathcal{R}_S$. The structural risk is a theoretical criterion that can be computed for certain classes of models and estimated in most other cases. Choose the model that achieves the lowest upper bound.

$$\mathcal{R} = \hat{\mathcal{R}} + \mathcal{R}_S. \tag{2.20}$$

The idea is to impose a structure on the class of admissible functions $\mathcal{F}$, such that each individual function f_j which has lower capacity than the next f_{j+1}. This is depicted diagrammatically in Figure 2.2. Another closely related approach to capacity control is *regularisation*, which will be discussed below in Section 2.1.6. If we choose to control the capacity using a class of functions with bounded norm, we are in fact using the set of regularised functions, which gives an additional justification for this type of regularisation.

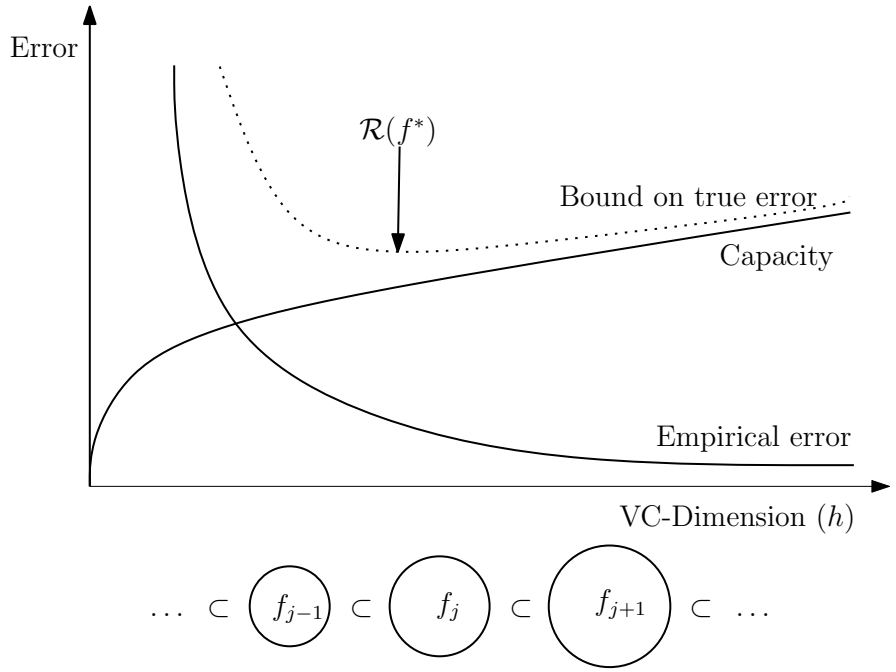


Figure 2.2: *Structural Risk Minimisation (adapted from [19]). The principle is to find the optimal function f^* that satisfies the trade-off between low capacity and low training error*

2.1.6 Regularisation

Inverse problems, such as (2.22) and (2.24) are often *ill-posed*. This is usually due to the condition number[1] of the matrix to be inverted, meaning that it needs to be re-formulated for numerical treatment. Typically this involves including additional assumptions, such as smoothness of solutions. This process is known in the statistics community as regularisation, and Tikhonov regularisation is one of the most commonly used types of regularisation for the solution of linear ill-posed problems [27]. There is also a secondary reason why regularisation is important: *overfitting*. Overfitting occurs when an inferred model describes the noise in the data rather than the underlying pattern. Overfitting generally occurs when the complexity of the model is too high in relation to the quantity of data available (*i.e.* in terms of degrees of freedom). A model which has been overfit will generally have poor generalisation performance on unseen data. Tikhonov regularisation is defined as,

$$\min_w \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \|\Lambda\mathbf{w}\|_2^2 \,, \tag{2.21}$$

where Λ is the Tikhonov matrix. Although at first sight the choice of the solution to this regularised problem may look artificial, the process can be justified from a Bayesian point of view. Note that for an ill-posed problem one must necessarily introduce some additional assumptions in order to get a stable solution. A statistical assumption might be that *a-priori* it is known that $\mathbf{X}$ is a random variable drawn from a multivariate normal distribution, which for simplicity is assumed to be mean zero and that each component is independent with standard deviation σ_x. The data is also subject to noise, and we take the errors in $\mathbf{y}$ to be also independent with zero mean and standard deviation σ_y. Under these assumptions,

[1] A "bad" condition number is one in which the quotient between the maximal and minimal eigenvalue of $\mathbf{\Sigma} = \mathbf{X}'\mathbf{X}$ is large

according to Bayes' theorem the Tikhonov-regularized solution is the most probable solution given the data and the *a-priori* distribution of $\mathbf{X}$. The Tikhonov matrix is then $\Lambda = \lambda\mathbf{I}$ for Tikhonov factor $\lambda = \sigma_y/\sigma_x$. Of course this Tikhonov factor is not known, so must be estimated in some way. If the assumption of normality is replaced by assumptions of homoscedasticity and that errors are uncorrelated, and still assume zero mean, then the Gauss-Markov theorem implies that the solution is a minimal unbiased estimate [28].

It is therefore justified to set the Tikhonov matrix to be a multiple of the identity matrix $\Lambda = \lambda\mathbf{I}$; this method is known in the statistics and ML literature as Ridge Regression (RR).

Ridge Regression

The primal formulation for RR is therefore given by,

$$\min_w \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 . \tag{2.22}$$

Similarly to (2.11), a closed form solution for RR exists,

$$\mathbf{w}^* = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}. \tag{2.23}$$

Using the duality theory of optimisation and the kernel trick once more, we obtain the following formulation for dual RR and hence Kernel Ridge Regression (KRR),

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \|\mathbf{X}\mathbf{X}'\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{X}'\boldsymbol{\alpha}\|_2^2 \\ =&\min_{\boldsymbol{\alpha}} \|\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \lambda\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} \end{aligned} \tag{2.24}$$

As with the unregularised case, there is again a closed form solution for this[2]

$$\begin{aligned} \boldsymbol{\alpha}^* &= (\mathbf{X}\mathbf{X}' + \lambda\mathbf{I})^{-1}\mathbf{y} \\ &= (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}. \end{aligned} \tag{2.25}$$

2.1.7 Sparse Regression

There is, however, nothing in either the primal (2.22) or the dual (2.24) formulations that would give rise to sparsity in the solutions ($\mathbf{w}^*$ or $\boldsymbol{\alpha}^*$ respectively). If we have prior knowledge that the weight vector generating the data was sparse, or alternatively we want to perform feature selection or subset selection, the above formulation can be modified to give sparse solutions. Replacing the ℓ_2-norm on the weights

[2]This comes from the normal equation $(\mathbf{K}^2 + \lambda\mathbf{K})\boldsymbol{\alpha} = \mathbf{K}\mathbf{y}$, so the closed form solution again depends on $\mathbf{K}$ (or $\mathbf{X}\mathbf{X}'$) being invertible.

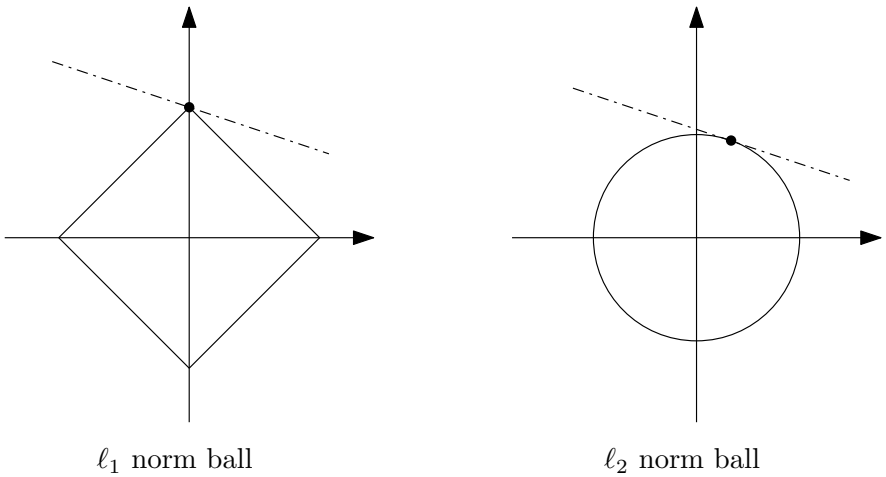


Figure 2.3: *Depiction of minimisation onto the ℓ_1 and ℓ_2 norm balls in $\mathbb{R}^2$. Note that at the optimal solution, the first coefficient (x-axis) is zero, and hence the solution is sparse. Note also that this will almost never be the case for the ℓ_2 norm.*

with the pseudo ℓ_0-norm [3] leads to the following optimisation,

$$\min_w \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_0 . \tag{2.26}$$

Finding this ℓ_0 solution is known to be $NP-$hard. However the ℓ_1 optimisation problem

$$\min_w \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_1 \tag{2.27}$$

is a convex quadratic programming problem, and is known to approximate the ℓ_0 solution (under certain conditions the solutions are identical see *e.g.* [29]). Since it is non-differentiable, unlike (2.11) or (2.22), there is no closed-form solution. The problem is variously known as the LASSO [30] and Basis Pursuit (BP) [31]. The reason for the sparsity in ℓ_1 solutions can be seen graphically in Figure 2.3. Methods for solving the LASSO problem include the forward stepwise regression algorithm [32], or the Least Angle Regression Solver (LARS) [2]. The LARS algorithm computes the full regularisation path, which is a piecewise linear function between $\lambda = 0$ and $\lambda = \infty$, which is a useful property if cross-validation (CV) is employed for model selection.

Whilst the dual optimisation for LASSO can be formulated [33], it does not lend itself easily to "kernelisation" - *i.e.* the weights cannot easily be represented as a linear combination of the data points in the form $\mathbf{w} = \mathbf{X}'\boldsymbol{\alpha}$. However, it is possible to perform "soft" kernelisation, where the inputs are simply replaced with the kernel matrix and the primal weight vector is replaced with the "soft" dual. This is the approach taken by [6] for the algorithm they call KBP, the formulation for which is,

$$\min_{\boldsymbol{\alpha}} \|\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 , \tag{2.28}$$

which can then be solved using any of the methods used to solve (2.27).

[3] The ℓ_0 pseudo-norm of a vector is simply a count of the non-zero entries

2.1.8 Classification

This Section will introduce methods for classification - *i.e.* where we want to separate our data into two or more classes. The most obvious way to do this is to create a discriminant function, and as such two methods will be introduced for creating such functions: Fisher Discriminant Analysis (FDA) and the margin-based approach of the Support Vector Machine (SVM). Following on from this two further algorithms will be presented which are based on the notion of *boosting* - Adaptive Boosting (AdaBoost) and Linear Programming Boosting (LPBoost) - and show how they are related to the margin maximisation principle of the SVM but also in the case of LPBoost to the LASSO approach described earlier.

Preliminaries

Assume we have a sample S containing examples $\mathbf{x} \in \mathbb{R}^n$ and labels $y \in \{-1, 1\}$. As before let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)'$ be the input vectors stored in matrix $\mathbf{X}$ as row vectors, and $\mathbf{y} = (y_1, \ldots, \mathbf{y}_m)'$ be a vector of outputs, where $'$ denote the transpose of vectors or matrices. For simplicity it will be assumed that the examples are already projected into the kernel defined feature space, so that the kernel matrix $\mathbf{K}$ has entries $\mathbf{K}[i, j] = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$.

2.1.9 Loss functions for classification

Before going on to give specific examples of learning algorithms for classification, as with the regression case it is worth introducing the different loss functions that are commonly used for classification. Again there is a focus on convex functions, as these lead to optimisation problems that can (in general) be solved exactly. Perhaps the simplest loss function for classification is the *zero-one* loss, defined as,

$$\mathcal{L} = \begin{cases} 0 & \text{if} \quad y_i = \text{sgn}(f(x_i)) \\ 1 & \text{otherwise.} \end{cases} \tag{2.29}$$

If the output of the classifier can be considered a confidence level, it may make sense to penalise larger errors more. A simple modification of the zero-one loss leads to the *hinge* loss,

$$\mathcal{L} = \begin{cases} 0 & \text{if} \quad y_i f(x_i) \geq 1 \\ 1 - y_i f(x_i) & \text{otherwise} \end{cases} \tag{2.30}$$

where $f(x_i) \in \mathbb{R}$. This in turn closely resembles the *logistic* loss, defined as

$$\mathcal{L} = \log\left(1 + \exp\left(-y_i f(x_i)\right)\right). \tag{2.31}$$

The square loss, which is closely related to the square loss for regression, and is defined as,

$$\mathcal{L} = (1 - y_i f(x_i))^2. \tag{2.32}$$

Finally, the linear loss, which relates to a Laplace noise model as it did for regression, is defined as,

$$\mathcal{L} = |1 - y_i f(x_i)| \,. \tag{2.33}$$

The relations between these loss function can be seen graphically in Figure 2.4. These loss functions

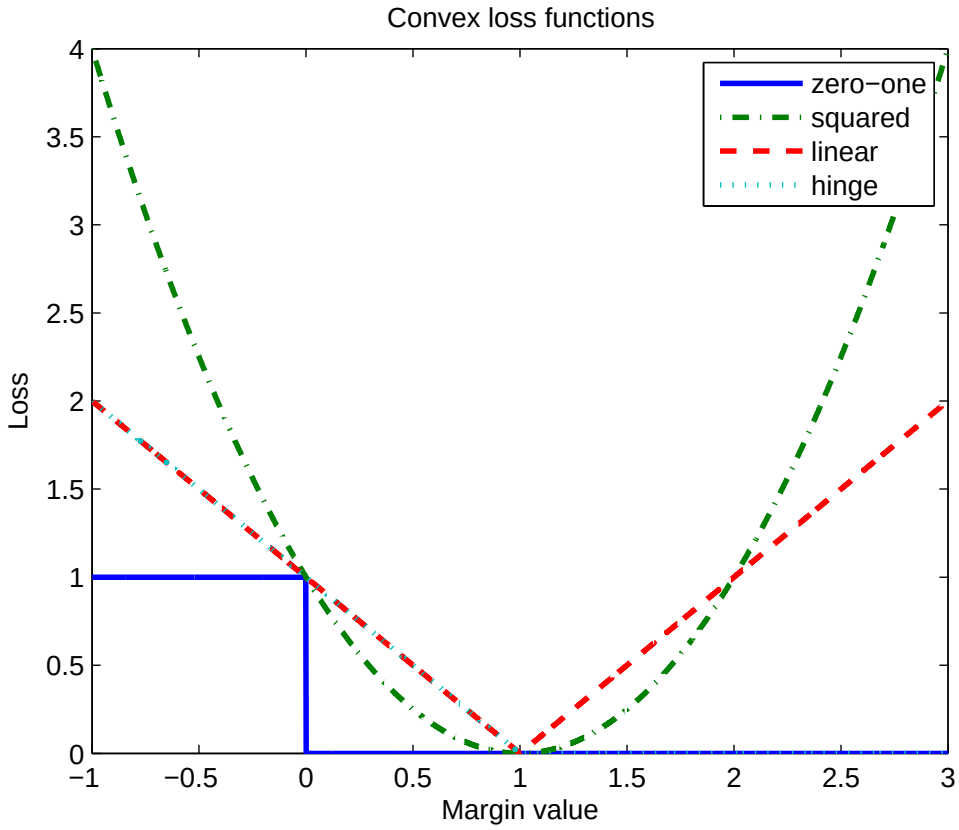


Figure 2.4: *Some examples of convex loss functions used in classification. Note that the hinge loss follows the linear loss for margin values less than 1, and is zero otherwise. Also note that the hinge loss is a convex upper bound on the zero-one loss.*

will play an important role in the rest of the discussion on classification. I will introduce FDA and its kernel equivalent, before showing how this can be cast as a convex optimisation problem using the square loss or the logistic loss.

Fisher Discriminant Analysis

We first review Kernel Fisher Discriminant Analysis (KFDA) in the form given by [3]. The Fisher discriminant chooses $\mathbf{w}$ to solve the following optimisation problem

$$\max_{\mathbf{w}} \frac{\mathbf{w}\mathbf{X}'\mathbf{y}\mathbf{y}'\mathbf{X}\mathbf{w}}{\mathbf{w}'\mathbf{X}'\mathbf{B}\mathbf{X}\mathbf{w}} \tag{2.34}$$

where $\mathbf{B}$ is a matrix incorporating the label information and the balance of the dataset as follows:

$$\mathbf{B} = \mathbf{D} - \mathbf{C}^+ - \mathbf{C}^-$$

where $\mathbf{D}$ is a diagonal matrix with entries

$$\mathbf{D}_{ii} = \begin{cases} 2m^-/m & \text{if } y_i = +1 \\ 2m^+/m & \text{if } y_i = -1 \end{cases}$$

and $\mathbf{C}^+$ and $\mathbf{C}^-$ are given by

$$\mathbf{C}^+_{ij} = \begin{cases} 2m^-/(mm^+) & \text{if } y_i = +1 = y_j \\ 0 & \text{otherwise} \end{cases}$$
$$\mathbf{C}^-_{ij} = \begin{cases} 2m^+/(mm^-) & \text{if } y_i = -1 = y_j \\ 0 & \text{otherwise} \end{cases}$$

Note that for balanced datasets $\mathbf{B}$ will be close to the identity matrix $\mathbf{I}$. The motivation for this choice is that the direction chosen maximises the separation of the means of each class scaled by the variances in that direction.

To solve this problem in the kernel defined feature space $\mathcal{F}$ we first need to show that there exists a linear expansion $\mathbf{w} = \sum_{i=1}^m \boldsymbol{\alpha}_i \mathbf{x}_i$ of the primal weight vector $\mathbf{w}$ [34, 3]. This leads to the following optimisation problem:

$$\begin{aligned} \rho &= \max_{\boldsymbol{\alpha}} \frac{\boldsymbol{\alpha}'\mathbf{X}\mathbf{X}'\mathbf{y}\mathbf{y}'\mathbf{X}\mathbf{X}'\boldsymbol{\alpha}}{\boldsymbol{\alpha}'\mathbf{X}\mathbf{X}'\mathbf{B}\mathbf{X}\mathbf{X}'\boldsymbol{\alpha}} \qquad (2.35) \\ &= \max_{\boldsymbol{\alpha}} \frac{\boldsymbol{\alpha}'\mathbf{K}\mathbf{y}\mathbf{y}'\mathbf{K}\boldsymbol{\alpha}}{\boldsymbol{\alpha}'\mathbf{K}\mathbf{B}\mathbf{K}\boldsymbol{\alpha}} \\ &= \max_{\boldsymbol{\alpha}} \frac{\boldsymbol{\alpha}'\mathbf{Q}\boldsymbol{\alpha}}{\boldsymbol{\alpha}'\mathbf{K}\mathbf{R}\boldsymbol{\alpha}} \qquad (2.36) \end{aligned}$$

where $\mathbf{Q} = \mathbf{K}\mathbf{y}\mathbf{y}'\mathbf{K}$ and $\mathbf{R} = \mathbf{B}\mathbf{K}$. The bias term b must be calculated separately, and there is no fixed way to do this. The most common method is to adjust b such that the decision boundary bisects the line joining the two centres of mass,

$$\begin{aligned} b &= -0.5\mathbf{y}'\mathbf{X}\mathbf{w} \\ &= -0.5\mathbf{y}'\mathbf{K}\boldsymbol{\alpha} \qquad (2.37) \end{aligned}$$

The classification function for KFDA is then,

$$\begin{aligned} f(\mathbf{x}_i) &= \text{sgn}(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \\ &= \text{sgn}(\mathbf{K}[:, i]'\boldsymbol{\alpha} + b), \qquad (2.38) \end{aligned}$$

by substituting $\mathbf{w} = \mathbf{X}'\boldsymbol{\alpha}$. There are several ways in which the optimisation problem (2.36) can then

be solved. Some algebra shows that it can be solved as the generalised eigenproblem $\mathbf{Q}\boldsymbol{\alpha} = \lambda\mathbf{K}\mathbf{R}$, by selecting the $\boldsymbol{\alpha}$ corresponding to the largest generalised eigenvalue λ, or in closed form as given by [3], $\boldsymbol{\alpha} = \mathbf{R}^{-1}\mathbf{y}$. Note that $\mathbf{R}$ is likely to be singular, or at best ill-conditioned, and so a regularised solution is obtained by substituting $\mathbf{R} = \mathbf{R} + \mu\mathbf{I}$, where μ is a regularisation constant. This is equivalent to imposing an l_2 penalty on the primal weight vector.

However, it has been shown [35, 36] that it is possible to exploit the structure of (2.36) to formulate KFDA as a quadratic program. This is reviewed below.

Convex Fisher Discriminant Analysis

First note that any multiple of $\boldsymbol{\alpha}$ is also a solution to (2.36). One can further use the observation that the matrix $\mathbf{Q}$ is rank one. This means that $\boldsymbol{\alpha}'\mathbf{K}\mathbf{y}$ can be fixed to any non-zero value, *e.g.* 2. By minimising the denominator, the following quadratic programme results,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}'\mathbf{K}\mathbf{R}\boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{\alpha}'\mathbf{K}\mathbf{y} = 2. \end{aligned} \tag{2.39}$$

Casting the optimisation problem (2.36) as the convex optimisation problem (2.39) gives several advantages. Firstly, for large sample size m, solving the eigenproblem is very costly due to the size of $\mathbf{Q}$ and $\mathbf{R}$. The convex formulation also avoids inverting $\mathbf{R}$ in the closed form solution which can be unstable. It is also possible to introduce sparsity into the $\boldsymbol{\alpha}$ solutions through the use of a different regularisation operator. Finally, it will enable the extension of the formulation naturally to multiple views, which is not easily done otherwise (see Section 3.5.2 in the following Chapter). However the unintuitive matrix $\mathbf{B}$ still remains in this formulation. Using the fact that KFDA minimises the variance of the data along the projection, whilst maximising the separation of the classes, it is possible to proceed by characterising the variance within a vector of slack variables $\boldsymbol{\xi} \in \mathbb{R}^n$. The variance can then be directly minimised as follows,

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \quad & \mathcal{L}(\boldsymbol{\xi}) + \mu\mathcal{P}(\boldsymbol{\alpha}) \\ \text{s.t.} \quad & \mathbf{K}\boldsymbol{\alpha} + \mathbf{1}b = \mathbf{y} + \boldsymbol{\xi} \\ & \boldsymbol{\xi}'\mathbf{e}^c = 0 \ \text{ for } \ c = 1, 2, \end{aligned} \tag{2.40}$$

where

$$\mathbf{e}_i^c = \begin{cases} 1 & \text{if } y_i = c \\ 0 & \text{otherwise.} \end{cases}$$

$\mathcal{L}(\cdot)$, $\mathcal{P}(\cdot)$ are the loss function and regularisation functions respectively as follows,

$$\mathcal{L}(\boldsymbol{\xi}) = \|\boldsymbol{\xi}\|_2^2, \tag{2.41}$$

$$\mathcal{P}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha}; \tag{2.42}$$

where: the first constraint forces the outputs onto the class labels whilst minimising their variance; the second constraint ensures that the label mean for each class is the label for that class, *i.e.* for ± 1 labels, and the average distance between the classes is two. It has been shown by [35] that any optimal solution $\boldsymbol{\alpha}$ of (2.40) is also a solution of (2.39). Note that now the bias term is explicitly in the optimisation, and therefore does not need to be calculated separately. The formulation (2.40) has appealing properties that will be used later.

2.1.10 Maximum Margin classification

Geometrically speaking, a maximum-margin hyperplane is a hyperplane that separates two sets of points such that it is equidistant from the closest point in each set and is perpendicular to the line joining the two points. In ML, the concept of large margins encompasses many different approaches to the classification of data from examples, including boosting, mathematical programming, neural networks, and SVM. The key fact is that it is the margin (which can be viewed as a confidence level) of a classification rather than a raw training error that is used when training a classifier [37]. This is known as the *hard margin SVM*, in which the margin γ is maximised as follows,

$$\begin{aligned} \min_{\mathbf{w},b,\gamma} \quad & -\gamma \\ \text{s.t.} \quad & y_i\left(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b\right) \geq \gamma, \quad i = 1, \ldots, m \\ & \|\mathbf{w}\|_2^2 = 1. \end{aligned} \tag{2.43}$$

Note that this is equivalent to using the hinge loss defined in Equation (2.30). Cortes and Vapnik [38] modified the maximum margin idea (also known as hard margin) to allow for mislabeled examples. In the absence of a hyperplane that can split the positive and negative examples, the soft margin method chooses a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. The method introduces slack variables, ξ_i, which measure the degree of misclassification of the point x_i. The objective function is then increased by a function which penalises non-zero ξ_i, and the optimisation becomes a trade off between a large margin and a small error penalty. The *2-norm soft margin SVM* is defined as the following optimisation problem

$$\begin{aligned} \min_{\mathbf{w},b,\gamma,\boldsymbol{\xi}} \quad & -\gamma + C\,\|\boldsymbol{\xi}\|_2^2 \\ \text{s.t.} \quad & y_i\left(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b\right) \geq \gamma - \xi_i, \quad i = 1, \ldots, m \\ & \|\mathbf{w}\|_2^2 = 1 \end{aligned} \tag{2.44}$$

where the parameter C controls the trade-off between maximising the margin and the size of the slack variables. The resulting algorithm is robust to noise in the data but not sparse in its solutions. In order to enforce sparsity, the ℓ_1 norm is used once again, giving the *1-norm soft margin SVM*,

$$\begin{aligned} \min_{\mathbf{w},b,\gamma,\boldsymbol{\xi}} \quad & -\gamma + C\left\|\boldsymbol{\xi}\right\|_1 \\ \text{s.t.} \quad & y_i\left(\langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b\right) \geq \gamma - \xi_i, \;\; i = 1, \ldots, m \\ & \left\|\mathbf{w}\right\|_2^2 = 1. \\ & \xi_i \geq 0, \;\; i = 1, \ldots, m. \end{aligned} \tag{2.45}$$

The dual of this optimisation problem can then be derived, giving us the kernel formulation,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \kappa(x_i, x_j) \\ \text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0, \\ & \sum_{i=1}^{m} \alpha_i = 1, \text{and} \\ & 0 \leq \alpha_i \leq C, \;\; i = 1, \ldots, m \end{aligned} \tag{2.46}$$

The SVM in this form can be solved by quadratic programming, or alternatively via iterative methods such as the Sequential Minimal Optimisation (SMO) algorithm [39].

2.1.11 Boosting

The term boosting describes any meta-algorithm for performing supervised learning, in which a set of "weak learners" create a single "strong learner". A weak learner is defined to be a classifier which is only slightly correlated with the true classification (*i.e.* slightly better than chance). By contrast, a strong learner is strongly correlated with the true classification [40].

Boosting algorithms are typically iterative, incrementally adding weak learners to a final strong learner. At every iteration, a weak learner learns the training data with respect to a distribution. The weak learner is then added to the current strong learner. This is typically done by weighting the weak learner in some manner, which is typically related to the weak learner's accuracy. After the weak learner is added to the strong learner, the data is reweighted: examples that are misclassified gain weight and examples that are classified correctly lose weight. Thus, future weak learners will focus more on the examples that previous weak learners misclassified.

Adaboost

AdaBoost is the best known example of a boosting algorithm [41]. Without *a-priori* knowledge, small decision trees, or decision stumps (decision trees with two leaves) are often used. The algorithm works

by iteratively adding in the weak learner that minimises the error with respect to the distribution D_t at step t over the weak learners,

$$h(t) = \arg\min_{h_j \in \mathcal{H}} \epsilon_t = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(\mathbf{x}_i)], \tag{2.47}$$

and then updating the distribution by using the weighted error rate of the classifier h_j,

$$\alpha_t = \frac{1}{2} \log \frac{1 - \epsilon_t}{\epsilon_t} \tag{2.48}$$

as follows,

$$D_{t+1}(i) = \frac{D_t(i) \exp\left(-\alpha_i y_i h_t(\mathbf{x}_i)\right)}{Z} \tag{2.49}$$

where Z is a normalisation constant to ensure that $\sum_{i=1}^{m} D_{t+1}(i) = 1$.

The paper [42] describes how the original [41] AdaBoost methods can be extended to the multiclass case[4]. One of the approaches taken, known as AdaBoost.MH, uses the Hamming loss of the hypotheses generated from ℓ orthogonal binary classification problems. The Hamming loss can be regarded as an average of the error rate h on these ℓ binary problems. Formally, for each weak hypothesis $h : \mathbf{X} \to 2^{\mathbf{Y}}$, and with respect to a distribution D, the loss is

$$\frac{1}{Z} \mathbb{E}_{(\mathbf{x},\mathbf{Y}) \sim D} \left[|h(\mathbf{x}) \Delta \mathbf{Y}| \right], \tag{2.50}$$

where Δ denotes the symmetric difference, and the leading $1/Z$ ensures that values lie in [0,1].

The resulting algorithm, called AdaBoost.MH, maintains a distribution over examples i and labels ℓ. On round t, the weak learner accepts such a distribution D_t and the training set, and generates a weak hypothesis $h_t : \mathbf{X} \times \mathbf{Y} \to \mathbb{R}$. This reduction leads to the choice of final hypothesis, which is

$$H(\mathbf{x}, \ell) = \operatorname{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}, \ell) \right). \tag{2.51}$$

The algorithm for AdaBoost.MH is given in Algorithm 1,

Theorem 2.1.1. *The reduction used to derive this algorithm implies a bound on the Hamming loss of the final hypothesis:*

$$\mathbb{E}(H) \leq \sum_{t=1}^{T} Z_t \tag{2.52}$$

In the binary classification problem, the goal is to minimise

$$Z_t = \sum_{i,\ell} D_t(i, \ell) \exp(-\alpha_t Y_{\{i,\ell\}} h_t(\mathbf{x}_i, \ell)) \tag{2.53}$$

[4]The authors also consider the more general multi-label case in which a single example may belong to any number of classes.

Algorithm 1 AdaBoost.MH: A multiclass version of AdaBoost based on Hamming Loss

Given training examples $(\mathbf{x}_1, Y_1), \ldots (\mathbf{x}_m, Y_m), Y_i \in \{+1, -1\}^{\ell}$, number of iterations T
Initialise $D_0(i, \ell) = \frac{1}{m}T$
for $t = 1 \ldots T$ **do**
pass distribution D_t to weak learner
get weak hypothesis $h_t : \mathbf{X} \times \mathbf{Y} \to \mathbb{R}$
choose α_t (based on performance of h_t)
update

$$D_{t+1}(i, \ell) = D_t(i, \ell) \exp(-\alpha_t Y_{\{i,\ell\}} h_t(\mathbf{x}_i, \ell))/Z_t$$

where Z_t is a normalisation factor chosen so that D_{t+1} will be a distribution
end for
Output final hypothesis: $H(\mathbf{x}, \ell) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}, \ell))$

on each round, where $i = 1 \ldots m$ and $\ell = 1 \ldots k$ (m is the number of examples and k is the number of classes). Since each h_t is required to be in the range $-1, +1$, each α_t is chosen as follows,

$$\alpha_t = \frac{1}{2} \log \left(\frac{1 + r_t}{1 - r_t} \right) \tag{2.54}$$

where

$$r_t = \sum_{i,\ell} D_t(i, \ell) Y_{\{i,\ell\}} h_t(\mathbf{x}_i, \ell) \tag{2.55}$$

This gives

$$Z_t = \sqrt{1 - r_t^2} \tag{2.56}$$

and the goal of the weak learner becomes maximisation of $|r_t|$. The quantity $(1 - r_t)/2$ is the weighted Hamming loss with respect to D_t.

To relate AdaBoost to the previous discussion of loss functions in Section 2.1.9, the statistical viewpoint is that boosting can be seen as the minimisation of a convex loss function over a convex set of functions [43]. Specifically, the loss being minimized is the exponential loss

$$\mathcal{L} = \sum_{i=1}^{m} \exp(-y_i H(\mathbf{x}_i)) \tag{2.57}$$

where $H(\mathbf{x}_i) = \sum_{t=1}^{T} f(\mathbf{x}_i)$ is the final hypothesis.

Linear Programming Boosting (LPBoost)

Referring back to the 1-norm soft margin SVM in Equation (2.45), it is possible to perform the same optimisation using the weak hypothesis matrix $\mathbf{H}$, where $\mathbf{H} = \sum_i y_i h(x_i, \cdot)$, which is equivalent to

$\mathbf{y}'(\phi(\mathbf{x}) + b)$. This would result in the following optimisation (written in matrix form),

$$\begin{aligned} \min_{\mathbf{w},\gamma,\boldsymbol{\xi}} \quad & -\gamma + C\mathbf{1}'\boldsymbol{\xi} \\ \text{s.t.} \quad & \mathbf{H}\mathbf{w} \geq \gamma\mathbf{1} - \boldsymbol{\xi}, \\ & \|\mathbf{w}\|_2^2 = 1, \end{aligned} \tag{2.58}$$

where $\mathbf{1}$ is the vector of all ones. Since the number of weak learners in the matrix $\mathbf{H}$ is potentially very large, it is logical to enforce sparsity in the primal weight vector $\mathbf{w}$, which can be done by replacing the ℓ_2-norm constraint with an ℓ_1-norm constraint. This results in the following linear programme,

$$\begin{aligned} \min_{\mathbf{w},\gamma,\boldsymbol{\xi}} \quad & -\gamma + C\mathbf{1}'\boldsymbol{\xi} \\ \text{s.t.} \quad & \mathbf{H}\mathbf{w} \geq \gamma\mathbf{1} - \boldsymbol{\xi}, \\ & \mathbf{1}'\mathbf{w} = 1, \\ & \mathbf{w} \geq \mathbf{0}, \;\; \boldsymbol{\xi} \geq \mathbf{0}. \end{aligned} \tag{2.59}$$

The dual of this optimisation can then be formulated as follows,

$$\begin{aligned} \min_{\boldsymbol{\alpha},\beta} \quad & \beta \\ \text{s.t.} \quad & \mathbf{H}'\boldsymbol{\alpha} \leq \beta\mathbf{1}, \\ & \mathbf{1}'\boldsymbol{\alpha} = 1, \\ & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \end{aligned} \tag{2.60}$$

with dual variables $\boldsymbol{\alpha}$ and β, and the box constraints on the $\boldsymbol{\alpha}$ variables are due to the primal slack variables $\boldsymbol{\xi}$.

The paper by [5] describes an efficient algorithm called LPBoost mimics a simplex based method known as column generation in order to solve the optimisation problem (2.60). The simplex algorithm is a method for finding the numerical solution of the linear programming problem, first introduced by George Dantzig [44]. A simplex is a polytope of $n + 1$ vertices in n dimensions: a polygon on a line, a pyramid on a plane, etc.

The column generation method involves formulating the problem as if all possible weak hypotheses had already been generated, with the resulting labels becoming the new feature space of the problem. The task that is solved by boosting is to construct a learning function within the output space that minimises misclassification error and maximises the (soft) margin. They prove that for the purposes of classification, minimising the 1-norm soft margin error function is equivalent to optimising a generalisation error bound. The linear programme is efficiently solved using a technique known as column generation. LPBoost has the advantages over gradient based methods (such as AdaBoost) that it converges in a finite number of iterations to a global solution that is optimal within the hypothesis space, and that these

solutions are very sparse.

The paper cites results that demonstrate that LPBoost performs competitively with AdaBoost on a variety of datasets. The authors also demonstrate that the algorithm is computationally tractable. For both small and large datasets, the computation of the weak learners outweighs the linear programme running time, which means that in general the time for LPBoost iterations are in the same order of magnitude as AdaBoost, though slightly higher.

Many linear programs are too large to consider all the variables explicitly. Since most of the variables will be zero in the optimal solution, only a subset of variables need to be considered. Column generation generates only variables which have the potential to improve the objective function (*i.e.* negative reduced cost). The problem being solved is split into two problems, known as the master problem and the subproblem. The master problem is the original problem with only a subset of variables, and the subproblem is a new problem created to identify a new variable. The objective function of the subproblem is the reduced cost of the new variable with respect to the current dual variables. LPBoost can be proved to converge in a finite number of iterations to a globally optimal solution within the hypothesis space. In the dual form the constraints are the weak learners.

The algorithm proceeds by adding a weak learner, and checking if the linear programme is solved. If not then the weak learner is found that violates the constraints the most. This process is repeated until the linear programme constraints are not violated, which leads to the global optimum solution. LPBoost iterations are typically slower than AdaBoost, but it converges much more quickly. The LPBoost algorithm is given in Algorithm 2.

Algorithm 2 LPBoost algorithm

Given training examples $(\mathbf{x}_1, y_1), \ldots (\mathbf{x}_m, y_m), y_i \in \{+1, -1\}$, upper limit on weights C
Initialise $\boldsymbol{\alpha} \leftarrow \frac{1}{m}\mathbf{1}$, $\mathbf{H} \leftarrow ()$
while $\mathbf{H}'\boldsymbol{\alpha} > \beta$ **do**
$h \leftarrow \max_{h \in \mathcal{H}} \sum_{i=1}^{m} y_i \alpha_i h_{i,\cdot}$
$H \leftarrow \binom{\mathbf{H}}{h}$
Update $\boldsymbol{\alpha}$: Solve Linear Programme:

$$\begin{aligned} \arg\min \quad & \beta \\ \text{s.t.} \quad & \mathbf{H}'\boldsymbol{\alpha} \leq \beta\mathbf{1}, \\ & 0 < \boldsymbol{\alpha} < C\mathbf{1}. \end{aligned}$$

end while
Set $\mathbf{w}$ to Lagrangian multipliers

Although at first the boosting methods described above seem rather disjoint from the convex methods described under the general loss minimisation and regularisation framework, there are in fact distinct similarities. If one considers that a general ML principle is to minimise the regularised empirical loss:

$$\min_{\boldsymbol{\alpha}} \mathcal{L} + \mathcal{P}(\boldsymbol{\alpha}), \tag{2.61}$$

it can be seen that in fact there is a direct relation between LPBoost and LASSO which both use ℓ_1

regularisation with differening loss functions (hinge loss and quadrtic loss respectively), and between regularised forms of AdaBoost[45] (exponential loss) and the SVM (hinge loss). We can also see the relation between KRR and the convex formulation of KFDA given in Section 2.1.9 where the differences are only in the constraints. See for example [46, 47, 48] for recent discussions of this issue.

2.1.12 Subspace Methods

In standard single view subspace learning, a parallel can be drawn between subspace projections that are independent of the label space, such as Principal Components Analysis (PCA), and those that incorporate label information, such as Fisher Discriminant Analysis (FDA). PCA searches for directions in the data that have largest variance and project the data onto a subset of these directions. In this way a lower dimensional representation of the data is obtained that captures most of the variance. PCA is an unsupervised technique and as such does not include label information of the data. For instance, given 2-dimensional data from two classes forming two long and thin clusters, such that the clusters are positioned in parallel and very closely together, the total variance ignoring the labels would be in the lengthwise direction of the clusters. For classification, this would be a poor projection, because the labels would be evenly mixed. A much more useful projection would be orthogonal to the clusters, *i.e.* in the direction of least overall variance, which would perfectly separate the two classes. We would then perform classification in this 1-dimensional space. FDA would find exactly this projection.

However if classification is not the goal, but instead the goal is to take a subset of the principal axes of the training data and project both the train and test data into the space spanned by this subset of eigenvectors, the PCA performs this projection by maximising the following criterion,

$$\begin{aligned} \max_{\mathbf{w}} \quad & \mathbf{w}'\boldsymbol{\Sigma}\mathbf{w}, \\ \text{s.t.} \quad & \|\mathbf{w}\|_2 = 1, \end{aligned} \tag{2.62}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the *centred* data - *i.e.* $\boldsymbol{\Sigma} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})$[5]. The dual form of PCA can be formed as follows,

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}'\mathbf{X}\mathbf{X}'\mathbf{X}\mathbf{X}'\boldsymbol{\alpha}, \\ \text{s.t.} \quad & \boldsymbol{\alpha}'\mathbf{X}\mathbf{X}'\boldsymbol{\alpha} = 1. \end{aligned} \tag{2.63}$$

Using again the kernel trick, the nonlinear version of PCA known as Kernel Principal Components Analysis (KPCA) [49] is defined as follows,

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}'\mathbf{K}^2\boldsymbol{\alpha}, \\ \text{s.t.} \quad & \boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} = 1. \end{aligned} \tag{2.64}$$

[5]The purpose of centering data (transforming data to z-scores) is to remove undesirable fluctuations. Part of the PCA solution is the minimisation of the sum of squared errors. Overall, the goal is to find the best affine linear subspace.

Each of these problems can be solved efficiently as eigenproblems.

2.1.13 Multi-view Learning

Canonical Correlation Analysis (CCA), introduced by Harold Hotelling in 1936 [50], is a method of correlating linear relationships between two sets of multidimensional variables. CCA makes use of two views of the same underlying semantic object to extract a common representation of the semantics. CCA can be viewed as finding basis vectors for two sets of variables such that the correlations between the projections onto these basis vectors $x_a = \mathbf{w}'_{\mathbf{a}}\phi_{\mathbf{a}}(\mathbf{x})$ and $x_b = \mathbf{w}'_{\mathbf{b}}\phi_{\mathbf{b}}(\mathbf{x})$ are mutually maximised. Defining the covariance between the two views as $\mathbf{\Sigma}_{ab}$ and the variance of the views as $\mathbf{\Sigma}_{aa}$ and $\mathbf{\Sigma}_{bb}$ respectively, we have the following optimisation problem,

$$\begin{aligned} \max_{\mathbf{w}_a,\mathbf{w}_b} \quad & \mathbf{w}'_a\mathbf{\Sigma}_{ab}\mathbf{w}_b \\ \text{s.t.} \quad & \mathbf{w}'_a\mathbf{\Sigma}_{aa}\mathbf{w}_a = 1, \\ & \mathbf{w}'_b\mathbf{\Sigma}_{bb}\mathbf{w}_b = 1. \end{aligned} \tag{2.65}$$

The major limitation of CCA is its linearity, but the method can be extended to find nonlinear relationships using a the kernel trick once again. Kernel Canonical Correlation Analysis (KCCA) is an implementation of this method that results in a nonlinear version of CCA. Each of the two views of the data are projected into distinct feature spaces such that $\mathbf{w}_a = \mathbf{X}'_a\boldsymbol{\alpha}_a$ and $\mathbf{w}_b = \mathbf{X}'_b\boldsymbol{\alpha}_b$, before performing CCA in the new feature space. The dual form of CCA is

$$\begin{aligned} \max_{\boldsymbol{\alpha}_a,\boldsymbol{\alpha}_b} \quad & \boldsymbol{\alpha}'_a\mathbf{X}_a\mathbf{X}'_a\mathbf{X}_b\mathbf{X}'_b\boldsymbol{\alpha}_b \\ \text{s.t.} \quad & \boldsymbol{\alpha}'_a\mathbf{X}_a\mathbf{X}'_a\mathbf{X}_a\mathbf{X}'_a\boldsymbol{\alpha}_a = 1, \\ & \boldsymbol{\alpha}'_b\mathbf{X}_b\mathbf{X}'_b\mathbf{X}_b\mathbf{X}'_b\boldsymbol{\alpha}_b = 1, \end{aligned} \tag{2.66}$$

(2.67)

which leads to the kernelised form, KCCA

$$\begin{aligned} \max_{\boldsymbol{\alpha}_a,\boldsymbol{\alpha}_b} \quad & \boldsymbol{\alpha}'_a\mathbf{K}_a\mathbf{K}_b\boldsymbol{\alpha}_b \\ \text{s.t.} \quad & \boldsymbol{\alpha}'_a\mathbf{K}_a^2\boldsymbol{\alpha}_a = 1, \\ & \boldsymbol{\alpha}'_b\mathbf{K}_b^2\boldsymbol{\alpha}_b = 1, \end{aligned} \tag{2.68}$$

where $\mathbf{K}_a$ and $\mathbf{K}_b$ are the kernel matrices of the two views.

There have been several successful experimental applications of KCCA on bilingual text corpora, firstly by [51] and later by [52]. In the latter study the authors compare the performance of KCCA with alternative retrieval method based on the Generalised Vector Space Model (GVSM), which aims to capture correlations between terms by looking at co-occurrence information. Their results show that

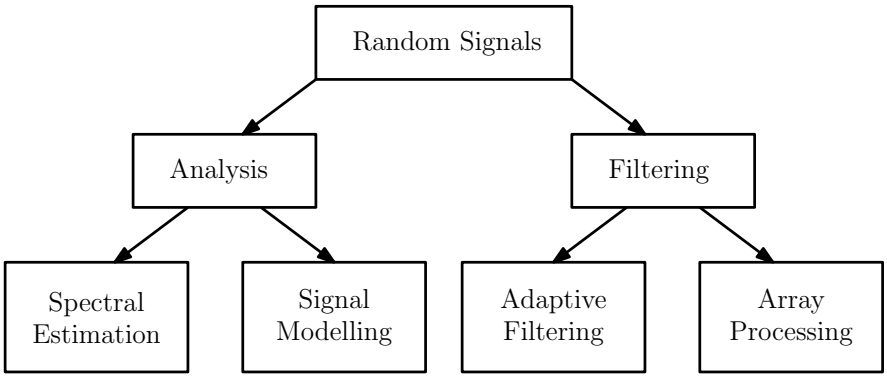


Figure 2.5: *Common tasks in Digital Signal Processing*

KCCA outperforms GVSM in both in content retrieval and in mate retrieval tasks.

Recent work [53] presents a novel method for solving CCA in a sparse convex framework using a greedy least squares approach, called Sparse Canonical Correlation Analysis (SCCA). Stability analysis using Rademacher Complexity is given for SCCA which provides a bound on the quality of the patterns found. The authors demonstrate on a paired English-Spanish corpus that the proposed method is able to outperform KCCA with a tighter bound.

2.2 Digital Signal Processing (DSP)

In this Section the focus moves to the principles underlying DSP. It will become clear that there are many links between ML and DSP, and that both fields are able to draw on each other to bring novel advances. For the sake of brevity, it will be assumed that the Analogue to Digital Conversion (ADC) process has already taken place, and as such all of the signals under consideration are discrete with equal time steps. All of the theory is able to deal with unequal time steps, but the analysis becomes more complicated. However some of the formulas used to describe quantities and operations will be given for continuous signals, as their presentation is more straightforward. Some common tasks in DSP are depicted in Figure 2.5. Within the scope of this thesis the primary concern is *signal analysis*, and hence spectral estimation and signal modelling. However many results can be carried over to filtering as well.

2.2.1 Bases, Frames, Dictionaries and Transforms

A frame of a vector space V with an inner product can be seen as a generalisation of the idea of a basis to sets which may be linearly dependent. More precisely, a frame is a set of elements of V which satisfy the following condition:

Frame condition: There exist two real numbers, A and B such that

$$0 < A \leq B < \infty,$$
$$A \left\|\mathbf{v}\right\|^2 \leq 0 \leq B \left\|\mathbf{v}\right\|^2.$$

Parseval's identity is a fundamental result on the summability of the Fourier series of a function. Geometrically, it is the Pythagorean theorem for inner-product spaces.

Theorem 2.2.1 (Parseval's Theorem [54])**.** *If $\{e_j : j \in J\}$ is an orthonormal basis of a Hilbert space H, then for every $x \in H$ the following equality holds:*

$$\|x\|^2 = \sum_{j \in J} |\{x, e_j\}|^2.$$

Although frames do not in general consist of orthonormal vectors, the frame representation of a vector may still satisfy Parseval's identity. The constants A, B are called the lower and upper frame bounds respectively. When $A = B$ the frame is a tight frame.

Fourier analysis represents any finite continuous energy function $f(t)$ as a sum of sinusoidal waves $\exp(i\omega t)$,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) \exp(i\omega t) d\omega. \tag{2.69}$$

The amplitude $\hat{f}(\omega)$ of each sinusoid is equal to its correlation with f, also called the *Fourier transform*,

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) \exp(-i\omega t) dt. \tag{2.70}$$

The more regular the function $f(t)$ is, the faster the decay of the amplitude $|\hat{f}(\omega)|$ as ω increases. If $f(t)$ is defined only over an interval, *e.g.* $[0, 1]$, the Fourier transform becomes a decomposition into an *orthonormal basis*: $\{\exp(i2\pi mt)\}_{m \in \mathbb{Z}}$ of $\mathbb{L}_2[0, 1]$ [6]. If the signal is uniformly regular, then the Fourier transform can represent the signal using very few nonzero coefficients. Hence this class of signal is said to be sparse in the Fourier basis. The wavelet basis was introduced by Haar [55] as an alternative way of decomposing signals into a set of coefficients on a basis. The Haar wavelet basis defines a sparse representation of piecewise regular signals, and has therefore received much attention from the image processing community. The piecewise constant function, or Haar *atom*, is defined as,

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 0.5 \\ -1 & \text{if } 0.5 \leq t < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.71}$$

An orthonormal basis on $\mathbb{L}_2$ can be formed by dilating and translating these atoms as follows,

$$\left\{ \Psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - 2^j n}{2^j} \right) \right\}_{j,n \in \mathbb{Z}^2} \tag{2.72}$$

Thus far all definitions have been for continuous signals. That is because a dictionary can be created through dilations and translations of the single function ψ, but dilations and translations are not defined for discrete signals. The transition from continuous to discrete time must be done with great care to

[6] $\mathbb{L}_2[0, 1]$ is the set of functions such that $\int_0^1 |f(t)| dt < \infty$

preserve important properties such as orthogonality.

The definition of a time-frequency dictionary $\Psi = \{\psi_\gamma\}_{\gamma\in\Gamma}$ is that it is composed of waveforms of unit norm ($\|\psi_\gamma\|_2 = 1$) which have a narrow spread in time (u) and frequency (σ^2).

Choice of the dictionary Ψ should, if possible, be based on knowledge of properties of the signal. One of the most common choices for a general class of real-world signals is the Gabor dictionary, as it can represent a wide range of smooth signals. The Chirp dictionary is a generalisation of the Gabor dictionary with an extra parameter (the chirp rate). Both of these will be described below, and empirical comparisons will be made between each method.

Gabor Dictionary

Gabor time-frequency atoms are scaled, translated and modulated Gaussian functions $g(t)$ (Gabor atoms) [56]. Without loss of generality, discrete real Gabor atoms will be considered, which are given by

$$g_{\gamma,\phi}(t) = \frac{1}{Z}\cdot g\left(\frac{t-u}{s}\right)\cdot\cos(\theta t+\phi) \tag{2.73}$$

where Z is a normalisation factor (to ensure that for each atom $\|g_{\gamma,\phi}\| = 1$), $\gamma_n = (s_n, u_n, \theta_n)$ denotes the series of parameters of the functions of the dictionary, and $g(t) = \exp^{-\pi t^2}$ is the Gaussian window.

Chirp Dictionary

Chirp atoms were introduced to deal with the nonstationary behavior of the instantaneous frequency of some signals, and shown to form an orthonormal basis [57]. In the present analysis only linear chirps are required for the empirical applications provided later. A real chirp atom is then given by

$$g_{\gamma,\phi,c}(t) = \frac{1}{Z}\cdot g\left(\frac{t-u}{s}\right)\cdot\cos(\theta(t-u)+\frac{c}{2}(t-u)^2+\phi) \tag{2.74}$$

where c is the chirp rate and all other parameters are the same as for the real Gabor atom. The chirp atom has an instantaneous frequency $\omega(t) = \theta + c(t-u)$ that varies linearly with time.

Dyadic Sampling

A sampling pattern is dyadic if the daughter wavelets are generated by dilating the mother wavelet as in Equation 2.72 by 2^j and translating it by $k2^j$, *i.e.* $s = 2^j,\ u = k2^j$. Dyadic sampling is optimal because the space variable is sampled at the Nyquist rate for any given frequency. The dictionary is then defined as,

$$\Psi_{j,\Delta} = \{\psi_n = g_{\gamma,\phi}(t)\}_{0\le q<\Delta N2^{-j}, 0\le k<\Delta 2^j}\,, \tag{2.75}$$

where $g_{\gamma,\phi}(t)$ is the discrete Gabor atom or Chirp atom as defined above in Equations 2.73 and 2.74 respectively. An example of this sampling scheme is given in Table 2.2 for a signal of length 128 and dilation factor $\Delta = 2$.

j	2^j	2^-j	$N2^-j$	q	k
2	4	$1/2$	64	0:128	0:8
3	8	$1/4$	32	0:64	0:16
4	16	$1/8$	16	0:32	0:32
5	32	$1/16$	8	0:16	0:64
6	64	$1/32$	4	0:8	0:128

Table 2.2: *Example of the dyadic sampling scheme for a signal of length 128 and* $\Delta = 2$.

2.2.2 Sparse and Redundant Signals

As with ML, finding sparse solutions to underdetermined inverse problems is a fundamental challenge encountered in a wide range of DSP applications, from signal acquisition to source separation. Recent theoretical advances in our understanding of this problem have further increased interest in their application to various domains. In many areas, such as for example medical imaging or geophysical data acquisition, it is necessary to find sparse solutions to very large underdetermined inverse problems that therefore require fast methods. The decomposition of a signal $\mathbf{x}$ into a dictionary $\Psi \in \mathbb{R}^{n \times p}$ solves the following problem,

$$\Psi\alpha = \mathbf{x}. \tag{2.76}$$

If the dictionary is a tight frame, the simplest solution to this would then be the inverse problem

$$\alpha = \Psi^{-1}\mathbf{y}. \tag{2.77}$$

If additionally all of the atoms of the dictionary are orthonormal then $\Psi^{-1} = \Psi'$. However in most practical applications, the dictionary is designed to be *overcomplete - i.e.* $p \gg n$, and hence there are many possible solutions to this inverse problem. The *method of frames* [58] uses the minimum ℓ_2-norm solution (also called minimum energy or minimum length solution):

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \|\boldsymbol{\alpha}\|_2^2 \\ \text{s.t.} \quad & \mathbf{x} = \Psi\boldsymbol{\alpha}. \end{aligned} \tag{2.78}$$

It can be seen that this is equivalent to the least squares solution to the regression problem as defined in Equation 2.11, and that it likewise has a closed form solution $\boldsymbol{\alpha} = (\Psi'\Psi')^{-1}\Psi'\mathbf{x}$. However, the unknown (not sampled) coefficients seldom have zero energy. A more attractive solution would be minimising the ℓ_0-norm, or equivalently maximising the number of zero coefficients in the new basis:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \|\boldsymbol{\alpha}\|_0 \\ \text{s.t.} \quad & \mathbf{x} = \Psi\boldsymbol{\alpha}. \end{aligned} \tag{2.79}$$

However, this is NP-hard (it contains the subset-sum problem), and so is computationally infeasible for all but the smallest datasets. Thus, following [59], the ℓ_1-norm, is usually what is minimised. This leads to comparable results to using the ℓ_0-norm, often yielding results with many coefficients being zero,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \|\boldsymbol{\alpha}\|_1 \\ \text{s.t.} \quad & \mathbf{x} = \Psi\boldsymbol{\alpha}. \end{aligned} \tag{2.80}$$

This method is known as Basis Pursuit (BP) [31]. Note that if we bring the constraint into the optimisation using a Lagrange multiplier, this is in fact equivalent to the LASSO problem for regression that was defined earlier in Equation 2.27.

2.2.3 Greedy Methods for Sparse Estimation

There are other ways to approximate the ℓ_0 solution, such as by greedy iterative methods. These include (but are not limited to) Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP), [56]), Polytope Faces Pursuit (PFP) [60, 61] and more recently with non-convex penalties and Difference of Convex (DC) programming [62, 63]. There are also many modifications of each of these methods, including stepwise approaches that bring more than one basis into the solution at each step. For brevity these will not be covered here, but offer an interesting path for possible modifications of algorithms based on these methods.

Matching Pursuit and Orthogonal Matching Pursuit

Matching Pursuit (MP) was proposed as an attempt at finding a sparse set of basis functions (atoms) for a signal from a given dictionary [56]. In many ways this problem can be interpreted as a sparse version of least squares regression when the Orthogonal Matching Pursuit (OMP) version is applied [64]. In OMP each time a dictionary atom is chosen, the remaining weight vectors are projected into a space orthogonal to those chosen such that future atoms are only considered from a set far from those already picked. To link back to ML once again, as with Kernel Basis Pursuit (KBP), Kernel Matching Pursuit (KMP) [65] has been proposed as the kernel counterpart of MP.

Given a signal f and dictionary $\Psi = \{\psi_p\}_{p\in\Gamma}, |\Gamma| \gg n$ of atoms with unit norm, MP begins by initialising the residue $\mathbf{r}_0 = f$, and then iterates by projecting the function f onto all of the vectors $\psi_p \in \Psi$ and computing their residue $\mathbf{r}$,

$$f = \alpha_p\psi_p + \mathbf{r}, \quad p = 1, \ldots, |\Gamma|, \tag{2.81}$$

implying that $\alpha_p = \langle f, \psi_p\rangle$. The atom with the maximum inner product $\langle\psi_t, \psi_i\rangle$ is then selected along

with its weight α_i.

$$\begin{aligned} i &= \arg\max_{p\in\Gamma} \alpha_p, \\ \alpha_t &= \alpha_i, \\ \psi_t &= \psi_i. \end{aligned} \tag{2.82}$$

The residue is then updated as follows,

$$\mathbf{r}_{t+1} = \mathbf{r}_t - \alpha_t \psi_t \tag{2.83}$$

The final solution is then given by $\sum_{t=1}^{T} \alpha_t \psi_t$, which can be shown to converge to the optimal solution given that the dictionary forms a tight frame [56]. MP approximations are improved by orthogonalising the directions of the projection using a Gram-Schmidt procedure [66]. The resulting pursuit then converges within a finite number of iterations T instead of in the limit, which balances the fact that the orthogonalisation is expensive to compute. The Gram-Schmidt algorithm orthogonalises ψ_p with respect to $\{\psi_q\}_{q:p\notin P}$ as follows,

$$\hat{\psi_p} = \psi_p - \sum_{q:p\notin P} \frac{\psi_p, \psi_q}{\|\psi_q\|_2^2} \psi_q. \tag{2.84}$$

The orthogonalised version of the atom $\hat{\psi_p}$ is then used for calculation of the residue. The next Section describes a further modification of the MP/OMP framework that makes use of the geometry of the solution space.

Polytope Faces Pursuit

The algorithm Polytope Faces Pursuit (PFP) [61] is based on the geometry of the polar polytope [60] where at each step a basis function is chosen by finding the maximal vertex using a path-following method.

Further investigation of the criteria under which ℓ_0/ℓ_1 equivalence holds led to consideration of the $d-$dimensional *polytope* (the $d-$dimensional generalisation of a polygon) [60]. Using this geometric interpretation, a greedy algorithm called PFP has been proposed [67] which adopts a path-following approach through the relative interior faces of the polar polytope. The first step is to convert (2.80) into its standard form,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \|\tilde{\boldsymbol{\alpha}}\|_1 \\ \text{s.t.} \quad & \mathbf{x} = \tilde{\Psi}\tilde{\boldsymbol{\alpha}}, \quad \tilde{\boldsymbol{\alpha}} \geq 0, \end{aligned} \tag{2.85}$$

where $\tilde{\Psi} = [\Psi, -\Psi]$ and $\tilde{\boldsymbol{\alpha}}$ has $2m$ nonnegative components, with the standard weight vector recoverable

by $\alpha_i = \tilde{\boldsymbol{\alpha}}_i - \tilde{\boldsymbol{\alpha}}_{i+m}$ [68]. The corresponding dual of this linear program is,

$$\begin{aligned} \max_{\mathbf{c}} \quad & \mathbf{y}'\mathbf{c} \\ \text{s.t.} \quad & \tilde{\Psi}'\mathbf{c} \leq 1 \end{aligned} \tag{2.86}$$

which has an optimal dual weight vector $\mathbf{c}$ which corresponds to the optimum $\boldsymbol{\alpha}$ of the primal formulation. At each step the approach to the solution of this problem is to identify the optimal vertex which is the maximiser of $\mathbf{x}'\mathbf{c}$, which is similar to the way in which OMP builds up its solution. However the difference is that at each step, the path is constrained on the polytope face F given by the vertex of the previous step. This is achieved by projecting $\mathbf{x}$ into a subspace parallel to F to give $\mathbf{r} = (\mathbf{I} - \mathbf{Q})\mathbf{x}$ where $\mathbf{Q} = \frac{\tilde{\Psi}_i \tilde{\Psi}_i'}{\|\Psi_i\|_2^2}$. Since $\boldsymbol{\alpha} = \tilde{\Psi}_i^\dagger \mathbf{x}$ (where $\mathbf{A}^\dagger$ is defined as the Moore-Penrose pseudo-inverse of a matrix $\mathbf{A}$[7]), and $\hat{\mathbf{x}} = \tilde{\Psi}_i \boldsymbol{\alpha}$, it follows that $\mathbf{r} = \mathbf{x} - \tilde{\Psi}_i \boldsymbol{\alpha} = \mathbf{x} - \hat{\mathbf{x}}$ meaning that $\mathbf{r}$ is the residual from the approximation at step i. The second step, which is where the main difference between OMP and PFP arises, involves projecting within the face F that has just been found, rather than from the origin. This is done by projecting along the residual $\mathbf{r}$. Therefore to find the next face at each step, the maximum *scaled* correlation is found

$$\mathbf{i}_i = \arg\max_{i \notin \mathbf{i}} \frac{\tilde{\Psi}_i' \mathbf{r}}{(1 - \tilde{\Psi}_i' \mathbf{c})} \tag{2.87}$$

where bases are only considered such that $\tilde{\Psi}_i' \mathbf{r} > 0$.

PFP then proceeds by removing any constraints that violate the condition that $\tilde{\boldsymbol{\alpha}}$ contains any negative entries. This is achieved by finding $j \in \mathbf{i}$ such that $\tilde{\boldsymbol{\alpha}}_j < 0$, removing j from $\mathbf{i}$ and removing the face from the current solution. $\tilde{\boldsymbol{\alpha}}$ is then recalculated, and the algorithm continues until $\boldsymbol{\alpha}_j \geq 0, \forall\, j$.

The algorithmic complexity is of a similar order to OMP whilst being able to solve problems known to be hard for MP and OMP.

2.2.4 Compressed Sensing (CS)

In this Section, some of the theory of Compressed Sensing (CS) (also known as compressive sampling and sparse sampling) will be reviewed. CS is a technique that allows signals to be acquired or reconstructed sparsely, by using prior knowledge that the signal is sparse in a given basis [59, 69]. The main result is that signals can be reconstructed exactly even with data deemed insufficient by the Nyquist-Shannon criterion[8]. Formally, given a signal $\mathbf{x} \in \mathbb{R}^n$ and a dictionary $\Psi \in \mathbb{R}^{n \times d}$ which forms an orthonormal basis, $\mathbf{x}$ is said to be sparse if $\mathbf{x}$ can be represented as a linear combination of k atoms from Ψ, *i.e.* $x = \sum_{i=1}^{k} \alpha_i \Psi_{.,i}$ where $k \ll d$. According to the CS theory it is possible to construct a measurement matrix $\Phi \in \mathbb{R}^{m \times n}$ with $m \ll n$, and perform stable reconstructions of the signal from measurements $\mathbf{y} = \Phi\mathbf{x}$ if and only if the measurement matrix is incoherent with the dictionary, *i.e.* the

[7] Note that if Ψ_i forms a tight frame then $\Psi_i^\dagger = \Psi_i'$ - *i.e.* the inverse is equal to the transpose.

[8] The Nyquist-Shannon sampling theorem states that if a function $f(t)$ contains no frequencies higher than $B\ Hz$, it is completely determined by giving its ordinates at a series of points spaced $1/(2B)$ seconds apart

sensing waveforms have an extremely dense representation in Φ[9]. Ordinarily, the problem of reconstructing $\mathbf{x}$ from $\mathbf{y}$ would be severely undetermined.

Estimating a sparsely represented function from a set of training examples is a classical problem in regression. Fortunately the methods used for sparse regression can be directly applied to CS. Again, beginning with the ℓ_0-minimisation,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \|\boldsymbol{\alpha}\|_0 \\ \text{s.t.} \quad & \mathbf{y} = \Phi\Psi\boldsymbol{\alpha}. \end{aligned} \tag{2.88}$$

Finding this ℓ_0 solution is known to be $NP-$hard. However the equivalent ℓ_1 optimisation problem

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \|\boldsymbol{\alpha}\|_0 \\ \text{s.t.} \quad & \mathbf{y} = \Phi\Psi\boldsymbol{\alpha}. \end{aligned} \tag{2.89}$$

is a convex optimisation problem and can be solved using general purpose solvers. As before, this can be reformulated such that it directly minimises the regression loss, as with the LASSO [30], which is given by

$$\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \Phi\Psi\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 , \tag{2.90}$$

i.e. a form of ℓ_1-penalised least squares. This can then be solved with the Least Angle Regression Solver (LARS) as before, or with greedy methods such as OMP or PFP.

2.2.5 Incoherence With Random Measurements

One major issue that has not been addressed is how to design the measurement matrix Ψ such that when sampled using this matrix, the signal will be sparse within the basis of the dictionary Φ. The CS theory states when certain conditions hold, namely that the functions $\psi_m \in \Psi$ cannot sparsely represent the elements of the basis $\phi_m \in \Phi$ (a condition known as incoherence of the two dictionaries [59, 70, 69, 71] and the number of measurements n is large enough, then it is indeed possible to recover the signal $\mathbf{x}$ from a similarly sized set of measurements $\mathbf{y}$. This incoherence property holds for many pairs of bases, including for example, delta spikes and the sine waves of a Fourier basis, or the Fourier basis and wavelets. Significantly, this incoherence also holds with high probability between *any arbitrary fixed basis and a randomly generated one*. This means that in general, if i.i.d. Gaussian or Bernoulli matrices are used for Ψ, this incoherence will still hold with high probability. This surprising result is a direct follow-on from the Restricted Isometry Property (RIP) which characterises matrices which are nearly orthonormal when operating on sparse vectors.

[9] "Dense" here is in the sense that each of the measurement vectors (rows of Ψ) must be spread out in the Φ domain. An example would be a Dirac function (spike) which is dense in the Fourier domain as it has a flat frequency response. Conversely a sine wave has a sparse representation in the Fourier domain as it is represented by a single frequency

2.2.6 Multivariate Signal Processing

This Section will introduce some signal processing operations for multivariate signals. Given a set of signals $x_i(n),\ i = 1, \ldots, M$ from a system, it is important to study whether there are possible interdependencies between the signals. Such interdependencies cause redundancies, which can be exploited for data compression. Interdependencies between the individual signals can also contain useful information about the structure of the underlying systems that generated the set of signals. The individual signals are often mixtures of unknown (latent) source signals $s_j(n)$, such that,

$$x_i(n) = \sum_{j=1}^{M} a_{i,j}\mathbf{s}_j(n), \quad i = 1, \ldots, M \tag{2.91}$$

$$\Rightarrow \mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) \tag{2.92}$$

The problem of finding the source signals $\mathbf{s}(n)$ from a set of measured signals $\mathbf{x}(n)$ is called source-signal separation. If the mixing matrix $\mathbf{A}$ is known, it is trivial to determine the source signal $\mathbf{s}(n)$ by inverting the linear relation in Equation 2.91. However in most cases this is not known; the problem of finding the source signals from the measured signals in this situation is called *blind deconvolution*. In order to solve the blind deconvolution problem some assumptions on the source signals have to be made. The most natural ones are that they are mutually uncorrelated or independent. PCA, which was introduced in Section 2.1.12 can be used for signal decorrelation.

Independent Components Analysis (ICA) is a method that performs deconvolution under the assumption that the latent sources are independent. The algorithm works by adaptively calculating the vectors of $\mathbf{A}$ and setting up a cost function which either maximises the non-Gaussianity of the calculated $\mathbf{s} = \mathbf{A}'\mathbf{x}$ or minimises the Mutual Information (MI) [72]. In some cases, *a-priori* knowledge of the probability distributions of the sources can be used in the cost function.

The original sources s can be recovered by multiplying the observed signals $\mathbf{x}$ with the inverse of the mixing matrix $\mathbf{W} = \mathbf{A}^{-1}$, also known as the *unmixing matrix*. Here it is assumed that the mixing matrix is square $(n = m)$. If the number of basis vectors is greater than the dimensionality of the observed vectors, $n > m$, the task is overcomplete but is still solvable.

Chapter 3

Sparse Machine Learning Framework for Multivariate Signal Processing

Abstract

Building blocks. *This Chapter present a unified general framework for the application of sparse machine learning methods to multivariate signal processing. The methods presented can be seen as modular building blocks that can be applied to a variety of applications. Application specific prior knowledge can be used in various ways, resulting in a flexible and powerful set of tools. The motivation for the methods is to be able to learn and generalise from a set of multivariate signals.*

In Pursuit of a Sparse Basis. *Given a dictionary of atoms from a given basis, a significant body of research has focussed on methods to select a sparse set of bases to represent a signal. Similarly, sparsity has been seen to be desirable for Machine Learning, for reasons of computation efficiency, regularisation, and compressibility.*

Greed is Good. *Within the suite of tools described in this chapter are a set of sub-optimal greedy sequential solvers for the sparse recovery problem. These have been shown to have desirable properties in the signal processing and statistics literature, it is shown through analysis and experimentation that these properties are also desirable in Machine Learning applications.*

Two Eyes are Better than One. *The final part of the chapter will detail developments in the area of "Multi-View" or "Multi-Source" Learning. We will present algorithmic developments in this area which will allow the incorporation of two or more sets of signals from different sources that will prove to be valuable in applications.*

3.1 Framework Outline

The goal of this Chapter is to outline a general modular framework designed for performing Machine Learning (ML) tasks. These are general purpose methods that link together to enable efficient inference

on a particular class of data, namely multivariate signals. The general approach is to combine methods from Digital Signal Processing (DSP) with methods from ML in novel ways that leverage the power of the methods from both fields. The main focus for this chapter will be the development of the framework for ML, although various approaches to DSP will be outlined along the way. The key will be to take a set of signals (such as recordings of a set of individuals' brain activity), and learn patterns that are then generalisable to a new set of signals generated under the same conditions (*i.e.* another individual performing the same task).

Multivariate signal processing is a source of challenges and opportunities. The traditional approach to multivariate signals has been to perform mass univariate analysis of the signals making the assumption that the signals are independent. However this independence assumption is violated more often than not, and as a result a great body of work has grown up around trying to make the univariate statistics more robust. For the purposes of this work the assumption will be made that the sensor arrays being dealt with are distributed in space but measured simultaneously (or as near as is possible), and that the sampling rate is fixed. There are of course situations where this assumption does not hold, but the methods outlined here can be extended, although the technical details become more complicated. For a univariate signal, there exist many well refined techniques for processing and classifying signals. These include Bayesian methods (*e.g.* using Markov Chain Monte Carlo (MCMC) methods [73]), Autoregressive Moving Average (ARMA) models [74], and analysis of spectral qualities of the signal (such as in [75]).

Figure 3.1 shows a top-level diagrammatic view of the process of learning from signals. Whilst the importance of the preprocessing stage should not be underestimated, it is not the focus of the present work. Hence the preprocessing used in all of the empirical testing will be via tried and tested methods that are well established in the various application areas visited. Details of specific preprocessing methods will be given such that the results of the experiments can be reproduced, but an extensive discussion is beyond the present scope. In addition, the diagram separates out preprocessing from signal processing; of course most of the preprocessing is in fact signal processing, but I have chosen to separate out the processing that is necessary to clean up data and remove artefacts (such as eye-blinks in EEG data) from the processing that is necessary to generate a set of features that describe the signals, which are then used as inputs to ML algorithms. This approach allows the focus to be maintained on the aspects of the interplay between DSP and ML of interest to the current study.

Of course the process outlined in Figure 3.1 is rather simplistic, and in fact in some cases can be improved upon. Specifically, a central theme that will be repeated throughout the thesis is that, wherever possible, one should make use of multiple paths of information flow. This can take the form of Multi-Source Learning (MSL) (where two separate sources of information are combined), MVL (where two views of the same underlying semantic object are combined), and Multiple Kernel Learning (MKL) (where multiple kernels are generated from a single source or view). These concepts will be described further in Section 3.5, in which algorithms that attempt to take advantage of these various paradigms will be developed.

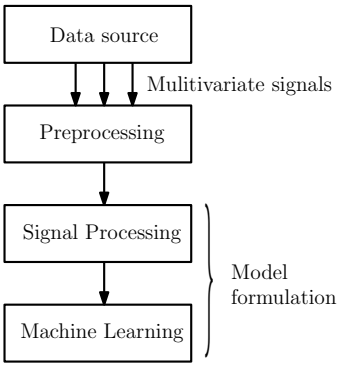


Figure 3.1: *Diagrammatic view of the process of machine learning from multivariate signals*

Sparse estimation and sparse recovery of patterns or signals are playing an increasingly important role in the statistics, signal processing, and ML communities. Several methods have recently been developed in both fields, which rely upon the notion of sparsity (*e.g.* penalty methods like the LASSO or greedy methods such as MP). Many of the key theoretical ideas and statistical analysis of the methods have been developed independently, but there is increasing awareness of the potential for cross-fertilization of ideas between statistics, signal processing and ML communities.

Much of the early effort has been dedicated to algorithms that solve sparsity inducing optimisation problems efficiently. This can be through first-order methods [76], or through *homotopy* methods that lead to the entire regularization path (*i.e.* , the set of solutions for all values of the regularization parameters) at the cost of a single matrix inversion [32]. A well-known property of the regularisation by the ℓ_1-norm is the sparsity of the solutions, *i.e.* , it leads to weight vectors with many zeros, and thus performs model selection on top of regularisation. Recent works have looked precisely at the model consistency of the LASSO [77, 78]. It has been shown that a condition known as the *irrepresentable* condition, which depends mainly on the covariance of the predictor variables, states that LASSO selects the true model consistently if and (almost) only if the predictors that are not in the true model are "irrepresentable" by predictors that are in the true model (see [77] for a discussion). This is effectively a statement that if it is known that the data were generated from a sparse weight vector, the LASSO does actually recover the sparsity pattern as the number of observations grows. This analysis has been extended to the Group LASSO and to MKL [78].

Furthermore, there are interesting links between penalty-type methods and boosting (particularly, LPBoost), as well as with sparse kernel regression. There has been interest in sparse methods within Bayesian ML (*e.g.* sparse PCA/CCA [79] or the Relevance Vector Machine (RVM) [80]). Sparse estimation is also important for unsupervised learning methods (*e.g.* sparse PCA and One-Class Support Vector Machine (OC-SVM) for outlier detection). Recent machine learning techniques for Multi-Task learning (MTL) [81, 82, 83] and collaborative filtering [84] have been proposed which implement sparsity constraints on matrices (rank, structured sparsity, etc.). At the same time, sparsity is playing an important role in various application fields, ranging from image and video reconstruction and compression, to speech classification, text and sound analysis.

In this Chapter we will begin by introducing a method that draws on the greedy method for sparse

signal reconstruction introduced in the previous chapter (OMP) and applies it to classification using the FDA objective function. Experimental results are given for this method showing that it performs competitively with state-of-the-art methods such as the SVM whilst producing solutions that are much more sparse. Furthermore, there is a clear performance gain when the datasets are very high dimensional and contain many potentially irrelevant features. Following on from this, we show that another greedy method from signal processing (PFP) can be applied to sparse regression problems in a kernel defined feature space. Again experimental results are given that show the power of this class of techniques. We will then go on to show that, surprisingly, it is in fact still possible to learn using a much simpler method of choosing basis vectors - that of random selection. The theoretical analysis shows that this result is due to a compression scheme being formed, which acts as a form of capacity control. Sparse learning can then be seen as a trade-off between finding the (near) optimal sparse solution by a greedy method, or finding sub-optimal solutions quickly that are *good enough*.

The final Section (3.5) of the Chapter is devoted to Multi-View Learning (MVL). The first contribution is an extension of the way in which KCCA projections are used for classification. Traditionally, an SVM (or any other standard ML algorithm) is trained on the projected subspace of the view of interest. However I show that good classification performance is possible using a method that is essentially *free* once the projections have been learnt. This method will be used for experimental analysis in Chapter 5. A natural extension to this is to try to incorporate the classification and the subspace learning into a single optimisation routine. This was the motivation for Multiview Fisher Discriminant Analysis (MFDA) and its variants, which will be presented towards the end of the chapter, along with some experimental results on toy data and benchmark datasets. Empirical analysis on real-world datasets will be presented in Chapter 5.

3.2 Greedy methods for Machine Learning

This Section will introduce two novel sparse ML methods. The first is based on the ideas of Matching Pursuit (MP) and Orthogonal Matching Pursuit (OMP) for sparse recovery in signal processing introduced in the last Chapter in Section 2.2.3, and focusses on the problem of classification using the KFDA algorithm outline in Section 2.1.9. This will be followed by a method based on Polytope Faces Pursuit (PFP).

3.2.1 Matching Pursuit Kernel Fisher Discriminant Analysis

A novel sparse version of KFDA is derived using an approach based on Orthogonal Matching Pursuit (OMP). This algorithm will be called Matching Pursuit Kernel Fisher Discriminant Analysis (MPKFDA). Generalisation error bounds are provided analogous to those constructed for the Robust Minimax algorithm together with a sample compression bounding technique. Experimental results are provided on real world datasets, which show that MPKFDA is competitive with the KFDA and the SVM on University of California, Irvine (UCI) datasets, and additional experiments that show that the

MPKFDA on average outperforms KFDA and SVM in extremely high dimensional settings.

The idea of MP is chosen for its fast greedy iterative property, and is applied to KFDA in order to impose dual sparsity. It will be proven that this sparse version results in generalisation error bounds guaranteeing its future success. The novel bounds come from the analysis by Shawe-Taylor *et. al.* [85] of the Robust Minimax algorithm of [86], which is similar in flavour to FDA. Together with the bounds of [85], a compression argument [87] is applied in order to gain an advantage due to the dual sparsity that results from the algorithm. However, the algorithm does not form a traditional compression scheme, so a similar idea to that of [88] is used to bound the generalisation error in the sparsely defined subspace by amalgamating both theories mentioned above. In some ways the bounds justify the choice of the fast iterative greedy strategy, which is not provably optimal [31], by guaranteeing that for a random choice of dataset from any fixed distribution, the predictions made will be *probably approximately correct* (PAC) [89].

One of the practical advantages of MPKFDA lies in the evaluation on test points - only k kernel evaluations are required (where k is the number of basis vectors chosen) compared to m (the number of samples) needed for KFDA. It is also worth stating that MPKFDA like KFDA has the advantage of directly delivering conditional probabilities of classification (unlike the SVM). There has been some research suggesting that one cannot estimate conditional probabilities without involving all of the data (see [90]) - hence kernel methods cannot deliver this efficiently - but here all of the data is taken into account whilst still having an efficient kernel representation.

Preliminaries

Most of the key quantities have already been introduced in Chapter 2, so this Section gives a brief summary. We denote with S a sample containing m examples $\mathbf{x} \in \mathbb{R}^n$ and labels $y \in \{-1, 1\}$. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)'$ be the input vectors stored in matrix $\mathbf{X}$ as row vectors, where $'$ denotes the transpose of vectors or matrices. For simplicity it is assumed that the examples are already projected into the kernel defined feature space, so that the kernel matrix $\mathbf{K}$ has entries $\mathbf{K}[i, j] = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. In the analysis Section, $\phi(\mathbf{x})$ will explicitly denote the feature map for some vector $\mathbf{x}$. The notation $\mathbf{K}[:, i]$ will denote the ith column of the matrix $\mathbf{K}$. When given a set of indices $\mathbf{i} = \{i_1, \ldots, i_k\}$ (say) then $\mathbf{K}[\mathbf{i}, \mathbf{i}]$ denotes the square matrix defined solely by the index set $\mathbf{i}$.

For analysis purposes it is assumed that the training examples are generated i.i.d. according to an unknown but fixed probability distribution that also governs the generation of the test data. Expectation over the training examples (empirical average) is denoted by $\hat{\mathbb{E}}[\cdot]$, while expectation with respect to the underlying distribution is denoted $\mathbb{E}[\cdot]$.

For the sample compression analysis the *compression function* Λ induced by a sample compression learning algorithm A on training set S is the map $\Lambda : S \longmapsto \Lambda(S)$ such that the *compression set* $\Lambda(S) \subset S$ is returned by A. A *reconstruction function* Ψ is a mapping from a compression set $\Lambda(S)$ to a set F of functions $\Psi : \Lambda(S) \longmapsto F$.

Let $A(S)$ be the function output by learning algorithm A on training set S. Therefore, a sample

compression scheme is a reconstruction function Ψ mapping a compression set $\Lambda(S)$ to some set of functions F such that $A(S) = \Psi(\Lambda(S))$. If F is the set of Boolean-valued functions then the sample compression scheme is said to be a classification algorithm.

Define $\hat{\boldsymbol{\mu}}(\boldsymbol{\mu})$ to be the empirical (true) mean of a sample of m points from the set S projected into a higher dimensional space using ϕ,

$$\boldsymbol{\mu} = \mathbb{E}[\phi(\mathbf{x})],$$
$$\hat{\boldsymbol{\mu}} = \frac{1}{m}\sum_{i=1}^{m}\phi(\mathbf{x}_i),$$

and $\hat{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma})$ its empirical (true) covariance matrix.

Algorithm

OMP can be formalised as a general framework in ML, that involves repeating the following two steps:

1. Function maximisation; *and*
2. Deflation (orthogonalisation).

It can result in OMP algorithms for learning tasks other than regression. This Section presents an application of this general framework to KFDA, resulting in a sparse form of KFDA that we refer to as MPKFDA.

An OMP algorithm for FDA can be built in the following way. Initially, one example $\mathbf{i} = \{i_\ell\}$ is chosen that maximises the FDA criterion and the remaining training examples are projected into the space defined by $\mathbf{i}$. Following this the data matrix $\mathbf{X}$ (or kernel $\mathbf{K}$) is deflated to allow the next index to be chosen. Finally this results in a set $\mathbf{i}$ of training examples that can be used to compute the final weight vector $\mathbf{w}$, together with the FDA decision function $f(\mathbf{x}) = \text{sgn}\,(\mathbf{w}'\mathbf{x} + b)$ where b is the bias and $\mathbf{x}$ an example.

Using the notation from [3], the maximisation problem for FDA is given by the following:

$$\mathbf{w} \quad = \quad \max_{\mathbf{w}} \frac{\mathbf{w}'\mathbf{X}'\mathbf{y}\mathbf{y}'\mathbf{X}\mathbf{w}}{\mathbf{w}'\mathbf{X}'\mathbf{B}\mathbf{X}\mathbf{w}}, \tag{3.1}$$

where $\mathbf{B}$ is defined as in Section 2.1.9 of Chapter 2.

To begin with, the Nyström method of low-rank approximation of the Gram matrix [7] is applied. This is defined in the following Section.

3.2.2 Nyström Low-Rank Approximations

The Nyström method generates a low-rank approximation of a Gram matrix $\mathbf{G}$ using a subset $\mathbf{i} = (i_1, \ldots, i_k)$ of k of the columns [7]. The method will readily apply to RKHS simply by replacing $\mathbf{G}$ with the kernel matrix $\mathbf{K}$, but the more general definition will be given here. Given a sample of k columns of $\mathbf{G}$ selected by some method, let $\mathbf{N} = \mathbf{G}[:, \mathbf{i}]$ be the $n \times k$ matrix of the sampled columns, and

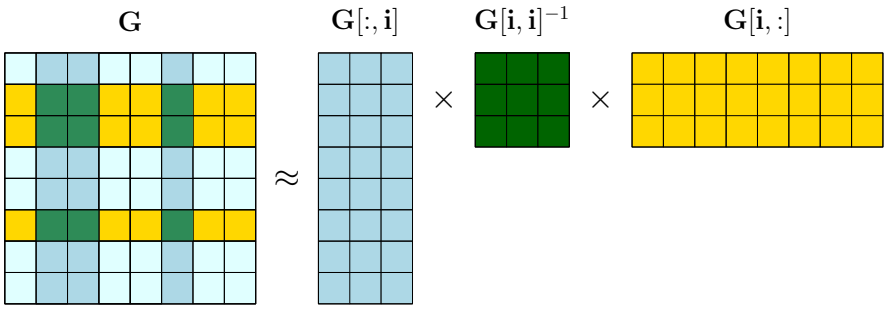


Figure 3.2: *Diagrammatic representation of the Nyström method*

$\mathbf{W} = \mathbf{G}(\mathbf{i}, \mathbf{i})$ be the $k \times k$ matrix consisting of the intersection of these k columns with the corresponding k rows of $\mathbf{G}$. The Nyström method uses $\mathbf{W}, \mathbf{N}$ to construct a rank-k approximation $\tilde{\mathbf{G}}_k$ to $\mathbf{G}$,

$$\tilde{\mathbf{G}}_k = \mathbf{N}\mathbf{W}_k^{-1}\mathbf{N}' \approx \mathbf{G}. \tag{3.2}$$

In practice the matrix $\mathbf{W}_k$ may not be invertible, especially for small k, in which case the pseudo-inverse[1] is used. The Nyström approximation is depicted in figure 3.2. Define $\mathbf{R}$ is the Cholesky decomposition of $\mathbf{W}_k^{-1}$ such that $\mathbf{R}$ is an upper triangular matrix that satisfies $\mathbf{R}'\mathbf{R} = \mathbf{G}[\mathbf{i}, \mathbf{i}]^{-1}$.

Nyström for Matching Pursuit Kernel Fisher Discriminant Analysis (MPKFDA)

Using the assumption that the inputs $\mathbf{X}$ have already been projected into the kernel defined feature space, the Nyström approximation can be applied to the kernel matrix $\mathbf{K}$. A greedy algorithm will be used to select a set of bases $\mathbf{i}$, such that $\mathbf{N} = \mathbf{K}[:, \mathbf{i}]$ and $\mathbf{W}_k = \mathbf{K}[\mathbf{i}, \mathbf{i}]$. The Nyström approximation for MPKFDA is then,

$$\begin{aligned} \tilde{\mathbf{K}} &= \mathbf{K}[:, \mathbf{i}]\mathbf{K}[\mathbf{i}, \mathbf{i}]^{-1}\mathbf{K}[:, \mathbf{i}]' \\ &= \mathbf{K}[:, \mathbf{i}]\mathbf{R}'\mathbf{R}\mathbf{K}[:, \mathbf{i}]' \approx \mathbf{K}, \end{aligned} \tag{3.3}$$

where $\mathbf{R}$ is the Cholesky decomposition of $\mathbf{K}[\mathbf{i}, \mathbf{i}]^{-1}$ such that $\mathbf{R}$ is an upper triangular matrix that satisfies $\mathbf{R}'\mathbf{R} = \mathbf{K}[\mathbf{i}, \mathbf{i}]^{-1}$.

However, rather than use the full $[m \times m]$ low rank approximation, it would be preferable to work in the $[k \times k]$ space where $k \ll m$. In order to do this $\mathbf{K}[:, \mathbf{i}]\mathbf{R}'$ is treated as a new input $\mathbf{X}$ in FDA, which results in a projection $\tilde{\phi}$ into a k-dimensional subspace:

$$\tilde{\phi}(\mathbf{x}_i) = \mathbf{K}[:, \mathbf{i}]\mathbf{R}'. \tag{3.4}$$

Within this space the following

$$\tilde{\mathbf{\Sigma}}_k = \mathbf{R}\mathbf{K}[:, \mathbf{i}]'\mathbf{K}[:, \mathbf{i}]\mathbf{R}', \tag{3.5}$$

[1] $\mathbf{A}^\dagger$ is defined as the Moore-Penrose pseudo-inverse of a matrix $\mathbf{A}$.

is the covariance matrix within this space. This enables large scale problems containing m data points to be solved with linear algorithms using k features. This trick allows nonlinear discriminant analysis to be performed on a sparse subspace using standard linear FDA.

Greedy Selection of Bases

For the algorithm to proceed, a method for the greedy selection of basis vectors is required. The following maximisation problem for a dual sparse version of FDA can be defined by setting $\mathbf{w} = \mathbf{X}'\mathbf{e}_i$ where $\mathbf{e}_i$ is the i^{th} unit vector of length m, and substituting into the FDA problem described above (ignoring constants) to yield:

$$\begin{aligned} \arg\max_i \rho_i &= \frac{\mathbf{e}_i'\mathbf{X}\mathbf{X}'\mathbf{y}\mathbf{y}\mathbf{X}\mathbf{X}'\mathbf{e}_i}{\mathbf{e}_i'\mathbf{X}\mathbf{X}'\mathbf{B}\mathbf{X}\mathbf{X}'\mathbf{e}_i} \\ &= \frac{\mathbf{K}[:,i]'\mathbf{y}\mathbf{y}'\mathbf{K}[:,i]}{\mathbf{K}[:,i]'\mathbf{B}\mathbf{K}[:,i]} \end{aligned} \tag{3.6}$$

Maximising the quantity above leads to maximisation of the Fisher Discriminant Ratio (FDR) corresponding to $\mathbf{e}_i$, and hence a sparse subset of the original KFDA problem. The goal is to find the optimal set of indices $\mathbf{i}$. The approach taken here is to proceed in a greedy manner (MP), in much the same way as [37] and [65]. The procedure involves choosing basis vectors that maximise the Fisher Discriminant ratio iteratively until some pre-specified number of k vectors are chosen.

The next step is to orthongalise the matrix $\mathbf{K}$ with respect to the chosen basis vector $\boldsymbol{\tau} = \mathbf{K}[:,i]$. In the primal form of PCA, the deflation can be carried out using Hotelling's method [91] with respect to the features (columns of an input matrix $\mathbf{X}$) by,

$$\tilde{\mathbf{X}}' = \left(\mathbf{I} - \frac{\mathbf{u}\mathbf{u}'}{\mathbf{u}'\mathbf{u}}\right)\mathbf{X}', \tag{3.7}$$

where $\mathbf{u}$ is a chosen eigenvector and $\tilde{\mathbf{X}}$ is the deflated version of $\mathbf{X}$. However because we are working in the dual (kernel) space, the projection directions are simply the examples in $\mathbf{X}$, so $\mathbf{u} = \mathbf{X}'\mathbf{e}$. If we define $\boldsymbol{\tau} = \mathbf{X}\mathbf{X}'\mathbf{e} = \mathbf{K}[:,i]$, the deflation $\tilde{\mathbf{K}}$ of the kernel with respect to the chosen basis i is then,

$$\tilde{\mathbf{K}} = \left(\mathbf{I} - \frac{\boldsymbol{\tau}\boldsymbol{\tau}'}{\boldsymbol{\tau}'\boldsymbol{\tau}}\right)\mathbf{K}. \tag{3.8}$$

This deflation ensures that remaining potential basis vectors will be chosen from a space that is orthogonal to those bases already picked[2]. After choosing the k training examples, giving $\mathbf{i} = (i_1, \ldots, i_k)$, $\mathbf{R}\mathbf{K}[:,\mathbf{i}]'$ can be defined as a new data matrix as defined in Section 3.2.2 above. FDA is then used for training as in Equation 3.1 in this new projected space to find a k-dimensional weight vector $\mathbf{w}_k$, which is indexed over the bases of the kernel matrix and hence has sparsity k in the dual sense. Given the index j of a test point $\mathbf{x}_j$, and using the train-test kernel on this point $\mathbf{K}[j,\mathbf{i}]$ and its projection

[2] It is assumed that the vectors of the matrix $\mathbf{K}$ do not form an orthonormal basis

$\phi(\mathbf{x}_j) = \mathbf{R}\mathbf{K}[j, \mathbf{i}]'$, predictions can be made using the FDA prediction function,

$$f(\mathbf{x}_j) = \operatorname{sgn}\left(\langle \tilde{\mathbf{w}}, \phi(\mathbf{x}_j)\rangle + b\right) \tag{3.9}$$

The algorithm for MPKFDA is given in Algorithm 3.

Algorithm 3 Matching Pursuit Kernel Fisher Discriminant Analysis

Input: kernel $\mathbf{K}$, sparsity parameter $k > 0$, training labels $\mathbf{y}$.

1: calculate matrix $\mathbf{B}$
2: initialise $\mathbf{i} = (\,)$
3: **for** $j = 1$ to k **do**
4: $\quad t \leftarrow \arg\max_i \frac{\mathbf{K}[:,i]'\mathbf{y}\mathbf{y}'\mathbf{K}[:,i]}{\mathbf{K}[:,i]'\mathbf{B}\mathbf{K}[:,i]}$
5: $\quad \mathbf{i} \leftarrow \{\mathbf{i}, t\}$
6: $\quad \boldsymbol{\tau} \leftarrow \mathbf{K}[:,t]$ to deflate kernel matrix like so:

$$\mathbf{K} \leftarrow \left(\mathbf{I} - \frac{\boldsymbol{\tau}\boldsymbol{\tau}'}{\mathbf{K}[t,t]}\right)\mathbf{K}$$

7: **end for**
8: calculate the projection $\mathbf{R}\mathbf{K}[:, \mathbf{i}]'$ where $\mathbf{R}$ is the Cholesky decomposition of $\mathbf{K}[\mathbf{i}, \mathbf{i}]^{-1}$ and $\mathbf{i} = (\mathbf{i}_1, \ldots, \mathbf{i}_k)$
9: train FDA using Equation 3.1 in this new projected space to find a sparse weight vector $\tilde{\mathbf{w}}$ and make predictions using Equation 3.9

Output: final set $\mathbf{i}$, (sparse) weight vector $\tilde{\mathbf{w}}$, bias term b

Generalisation Error Analysis

A generalisation error bound for MPKFDA can now be constructed by applying the results from [85] with a compression argument. The following two results from [85] will be needed. The first bounds the difference between the empirical and true means.

Theorem 3.2.1 (Bound on the true and empirical means)**.** *Let S be an m sample generated independently at random according to a distribution P. Then with probability at least $1 - \delta$ over the choice of S,*

$$\|\hat{\boldsymbol{\mu}} - \mathbb{E}_{\mathbf{x}}[\phi(\mathbf{x})]\| \leq \frac{R}{\sqrt{m}}\left(2 + \sqrt{2\ln\frac{1}{\delta}}\right), \tag{3.10}$$

where $\hat{\boldsymbol{\mu}} = \hat{\mathbb{E}}[\phi(\mathbf{x})]$ and where R is the radius of the ball in the feature space containing the support of the distribution. Consider the covariance matrix defined as

$$\boldsymbol{\Sigma} = \mathbb{E}\left\|(\phi(\mathbf{x}) - \boldsymbol{\mu})(\phi(\mathbf{x}) - \boldsymbol{\mu})'\right\|.$$

Let the empirical estimate of this quantity be

$$\hat{\boldsymbol{\Sigma}} = \hat{\mathbb{E}}\left\|(\phi(\mathbf{x}) - \hat{\boldsymbol{\mu}})(\phi(\mathbf{x}) - \hat{\boldsymbol{\mu}})'\right\|.$$

The following corollary bounds the difference between the empirical and true covariance.

Corollary 3.2.2 (Bound on the true and empirical covariances)**.** *Let S be an m sample generated independently at random according to a distribution D. Then with probability at least $1-\delta$ over the choice of S,*

$$\left\|\hat{\mathbf{\Sigma}}-\mathbf{\Sigma}\right\|_F \leq \frac{2R^2}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{2}{\delta}}\right), \tag{3.11}$$

where $\|\mathbf{A}\|_F = \sqrt{\mathrm{trace}\,(\mathbf{A}\mathbf{A}')}$ is the Frobenius norm of a matrix $\mathbf{A}$, and provided

$$m \geq \left(2+\sqrt{2\ln\frac{2}{\delta}}\right)^2.$$

The following Lemma is connected with a classification algorithm developed in [86]. The basis for the approach is the following Lemma.

Lemma 3.2.3. *Let μ be the mean of a distribution and $\mathbf{\Sigma}$ its covariance matrix, $\mathbf{w}\neq 0$, b given, such that $\mathbf{w}'\boldsymbol{\mu}\leq b$ and $\alpha\in[0,1)$, then if*

$$b-\mathbf{w}'\boldsymbol{\mu}\geq\varphi(\alpha)\sqrt{\mathbf{w}'\mathbf{\Sigma}\mathbf{w}},$$

where $\varphi(\alpha)=\sqrt{\frac{\alpha}{1-\alpha}}$, then

$$\Pr\left(\mathbf{w}'\phi(\mathbf{x})\leq b\right)\geq\alpha$$

We will of course be using empirical estimates of $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$. In order to provide a true error bound, the difference between the resulting estimate and the value that would have been obtained had the true mean and covariance been used must be bounded.

Bound for Matching Pursuit Kernel Fisher Discriminant Analysis

The above bound is applied to a subspace defined from a small number $k\ll m$ of basis vectors. Let $\mathbf{i}=(i_1,\ldots,i_k)$ be a vector of indices used to form a k- dimensional subspace such as the one defined by MPKFDA. The notation $S_{\mathbf{i}}$ is used to denote the samples pointed to by $\mathbf{i}$. Firstly a general bound is given, which is then specialised to the case of MPKFDA.

Theorem 3.2.4 (main)**.** *Let S be a sample of m points drawn independently according to a probability distribution D where R is the radius of the ball in the feature space containing the support of the distribution. Let $\hat{\boldsymbol{\mu}}_k$ ($\boldsymbol{\mu}_k$) be the empirical (true) mean of a sample of $m-k$ points from the set $S\setminus S_{\mathbf{i}}$ projected into a k-dimensional space, $\hat{\mathbf{\Sigma}}_k$ ($\mathbf{\Sigma}_k$) its empirical (true) covariance matrix, $\mathbf{w}_k\neq 0$ with norm 1, and b_k given, such that $\mathbf{w}_k'\boldsymbol{\mu}_k\leq b_k$ and $\alpha\in[0,1)$. Then with probability $1-\delta$ over the draw of the random sample, if*

$$b_k-\mathbf{w}_k'\hat{\boldsymbol{\mu}}_k\geq\varphi(\alpha)\sqrt{\mathbf{w}_k'\hat{\mathbf{\Sigma}}_k\mathbf{w}_k},$$

then

$$\Pr\left(\mathbf{w}_k'\phi(\mathbf{x}) - b_k > 0\right) < 1 - \alpha,$$

where

$$\alpha = \frac{\left(b_k - \mathbf{w}_k'\hat{\boldsymbol{\mu}}_k - A\right)^2}{\mathbf{w}_k'\hat{\boldsymbol{\Sigma}}_k\mathbf{w}_k + B + \left(b_k - \mathbf{w}_k'\hat{\boldsymbol{\mu}}_k - A\right)^2},$$

such that $\|\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k\| \leq A$ *where*

$$A = \frac{R}{\sqrt{m-k}}\left(2 + \sqrt{2k\ln\frac{em}{k} + 2\ln\frac{m}{\delta}}\right)$$

and $\left\|\hat{\boldsymbol{\Sigma}}_k - \boldsymbol{\Sigma}_k\right\|_F \leq B$ *where*

$$B = \frac{2R^2}{\sqrt{m-k}}\left(2 + \sqrt{2k\ln\frac{em}{k} + 2\ln\frac{2m}{\delta}}\right).$$

Proof. First, $b_k - \mathbf{w}'\mu \geq \varphi(\alpha)\sqrt{\mathbf{w}'\boldsymbol{\Sigma}\mathbf{w}}$ from Lemma 3.2.3 can be rearranged in terms of $\varphi(\alpha)$:

$$\varphi(\alpha) \leq \frac{b_k - \mathbf{w}'\boldsymbol{\mu}}{\sqrt{\mathbf{w}'\boldsymbol{\Sigma}\mathbf{w}}}. \tag{3.12}$$

These quantities are in terms of the true means and covariances in the chosen subspace. In order to achieve an upper bound, Theorem 3.2.1 and Corollary 3.2.2 must be applied for each of the $\binom{m}{k}$ choices of the compression set, and we further apply a factor of $1/m$ to δ to ensure one application of the bound for each possilbe choice of k. This leads to the substitution of $\delta/(m\binom{m}{k})$ in place of δ, and the substitution of $m-k$ for m for the size of the dataset,

$$\|\hat{\boldsymbol{\mu}}_k - \mathbb{E}_{\mathbf{x}}[\hat{\boldsymbol{\mu}}_k(\mathbf{x})]\| \leq \frac{R}{\sqrt{m-k}}\left(2 + \sqrt{2\ln\frac{m\binom{m}{k}}{\delta}}\right),$$

and

$$\left\|\hat{\boldsymbol{\Sigma}}_k - \boldsymbol{\Sigma}_k\right\|_F \leq \frac{2R^2}{\sqrt{m-k}}\left(2 + \sqrt{2\ln\frac{2m\binom{m}{k}}{\delta}}\right).$$

Use the fact that $\binom{m}{k}$ is upper bounded by $(em/k)^k$, and rearranging gives,

$$\|\hat{\boldsymbol{\mu}}_k - \mathbb{E}_{\mathbf{x}}[\hat{\boldsymbol{\mu}}_k(\mathbf{x})]\| \leq \frac{R}{\sqrt{m-k}}\left(2 + \sqrt{2k\ln\frac{em}{k} + 2\ln\frac{m}{\delta}}\right) := A,$$

and

$$\left\|\hat{\mathbf{\Sigma}}_k - \mathbf{\Sigma}_k\right\|_F \leq \frac{2R^2}{\sqrt{m-k}}\left(2 + \sqrt{2k\ln\frac{em}{k} + 2\ln\frac{2m}{\delta}}\right) := B.$$

Given Equation 3.12, the empirical quantities for the means and covariances can be used in place of the true quantities. However, in order to derive a genuine upper bound, the upper bounds between the empirical and true means also need to be taken into account. These are included in the expression above for $\varphi(\alpha)$ by replacing δ with $\delta/2$, to derive a lower bound, like so:

$$\varphi(\alpha) = \frac{b_k - \mathbf{w}_k'\hat{\boldsymbol{\mu}}_{S_k} - A}{\sqrt{\mathbf{w}_k'\hat{\mathbf{\Sigma}}_k\mathbf{w}_k + B}}.$$

Finally, making the substitution $\varphi(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}$ and solving for α yields the result. □

The following Proposition upper bounds the generalisation error of MPKFDA.

Proposition 3.2.5. *Let $\mathbf{w}_k$, b_k, be the (normalised) weight vector and associated threshold returned by the MPKFDA algorithm when presented with a training set S. Furthermore, let $\hat{\mathbf{\Sigma}}_k^+$ ($\hat{\mathbf{\Sigma}}_k^-$) be the empirical covariance matrices associated with the positive (negative) examples of the $m-k$ training samples from $S \setminus S_{\mathrm{i}}$ projected into a k dimensional space. Then with probability at least $1-\delta$ over the draw of the random training set S of m training examples, the generalisation error ϵ is bounded by*

$$\epsilon \leq \max(1-\alpha^+, 1-\alpha^-)$$

where α^j, $j = +, -$ are given by,

$$\alpha^j = \frac{\left(j(\mathbf{w}_k'\hat{\boldsymbol{\mu}}^j_{S_k} - b_k) - C^j\right)^2}{\mathbf{w}_k'\hat{\mathbf{\Sigma}}^j_k\mathbf{w}_k + D^j + \left(j(\mathbf{w}_k'\hat{\boldsymbol{\mu}}^j_{S_k} - b_k) - C^j\right)^2},$$

where

$$C^j = \frac{R}{\sqrt{m^j - k^j}}\left(2 + \sqrt{2k\ln\frac{em}{k} + 2\ln\frac{2m}{\delta}}\right),$$

and

$$D^j = \frac{2R^2}{\sqrt{m^j - k^j}}\left(2 + \sqrt{2k\ln\frac{em}{k} + 2\ln\frac{4m}{\delta}}\right).$$

Proof. For the negative (-1) part of the proof, $b_k - \mathbf{w}_k'\hat{\boldsymbol{\mu}}_k^- \geq \varphi(\alpha)\sqrt{\mathbf{w}_k'\hat{\mathbf{\Sigma}}_k^-\mathbf{w}_k}$ is required, which is a straight forward application of Theorem 3.2.4 with δ replaced with $\delta/2$. For the positive ($+1$) part, observe $-b_k + \mathbf{w}_k'\hat{\boldsymbol{\mu}}_k^+ \geq \varphi(\alpha)\sqrt{\mathbf{w}_k'\hat{\mathbf{\Sigma}}_k^+\mathbf{w}_k}$ is required, hence, a further application of Theorem 3.2.4 with δ replaced by $\delta/2$ suffices. □

Experiments

A comparison on 13 benchmark datasets derived from the UCI, Data for Evaluating Learning in Valid Experiments (DELVE) and STATLOG benchmark repositories follows. The performance of KFDA, MPKFDA, and SVM using RBF kernels are analysed. The data comes in 100 predefined splits into training and test sets (20 in the case of the image and splice datasets) as described in [34] [3]. For each of the datasets CV was used to select the optimal parameters (the RBF kernel width parameter, the C parameter in the SVM, and k the number of iterations in MPKFDA). 5-fold CV was used over the first five training datasets with a coarse range of parameter values, selecting the median over the five sets as the optimal value, followed by a similar process using a fine range of parameter values[4]. This way of estimating the parameters leads to more robust comparisons between the methods. The means and SDs of the generalisation error for each method and dataset are given in Table 3.1. It was found that the performance of KFDA and MPKFDA are very similar, and both are competitive with the SVM. This is demonstrated by the values for the mean over the datasets.

	Dim	Train	Test	KFDA		MPKFDA			SVM		
				Error	SD	Error	SD	k	Error	SD	k
Banana	2	400	4900	0.1069	0.00	0.1101	0.01	31	0.1068	0.00	122
Breast Cancer	9	200	77	0.2886	0.05	0.3174	0.04	19	0.2603	0.05	113
Diabetes	8	468	300	0.2596	0.02	0.2543	0.02	18	0.2332	0.02	260
Flare Solar	9	666	400	0.3500	0.02	0.3457	0.02	19	0.3239	0.02	557
German	20	700	300	0.2672	0.02	0.2808	0.02	27	0.2345	0.02	392
Heart	13	170	100	0.2125	0.03	0.1599	0.03	13	0.1543	0.03	98
Image	18	1300	1010	0.0092	0.02	0.0136	0.03	39	0.0061	0.01	27
Ringnorm	20	400	7000	0.0685	0.01	0.0573	0.03	15	0.0164	0.00	216
Splice	60	1000	2175	0.0397	0.08	0.0314	0.06	37	0.0223	0.05	110
Thyroid	5	140	75	0.0392	0.02	0.0699	0.03	29	0.0520	0.02	87
Titanic	3	150	2051	0.2259	0.02	0.2468	0.05	70	0.2256	0.01	76
Twonorm	20	400	7000	0.0253	0.00	0.0253	0.00	14	0.0280	0.00	231
Waveform	21	400	4600	0.1228	0.01	0.1027	0.00	13	0.1031	0.00	131
Mean				0.1550	0.02	0.1550	0.03	26.5	0.1359	0.02	185.3

Table 3.1: *Error estimates and Standard Deviations (SDs) and sparsity level k (number of bases for MPKFDA or number of support vectors for SVM) for 13 benchmark datasets.*

Results from the Neural Information Processing Systems (NIPS) 2003 challenge datasets [92] ARCENE, DEXTER and DOROTHEA are presented next[5]. These datasets were chosen with the belief that the main advantage of MPKFDA will be shown when the data lives in high dimensions. Comparisons were made between the performance of MPKFDA with standard KFDA and SVM, again using RBF kernels for each of the classifiers. 5-fold CV was used on the training set to select the optimal parameters for each algorithm as before, and then tested on the validation set. For each dataset the following are shown: the number of features; the number of examples in the training and validation sets; the generalisation error of each classifier on the validation set. All problems are two-class classification problems. As can be seen from Table 3.2, MPKFDA outperforms both KFDA and SVM on these high

[3] Available to download from: `http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm`

[4] The coarse values were $10^{-6,\ldots,3}$ and the fine range consisted of 9 logarithmically spaced values between 10^{v-1} and 10^{v+1} where v is $\log_{10}$ of the value chosen at the first stage

[5] The train and validation sets and associated labels are available for download from: `http://www.nipsfsc.ecs.soton.ac.uk/datasets/`

dimensional datasets, whilst giving very sparse solutions.

	Dim	Train	Test	KFDA	MPKFDA		SVM	
				Error	Error	k	Error	k
Arcene	10000	100	100	0.2000	0.1800	40	0.2600	80
Dexter	20000	300	300	0.1133	0.0800	40	0.0733	257
Dorothea	100000	800	350	0.0971	0.0571	11	0.0686	711
Mean				0.1368	0.1057	30.3	0.1340	349.3

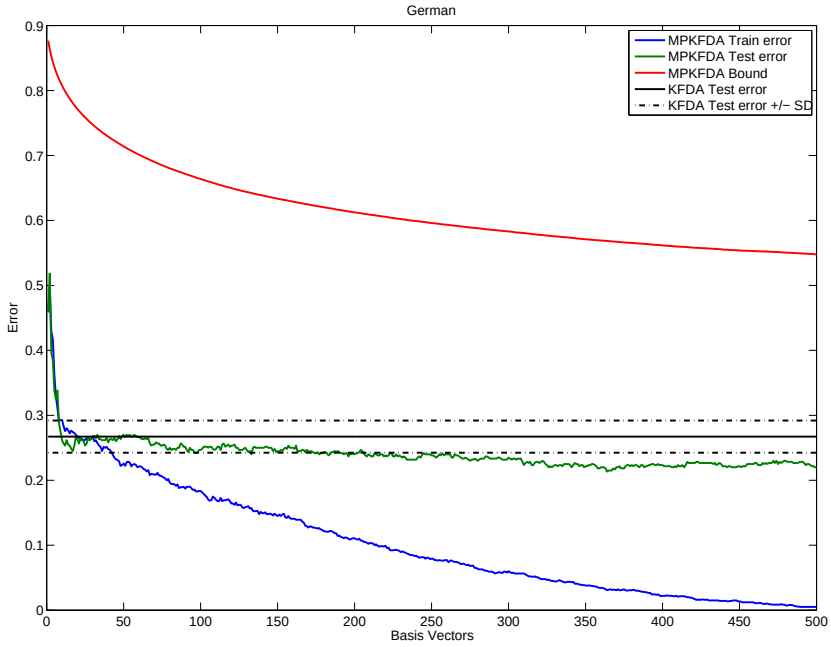
Table 3.2: *Error estimates for MPKFDA on 3 high dimensional datasets.*

Figures 3.3 a) and b) show plots of the train and test error of MPKFDA on two of the datasets ('German' and 'Banana') as k increases compared against KFDA. The plots demonstrate that MPKFDA algorithm is very resistant to overfitting, and gives good generalisation performance with relatively small k. The value of the bound is also plotted. However it is too pessemistic (it levels off for much higher k) and therefore cannot be used for model selection.

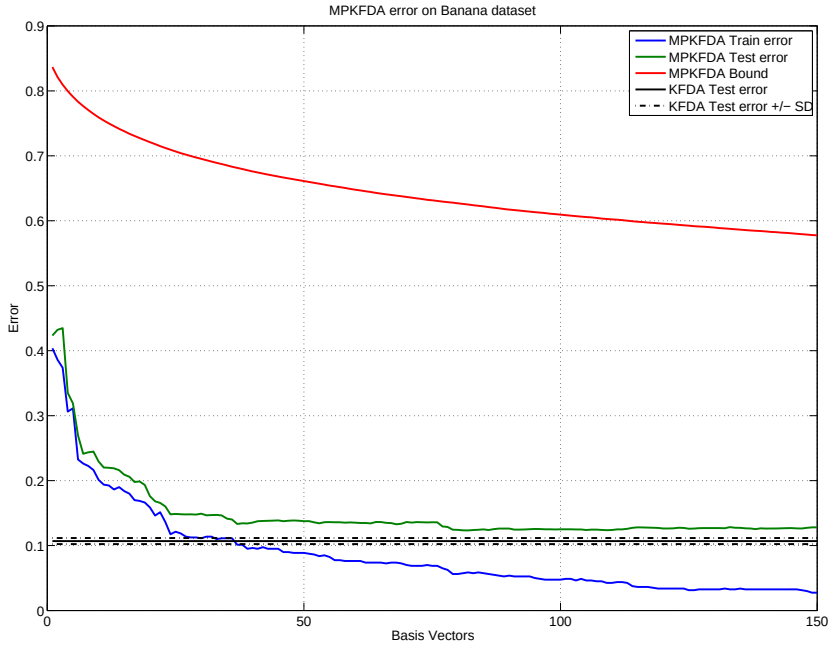
It is also interesting to investigate why the algorithm is resistant to overfitting. Firstly note that the deflation step means that the rank of the kernel matrix is being reduced by at least 1 at each iteration. Also, the Frobenius norm of the kernel matrix is being reduced, although the effect of this will be greater at earlier steps. Meanwhile, the norm of the weight vector (if unnormalised) grows as bases are added, but the rate of this reduction decreases over time. This means that as k grows the bases that are added will have less and less impact on the solution. Figure 3.4 shows the relative sizes of the Frobenius norm of the kernel matrix and the generalisation error as k increses (different scales on the y-axis). Effectively the deflation step is acting as a strong regulariser, which when combined with the intrinsic regularisation effects of the compression introduced by the sparsity of the solutions, leads to a resistance to overfitting.

In this Section a novel sparse version of KFDA was derived using an approach based on MP. Generalisation error bounds were provided that were analogous to that used in the Robust Minimax algorithm [86], together with a sample compression bounding technique. As shown the bound is too loose to perform model selection, but further analysis may enable the bound to drive the algorithm. Experimental results on real world datasets were presented, which showed that MPKFDA is competitive with both KFDA and SVM, and additional experiments that showed that MPKFDA performs well in high dimensional settings. In terms of computational complexity the demands of MPKFDA during training are higher, but during the evaluation on test points only k kernel evaluations are required compared to m needed for KFDA. This does, however, pose a problem for scaling to very large datasets, as the deflation step is $\mathcal{O}(m^3)$ at each step.

In the next Section an algorithm based on another greedy method, Polytope Faces Pursuit (PFP), is presented. This time the focus will be on nonlinear regression, showing that greedy methods are widely applicable in ML.



a)



b)

Figure 3.3: *Plot of generalisation error bound for different values of k using RBF kernels for the a) 'German' and b) 'Banana' data set. The generalisation error is shown on the y axis. The plot shows the training error (in blue), the test error (in green), the bound value (in red), and the test error of the KFDA classifier (in black, with dotted lines showing the Standard Deviation (SD)). Note that the MPKFDA algorithm is very resistant to overfitting, and gives good generalisation performance with relatively small k. The bound is too pessemistic (it levels off for much higher k) and therefore cannot be used for model selection.*

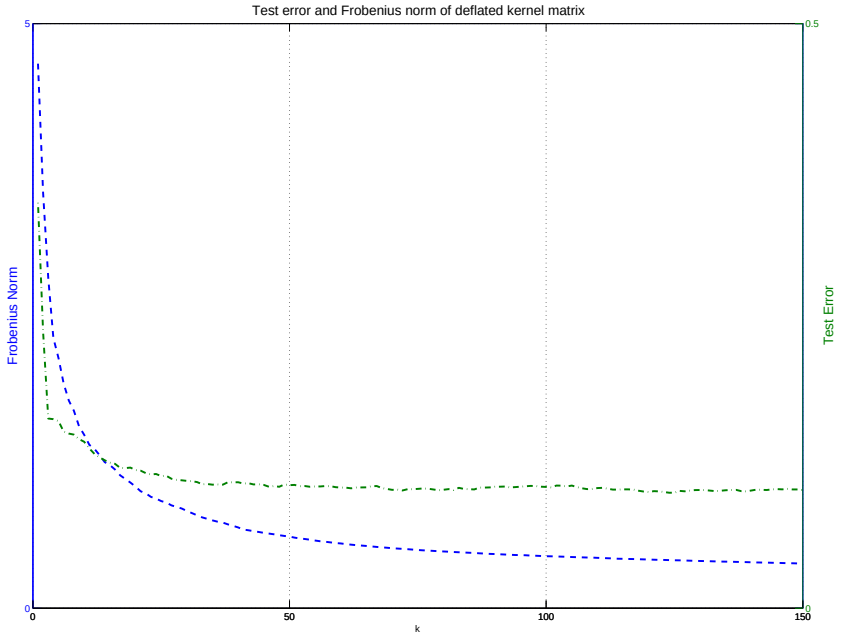


Figure 3.4: *Plot showing how the Frobenius norm of the deflated kernel matrix and the test error vary as basis vectors are added to the MPKFDA solution.*

3.3 Kernel Polytope Faces Pursuit

Polytope Faces Pursuit (PFP) is a greedy algorithm that approximates the sparse solutions recovered by ℓ_1 regularised least-squares (LASSO) [60, 61] in a similar way to MP and OMP [93]. The algorithm is based on the geometry of the polar polytope where at each step a basis function is chosen by finding the maximal vertex using a path-following method. The algorithmic complexity is of a similar order to OMP whilst being able to solve problems known to be hard for MP and OMP. MP was extended to build kernel-based solutions to machine learning problems, resulting in the sparse regression algorithm, KMP [65]. A new algorithm to build sparse kernel-based solutions using PFP is presented here, called Kernel Polytope Faces Pursuit (KPFP). The utility of this algorithm will be demonstrated firstly by providing a generalisation error bound [88] that takes into account a natural regression loss, and secondly with experimental results on several benchmark datasets. In the following the KPFP algorithm will be derived, which is a generalisation of PFP to a RKHS.

PFP was outlined in the previous Chapter in Section 2.2.3. At each step the approach to the solution of this problem is to identify the optimal vertex which is the maximiser of $\mathbf{y}'\mathbf{c}$, where $\mathbf{c}$ is the dimensional weight vector of the ℓ_1-minimisation in its standard form, which is similar to the way in which KMP builds up its solution. However the difference is that at each step, the path is constrained on the polytope face F given by the vertex of the previous step. This is achieved by projecting $\mathbf{y}$ into a subspace parallel to F to give $\mathbf{r} = (\mathbf{I}-\mathbf{Q})\mathbf{y}$ where $\mathbf{Q} = \frac{\mathbf{K}[:,i]\mathbf{K}[:,i]'}{\|\mathbf{K}[:,i]\|^2}$. Since $\boldsymbol{\alpha} = \mathbf{K}[:,i]'^{\dagger}\mathbf{y}$ and $\hat{\mathbf{y}} = \mathbf{K}[:,i]\boldsymbol{\alpha}$, it follows that $\mathbf{r} = \mathbf{y} - \mathbf{K}[:,i]\boldsymbol{\alpha} = \mathbf{y} - \hat{\mathbf{y}}$ meaning that $\mathbf{r}$ is the residual from the approximation at step i. The second step, which is where the main difference between OMP and PFP arises, involves projecting within the

face F that has just been found, rather than from the origin. This is done by projecting along the residual $\mathbf{r}$. Therefore to find the next face at each step, the maximum *scaled* correlation is found

$$\mathbf{i}_i = \underset{i \in \{1,\ldots,n\} \setminus \mathbf{i}}{\arg\max} \frac{\tilde{\mathbf{K}}[:,i]'\mathbf{r}}{(1 - \tilde{\mathbf{K}}[:,i]'\mathbf{c})} \tag{3.13}$$

where bases are only considered such that $\tilde{\mathbf{K}}[:,i]'\mathbf{r} > 0$.

Constraints are then removed that violate the condition that $\tilde{\boldsymbol{\alpha}}$ contains any negative entries. This is achieved by finding $j \in \mathbf{i}$ such that $\tilde{\boldsymbol{\alpha}}_j < 0$, removing j from $\mathbf{i}$ and removing the face from the current solution. $\tilde{\boldsymbol{\alpha}}$ is then recalculated, continuing until $\boldsymbol{\alpha}_j \geq 0, \forall\, j$. Although this step is necessary to provide exact solutions to (2.86), it may be desirable in some circumstances to remove this step due to the fact that the primal space is in fact the dual space of an RKHS. This would result in faster iterations but less sparse solutions. In Section 3.3.2, a comparison of the performance of the algorithm with and without this step (KPFP and KPFPv respectively) is made. The full algorithm is given in Algorithm 4.

Algorithm 4 Kernel Polytope Faces Pursuit

Input: kernel $\mathbf{K}$, sparsity parameter $k > 0$, training outputs $\mathbf{y}$

Initialise $\tilde{\mathbf{K}} = [\mathbf{K}, -\mathbf{K}]$, $\tilde{\boldsymbol{\alpha}} = [\,]$, $\boldsymbol{\alpha} = [\,]$, $\hat{\mathbf{y}} = \mathbf{0}$, $\tilde{A} = [\,]$, $\mathbf{r} = \mathbf{y}$, $\mathbf{c} = \mathbf{0}$
for $i = 1$ to k **do**
  Find face $\mathbf{i}_i = \arg\max_{i \notin \mathbf{i}} \tilde{\mathbf{K}}[:,i]'\mathbf{r}/(1 - \tilde{\mathbf{K}}[:,i]'\mathbf{c})$ where $\tilde{\mathbf{K}}[:,i]'\mathbf{r} > 0$
  Add constraint: $\tilde{\mathbf{A}} = [\tilde{\mathbf{A}}, \tilde{\mathbf{K}}[:,\mathbf{i}_i]]$
  Update $\mathbf{B} = (\tilde{\mathbf{A}})^\dagger$, $\tilde{\boldsymbol{\alpha}} = \mathbf{B}\mathbf{y}$
  (Optional) Release violating constraints:
  while $\exists\, \tilde{\boldsymbol{\alpha}}_j < 0, \forall\, j$ **do**
    Remove face j: $\tilde{\mathbf{A}} = \tilde{\mathbf{A}} \setminus \tilde{\mathbf{K}}[:,j]$, $\mathbf{i} = \mathbf{i} \setminus \{j\}$
    Update $\mathbf{B} = \tilde{\mathbf{K}}[:,\mathbf{i}_i]^\dagger$, $\boldsymbol{\alpha} = \mathbf{B}\mathbf{y}$
  end while
  Set $\mathbf{c} = \mathbf{B}'\mathbf{1}$, $\hat{\mathbf{y}} = \tilde{\mathbf{A}}\tilde{\boldsymbol{\alpha}}$, $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}$
end for
Calculate $\boldsymbol{\alpha}_i = \tilde{\boldsymbol{\alpha}}_i - \tilde{\boldsymbol{\alpha}}_{i+m},\ i = 1,\ldots,m$

Output: final set $\mathbf{i}$, (sparse) dual weight vector $\boldsymbol{\alpha}$, predicted outputs $\hat{\mathbf{y}}$

3.3.1 Generalisation error bound

For the generalisation error bound it is assumed that the data are generated i.i.d. from a fixed but unknown probability distribution D over the joint space $\mathcal{X} \times \mathcal{Y}$. Given the *true error* of a function f:

$$\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x},y)\sim D}\left[\mathcal{L}(f(\mathbf{x}), y)\right],$$

where $\mathcal{L}(\hat{y}, y)$ is the loss between the predicted $\hat{y}$ and true y, the *empirical risk* of f given S:

$$\hat{\mathcal{R}}(f) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(f(\mathbf{x}_i), y_i)$$

and the estimation error $\mathrm{est}(f)$

$$\mathrm{est}(f) = |\mathcal{R}(f) - \hat{\mathcal{R}}(f)|,$$

the aim is to find an upper bound for $\mathrm{est}(f)$. In order to construct this bound we can use Vapnik-Chervonenkis (VC) theory, which relies on the uniform convergence of the empirical risk to the true risk. For a general function class, a well known quantity to measure its size, which determines the degree of uniform convergence, is the *covering number* [94]. The covering number is calculated by discretising the parameter space so that the risk can be estimated at discrete locations.

Definition Let B be a metric space with metric p. Given observations $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$, and functions $f \in B^m$ that form a hypothesis class $\mathcal{H}$, the covering number in the ℓ_p-norm, as denoted by $\mathcal{N}_p(\epsilon, \mathcal{H}, \mathbf{X})$, is defined as the minimum number z of a collection of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_z \in B^m$, such that $\exists\, \mathbf{v}_j$:

$$\left\| \rho\left(f(\mathbf{x}), \mathbf{v}_j\right) \right\|_p \leq m^{1/p} \epsilon,$$

and further that $\mathcal{N}_p(\epsilon, \mathcal{H}, m) = \sup_{\mathbf{X}} \mathcal{N}_p(\epsilon, \mathcal{H}, \mathbf{X})$.

Note that from the definition and Jensen's inequality, we have that $\mathcal{N}_p \leq \mathcal{N}_q$ for $p \leq q$ (see [95] for a discussion), meaning that the ℓ_∞ covering number is always an upper bound on the ℓ_1 covering number. A result that is relevant here (Theorem 17.1 from [96]) bounds the rate of uniform convergence of a function class in terms of its covering number, (using the ℓ_∞ covering number as opposed to the ℓ_1 covering number):

$$\Pr\left\{ \exists f \in \mathcal{H} : |\mathcal{R}(f) - \hat{\mathcal{R}}(f)| \geq \epsilon \right\} \leq 4\, \mathcal{N}_\infty\left(\frac{\epsilon}{16}, \mathcal{H}, 2m\right) \exp\left(\frac{-\epsilon^2 m}{32}\right),$$

This covering number can be upper bounded using Theorem 12.2 from [96]:

$$\mathcal{N}_\infty(\epsilon, \mathcal{H}, m) \leq \left(\frac{emR}{\epsilon d}\right)^d,$$

where R is the support of the distribution and d denotes the *pseudo-dimension*. As with KMP [88], KPFP also has VC-dimension (pseudo-dimension) k, when k is the number of basis vectors chosen. However, in contrast to the KMP bound of [88] the pseudo-dimension is used to apply a natural regression loss function, the so-called squared error as defined in Section 2.1.3:

$$\mathcal{L}(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2.$$

Therefore there is no need to fix a bandwidth parameter as was the case with the bound of [88] *i.e.* , there is no need to map the regression loss into a classification one. The proof technique of [88] is followed but instead the sample compression technique is applied over pseudo-dimension bounds, which results in a slightly more involved proof.

Theorem 3.3.1. *Let $f \in \mathcal{H} : \mathbf{X} \mapsto [0, 1]$ be the function output by any sparse (dual) kernel regression algorithm which builds regressors using basis vectors, m the size of the training set S and k the size of the chosen basis vectors* $\mathbf{i}$*. Let $\bar{S} = S \setminus S_{\mathbf{i}}$ denote the examples outside of the set $S_{\mathbf{i}}$. Assume without loss of generality that the last k examples in S form the set $S_{\mathbf{i}}$. Let R be the radius of the ball containing*

the support of S, then with $1-\delta$ confidence the true error $\mathcal{R}(f)$ of function f given any training set S can be upper bounded by,

$$\mathcal{R}(f) \leq \quad \hat{\mathcal{R}}_{\bar{S}}(f) + \frac{\sqrt{32^2 + 128(m-k)\left(k\ln\frac{em}{k} + k\ln 32e(m-k)R + 1 + \ln\frac{4km}{\delta}\right)} - 32}{2(m-k)}.$$

Proof. First consider a fixed k and a fixed set of indices $\mathbf{i}$. Assume that the first $m-k$ points from S are drawn independently and apply Theorem 17.1 (and Theorem 12.2) from [96] to obtain the bound

$$\Pr\left\{\bar{S} : |\mathcal{R}(f) - \hat{\mathcal{R}}_{\bar{S}}(f)| \geq \epsilon\right\} \leq 4\left(\frac{32e(m-k)R}{\epsilon k}\right)^k \exp\left(\frac{-\epsilon^2(m-k)}{32}\right). \tag{3.14}$$

Given that the goal is to choose k basis vectors from m choices, there are $\binom{m}{k}$ different ways of selecting them. Multiplying the r.h.s. of Equation 3.14 by $\binom{m}{k}$ like so:

$$\begin{aligned}\Pr\left\{S : \exists\mathbf{i}, |\mathbf{i}| = k, \exists f \in \operatorname{span}\{S_\mathbf{i}\} \;\text{ s.t. } |\mathcal{R}(f) - \hat{\mathcal{R}}_{\bar{S}}(f)| \geq \epsilon\right\} & \\ \leq 4\binom{m}{k}\left(\frac{32e(m-k)R}{\epsilon k}\right)^k \exp\left(\frac{-\epsilon^2(m-k)}{32}\right), & \\ \leq 4\left(\frac{em}{k}\right)^k\left(\frac{32e(m-k)R}{\epsilon k}\right)^k \exp\left(\frac{-\epsilon^2(m-k)}{32}\right), &\end{aligned} \tag{3.15}$$

where we use the fact that $\binom{m}{k} \leq \sum_{i=0}^{k}\binom{m}{i} \leq \left(\frac{em}{k}\right)^k \rightarrow \ln\binom{m}{k} \leq k\ln\frac{em}{k}$. Next by setting the r.h.s. of Equation (3.15) to δ, taking logarithms and rearranging gives

$$\frac{\epsilon^2(m-k)}{32} = k\ln\frac{em}{k} + k\ln 32e(m-k)R - \ln\epsilon + \ln k + \ln\frac{4}{\delta}.$$

It would be desirable to write this bound in terms of ϵ and we therefore use the following result [97] which states that for any $\alpha > 0$, $\ln\epsilon \leq \ln\frac{1}{\alpha} - 1 + \alpha\epsilon$. Substituting this result with $\alpha = 1$ (a smaller α can be used but would make the bound less neat) gives

$$\epsilon^2(m-k) = 32\left(k\ln\frac{em}{k} + k\ln 32e(m-k)R - \ln 1 + 1 - \epsilon + \ln k + \ln\frac{4}{\delta}\right),$$

which yields the following quadratic equation:

$$(m-k)\epsilon^2 + 32\epsilon - 32\left(k\ln\frac{em}{k} + k\ln 32e(m-k)R + 1 + \ln\frac{4k}{\delta}\right) = 0.$$

Therefore, solving for ϵ gives the result when the bound is further applied m times for each value of k. [6] □

This bound can be specialised to the RBF kernel that uses the mean squared error loss and for which the support of the distribution $R = 1$, which leads to the following corollary. [7]

[6] The quadratic equation is solved only for the positive quadrant.
[7] The RBF kernel was used in the experiments.

Corollary 3.3.2. *For a RBF kernel and using all the definitions from Theorem 3.3.1 the loss of KPFP can be upper bounded by:*

$$\mathcal{R}(f) \leq \hat{\mathcal{R}}_{\bar{S}}(f) + \frac{\sqrt{32^2 + 128(m-k)\left(k \ln \frac{em}{k} + k \ln 32e(m-k) + 1 + \ln \frac{4km}{\delta}\right)} - 32}{2(m-k)},$$

where

$$\hat{\mathcal{R}}_{\bar{S}}(f) = \frac{1}{m-k} \sum_{i=1}^{m-k} \mathcal{L}_{\bar{S}}(f(\mathbf{x}_i), y_i).$$

Remark The consequences of Theorem 3.3.1 (and Corollary 3.3.2) is that although the pseudo-dimension can be infinite even in cases where learning is successful,[8] a bound will be generated that is *always* finite. Also, this is the first bound for KMP and KPFP to use the *natural regression loss* in order to upper bound generalisation error. The bound is naturally trading off empirical error with complexity – as the training error decreases the bound gets smaller, and as the number of basis vectors (complexity) increase the bound gets larger. A good trade-off is to find small training error whilst using a small number of basis vectors. Clearly, the KMP and KPFP algorithms try to optimise this trade-off, and the bound suggests that this will result in good generalisation.

It is quite obvious that the output of the function class $\mathcal{H} : \mathbf{X} \mapsto [0, 1]$ is not bounded between 0 and 1 in most 'real world' regression scenarios. Therefore, a more practically useful bound can be given for a function class $\mathcal{H} : \mathbf{X} \mapsto [-B, B]$ where the outputs are bounded in the range of $[-B, B] \in \mathbb{R}$.

Corollary 3.3.3. *Let* $\|\mathbf{w}\|_2 \leq B \in \mathbb{R}$ *and* $\|\mathbf{x}_i\|_2 \leq 1, i = 1, \ldots, m$. *Let* $f \in \mathcal{H} : \mathbf{X} \mapsto [-B, B]$ *be the function output by any* sparse (dual) kernel regression *algorithm which builds regressors using basis vectors,* m *the size of the training set* S *and* k *the size of the chosen basis vectors* $\mathbf{i}$. *Let* $\bar{S} = S \setminus S_{\mathbf{i}}$ *denote the examples outside of the set* $S_{\mathbf{i}}$. *Assume without loss of generality that the last* k *examples in* S *form the set* $S_{\mathbf{i}}$. *Let* R *be the radius of the ball containing the support of* S, *then with* $1 - \delta$ *confidence the true error* $\mathcal{R}(f)$ *of function* f *given any training set* S *can be upper bounded by,*

$$\mathcal{R}(f) \leq \hat{\mathcal{R}}_{\bar{S}}(f) + 2B\frac{\sqrt{32^2 + 128(m-k)\left(k \ln \frac{em}{k} + k \ln 32e(m-k)R + 1 + \ln \frac{4km}{\delta}\right)} - 32}{2(m-k)}.$$

Proof. Denote the function class $\tilde{\mathcal{H}} = \left\{\frac{f+B}{2B} : f \in \mathcal{H}\right\} : \mathbf{X} \mapsto [0, 1]$. Therefore, given any function $\tilde{f} \in \tilde{\mathcal{H}}$ Theorem 3.3.1 holds. Furthermore, for any function class $\mathcal{H} : \mathbf{X} \mapsto [-B, B]$ the following results:

$$\mathcal{R}(\tilde{f}) \leq 2B \cdot \hat{\mathcal{R}}_{\bar{S}}(\tilde{f}) + 2B\frac{\sqrt{32^2 + 128(m-k)\left(k \ln \frac{em}{k} + k \ln 32e(m-k)R + 1 + \ln \frac{4km}{\delta}\right)} - 32}{2(m-k)},$$

which completes the proof under the substitution $\hat{\mathcal{R}}_{\bar{S}}(f) = 2B \cdot \hat{\mathcal{R}}_{\bar{S}}(\tilde{f})$. □

[8] Note that the pseudo-dimension is a generalisation of the VC-dimension and hence the same problems of infinite VC-dimension also apply to the pseudo-dimension.

3.3.2 Experiments

A comparison on 9 benchmark datasets derived from the UCI, StatLib, and DELVE benchmark repositories is presented. Details of the datasets are given in Table 3.3. The performance of KPFP, KMP, KRR and KBP are analysed using RBF kernels. KBP was implemented by solving the LASSO on the features defined by the RBF kernel using the LARS. 10 randomised splits into training and test sets were used. For each of the datasets CV was used to select the optimal RBF kernel width parameter for KRR. This kernel was then used as input to the KMP, KBP and KPFP algorithms. For both KMP and KPFP the initial sparsity level k was set in training by a heuristic method to the lesser of 100 or the number of training examples. The means and standard deviations of the generalisation error for each method and dataset are given in Table 3.4.

The results show that overall the sparse methods (KMP, KPFP, KBP) all perform better than KRR. It is interesting to compare the performance of KPFP with and without the release of violating constraints (KPFPv and KPFP respectively). KPFPv performs nearly as well as KMP on all datasets except for `cpusmall`, whilst requiring fewer bases in the final solutions. On the other hand, KPFP results in solutions that are the least sparse of the three methods, but results in the lowest generalisation error. KBP which gives an exact solution to the LASSO problem performs the worst here, showing that the ℓ_1 solution is not necessarily the optimal one for generalisation. The key to the performance of all of these methods is in selecting the appropriate stopping point k. This is quite difficult to achieve in KMP, as the algorithm tends to overfit quite quickly, and there is no obvious criterion for stopping. For example, if cross-validation were used to select k, the resulting value would be too low, as the number of bases would be selected from a smaller validation set. In the experiments it was found that by selecting an initial k through a heuristic method and then choosing the minimiser of the training error resulted in the best compromise. In KPFP and KPFPv the optimal value for k is more easily achieved, as the training and test error curves tend to follow each other quite well. Additionally there is an (optional) stopping parameter θ_{max}. In fact, the value of θ to which θ_{max} is compared also follows the error curves. It was found that by taking the minimiser of θ as the number of bases was a reliable way of estimating k.

Dataset	# examples	# dimensions
abalone	4177	8
bodyfat	252	14
cpusmall	8192	12
housing	506	13
mpg	392	7
mg	1385	6
pyrim	74	27
space_ga	3107	6
triazines	186	60

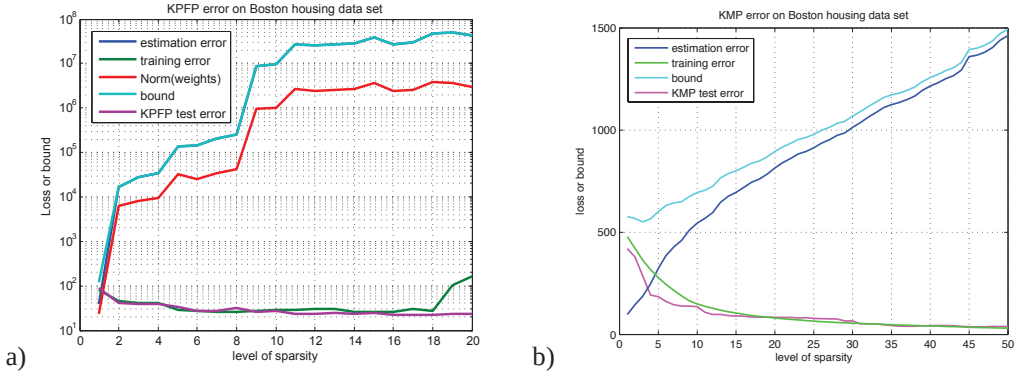
Table 3.3: *Number of examples and dimensions of each of the 9 benchmark datasets*

Dataset	KRR		KMP			KBP			KPFPv			KPFP		
	μ	σ	μ	σ	k	μ	σ	k	μ	σ	k	μ	σ	k
abalone	8.70	1.79	5.70	2.56	49.2	21.64	28.80	5.4	6.07	1.16	7.3	**4.82**	0.24	37.7
bodyfat	0.00	0.00	0.00	0.00	49.1	0.01	0.02	5.7	0.00	0.00	30.1	**0.00**	0.00	129.7
cpusmall	216.35	64.04	15.66	2.51	24.0	519.06	95.45	10.3	69.97	2.51	13.4	**12.50**	1.51	54.2
housing	72.19	19.59	**21.93**	7.17	50.3	56.84	19.35	8.9	34.16	8.19	21.9	23.22	6.67	150.8
mpg	39.47	24.57	20.70	14.37	50.6	42.05	48.27	7.7	13.11	3.35	11.5	**10.98**	1.97	161.1
mg	0.04	0.01	**0.02**	0.00	49.0	0.11	0.19	4.4	0.02	0.00	7.6	0.02	0.00	48.7
pyrim	0.02	0.01	0.02	0.02	24.3	0.02	0.01	11.6	0.02	0.01	17.8	**0.01**	0.01	39.0
space_ga	0.03	0.01	0.02	0.00	49.9	0.05	0.05	4.8	0.02	0.00	6.0	**0.02**	0.00	38.2
triazines	0.02	0.01	0.03	0.02	50.9	0.02	0.00	11.3	**0.02**	0.00	34.4	0.02	0.00	109.7
wins	3		34			6			9			39		

Table 3.4: *(Mean) Mean Squared Error (MMSE) (μ) and SDs (σ) for 9 benchmark datasets for KRR, KMP, KBP and KPFP with and without violation release (KPFPv, KPFP). The total number of wins over all splits of the data for each algorithm is given in the last row. Numbers in bold indicate the best performing algorithm for each dataset.*

3.3.3 Bound Experiments

Finally results of the performance of the bound will be presented. Figure 3.5 shows typical plots of the bound. For Figure 3.5 (b) the number of training examples chosen was 450 and the number of test examples was 56, with the RBF width parameter set to $\sigma = 0.035$. The bound values tend to fall as basis vectors are added, before rising again as the complexity of the solution rises. Hence the first minimum of the bound value could serve as an appropriate point to stop the algorithm. This is clearly much more efficient than using cross-validation to select the value of k, the number of basis vectors to use. However in the experiments this resulted in stopping too early, resulting in underfitting. Further refinement of the bound may improve its performance in this respect.



a) b)

Figure 3.5: *a) Plot of generalisation error bound for different values of k using RBF kernels for the 'Boston housing' data set. The $\log$ of the generalisation error is shown on the y axis. The plot shows the empirical error of the set $\bar{S}$ (denoted training error, in green), the estimation error (in blue), the norm of the weight vector (in red), the bound value which is calculated from these three values (in cyan), and the generalisation error (in magenta). Note that the empirical error follows the true error very well, which justifies its use in the setting of the sparsity parameter. However the bound value is swamped by the norm of the weight vector (needed according to Corrollory 3.3.3), and as such is not useful. b) The bound values for the KMP algorithm. Note that in this case the bound (which is valid for this algorithm too) is more useful, simply because the norm of the weight vector does not blow up as quickly.*

PFP is a greedy algorithm that approximates the sparse solutions recovered by ℓ_1 regularised least-squares LASSO [60, 61] in a similar way to MP and OMP [93]. The algorithm is based on the geometry of the polar polytope where at each step a basis function is chosen by finding the maximal vertex using a path-following method. The algorithmic complexity is of a similar order to OMP whilst being able to solve problems known to be hard for MP and OMP. In this Section the PFP algorithm was extended

to a kernel version, called KPFP. The utility of this algorithm was demonstrated by providing a novel generalisation error bound which used the natural regression loss and pseudo-dimension in order to upper bound its loss. The experimental results showed that KPFP was competitive against the KMP and KRR.

The next Section will present an alternative to the greedy strategies for the selection of bases presented thus far. It will be shown theoretically and empirically that, surprisingly, it is still possible to learn when the bases are selected at random, providing that certain assumptions hold.

3.4 Learning in a Nyström Approximated Subspace

"No random actions, none not based on underlying principles"

Marcus Aurelius, Meditations Book IV

Given m observations, it is possible to define a framework that carries out learning in a $k \leq \ell \ll m$ dimensional subspace that is constructed using the Nyström method. A recently advocated and theoretically justified approach of uniform sub-sampling without replacement will be adopted to cheaply find a k-dimensional subspace in time complexity $\mathcal{O}(1)$. Any linear learning algorithm can then be used in this uniformly sampled k-dimensional Nyström approximated subspace to help tackle large data sets. Furthermore, for any SVM constructed in this Nyström approximated space an upper bound on its objective function is proved in terms of the objective of the SVM solved in the original space, implying successful learning whenever the objective of the SVM in the original space is small. Finally, the proposed methodology will be demonstrated on several UCI repository datasets for both classification and regression, using primal SVM, FDA, and RR.

Kernel methods continue to play an important role in machine learning due to their ability in addressing real-world problems, which often have non-linear and complex structures. The key element of kernel methods is the mapping of data into a kernel induced Hilbert space where a dot product between the points can be computed efficiently. Therefore, given m sample points, an $m \times m$ symmetric positive semi-definite (SPSD) kernel matrix is all that needs to be computed. Computing the kernel matrix requires an operation with a complexity term of $\mathcal{O}(m^2)$. Despite the obvious advantages of kernel methods, the methodology begins to falter when m becomes very large.

This potential draw back of kernel methods has been addressed in the literature through the proposal of a number of methods for kernel matrix low-rank approximations. These methods have a computational complexity smaller than $\mathcal{O}(m^2)$. In particular, one would perform a low-rank approximation of $\mathbf{K} = \mathbf{C}'\mathbf{C}$, where $\mathbf{C} \in \mathbb{R}^{k \times m}$ such that $k \ll m$. For example, [37] have approximated the kernel matrix by incrementally choosing basis vectors so as to minimise an upper bound on the approximation error. Their algorithm has a complexity of $\mathcal{O}(k^2 m\ell)$ where ℓ is a random subset size. [98] have proposed a greedy sampling scheme, with complexity $\mathcal{O}(k^2 m)$, based on how well a sample point can be represented by a linear combination of the current subspace bases in the feature space. The Nyström approximation, originally proposed by [99] to solve integral equations, was proposed by [7] as a technique to approximate the kernel matrices to speed up kernel-based predictors. The Nyström approach samples

k columns of the kernel matrix to reconstruct the complete kernel matrix, it has a complexity term of $\mathcal{O}(k^3)$. When $k \ll m$ this is computationally much more efficient than the other methods.

It has recently been demonstrated that when approximating the kernel matrix using the Nyström approach, uniform subsampling without replacement is able to outperform other sampling techniques[100]. The authors show that the most computationally efficient, and cheapest, sampling technique is to randomly select columns of the kernel matrix. However whilst they provide upper bounds on the approximation error, they do not give a theoretical analysis of *learnability* in the Nyström subspace.

This question has in fact been investigated by Blum *et. al.* [101, 102] who show that a Nyström projection (their projection F_2, although they do not refer to it as a Nyström projection) preserves margins. By this they mean that if there is a classifier with margin γ, a suitably large Nyström subspace will have margin of at least $\gamma/2$ for a high proportion of the training data. In practice one would not normally expect data to have a large hard margin even in a high dimensional space, but rather have a small primal SVM objective that combines both the margin and the slack variables. Hence, their result leaves open the question of how the projection will affect the size of the SVM objective, since they do not take into account

- some points with non-zero slack variables may fail to achieve margin γ in the original space;
- the size of the slack variables of the fraction ϵ of points that fail to achieve margin $\gamma/2$ in the Nyström projection.

These issues will be investigated, resulting in a theoretical extension of the Blum *et. al.* approach, followed by experiments to verify the effect of the Nyström projection on the quality of generalisation obtained using Support Vector classification.

Section 3.2.1 gave details of a OMP algorithm for KFDA that is greedy in its approach to finding a small number of basis vectors with a complexity of $\mathcal{O}(m^3k)$. MPKFDA greedily chooses basis vectors by maximising the Fisher quotient to solve the FDA algorithm in the Nyström approximated space [15]. The KPFP algorithm described in Section 3.3, which was used to perform regression, has the same complexity [17]. The idea of uniformly sampling (with or without replacement) [100] will be used to generate the Nyström subspace and demonstrated experimentally in both of these settings, as well as for the SVM. The experimental results will be strengthened with significance testing.

Preliminaries

Recall the definition of the Nyström approximation of the Gram matrix $\mathbf{G}$, as defined in Section 3.2.2. For any such Gram matrix, there exists a $\mathbf{X} \in \mathbb{R}^{m \times n}$ such that $\mathbf{G} = \mathbf{X}\mathbf{X}'$. Again if we assume that the examples have already been projected into the kernel defined feature space this analysis will hold for kernel matrices $\mathbf{K}$ in place of the gram matrix $\mathbf{G}$.

$\ell \ll m$ columns of $\mathbf{G}$ are sampled at random uniformly without replacement. Let $\mathbf{N}$ be the $m \times \ell$ matrix of the sampled columns, and $\mathbf{W}$ be the $\ell \times \ell$ matrix consisting of the intersection of these ℓ columns with the corresponding ℓ rows of G. The Nyström method uses $\mathbf{W}, \mathbf{N}$ to construct a rank-k

approximation $\tilde{\mathbf{G}}_k$ to $\mathbf{G}$ for $k \leq \ell$, like so:

$$\tilde{\mathbf{G}}_k = \mathbf{N}\mathbf{W}_k^{\dagger}\mathbf{N}' \approx \mathbf{G}, \tag{3.16}$$

Recent studies [100, 103, 104] have shown that for a Gram matrix $\mathbf{G}$ and a Nyström approximated matrix $\tilde{\mathbf{G}}_k$, constructed from k uniformly sampled columns of $\mathbf{G}$, the expected loss of $\left\|\mathbf{G} - \tilde{\mathbf{G}}_k\right\|_F$ can be bounded by the difference between $\mathbf{G}$ and its optimal k rank approximation $\mathbf{G}_k$.

Theorem 3.4.1. *(Quoted from [100]) Let $\mathbf{G} \in \mathbb{R}^{m\times m}$ be a SPSD matrix. Assume that ℓ columns of $\mathbf{G}$ are sampled uniformly at random without replacement, let $\tilde{\mathbf{G}}_k$ be the rank-k Nyström approximation to $\mathbf{G}$ as described in Equation (3.2), and let $\mathbf{G}_k$ be the best rank-k approximation to $\mathbf{G}$. For $\epsilon > 0$, if $\ell \geq \frac{64k}{\epsilon^4}$, then*

$$\mathbb{E}\left[\left\|\mathbf{G} - \tilde{\mathbf{G}}_k\right\|_F\right] \leq \|\mathbf{G} - \mathbf{G}_k\|_F + \epsilon\left[\left(\frac{m}{\ell}\sum_{i\in D(\ell)}\mathbf{G}_{ii}\right)\left(\sqrt{m\sum_{i=1}^{m}\mathbf{G}_{ii}^2}\right)\right]^{\frac{1}{2}},$$

where $\sum_{i\in D(\ell)}\mathbf{G}_{ii}$ is the sum of the largest ℓ diagonal entries of $\mathbf{G}$. Further, let $\eta = \sqrt{\frac{\log(\frac{2}{\delta})\beta(\ell,m-\ell)}{\ell}}$, with $\beta(\ell, m-\ell) = \left(\frac{\ell(m-\ell)}{m-\frac{1}{2}}\right)\left(\frac{1}{1-\frac{1}{2\max\{\ell,m-\ell\}}}\right)$ and if $\ell \geq \frac{64k}{\epsilon^4}$ then with probability at least $1-\delta$,

$$\left\|\mathbf{G} - \tilde{\mathbf{G}}_k\right\|_F \leq \|\mathbf{G} - \mathbf{G}_k\|_F + \epsilon\left[\left(\frac{m}{\ell}\sum_{i\in D(\ell)}\mathbf{G}_{ii}\right)\left(\sqrt{m\sum_{i=1}^{m}\mathbf{G}_{ii}^2} + \eta\max(m\mathbf{G}_{ii})\right)\right]^{\frac{1}{2}}.$$

3.4.1 Theory of Support Vector Machine (SVM) in Nyström Subspace

The theories for the Nyström approximation have been the following:

- An upper bound on the expected reconstruction of the low rank matrix approximation described above.
- A bound which shows that if there exists a separator with hard margin γ in the original space a Nyström projection of dimension

$$d \geq \frac{8}{\epsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right] \tag{3.17}$$

  will with probability $1-\delta$ over the selection of the d points defining the projection create a margin of at least $\gamma/2$ for all but at most an ϵ fraction of the training data.

The second statement implies the potential for good generalisation since a large margin classifier misclassifying some points has a provable bound on generalisation. Nonetheless it is not clear that this will be found by the margin maximizing SVM, since it deals with margin errors using slack variables that do not simply count margin errors. Furthermore, the assumption that there exists a hard margin separator in the original space is in practice unrealistic. A SVM solution with small objective might

be found, implying good generalisation but at the expense of a number of points with non-zero slack variables. The theorem as stated would not apply to this case.

The main result of this work is an adaptation of [101] as follows.

Lemma 3.4.2. *Consider any distribution over labeled examples (with input vectors having support contained in the unit ball in Euclidean space) such that there exists a linear separator* $\langle \mathbf{w}, \mathbf{x} \rangle = 0$ *with margin* γ *on all but* k *points. Drawing*

$$d \geq \frac{8}{\epsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$$

examples $\mathbf{z}_1, \ldots, \mathbf{z}_d$ *i.i.d. from this distribution, with probability at least* $1-\delta$*, there exists a vector* $\tilde{\mathbf{w}} \in \mathrm{span}(\mathbf{z}_1, \ldots, \mathbf{z}_d)$ *that has error at most* $\epsilon + k/m$ *at margin* $\gamma/2$*.*

Proof. Given the set of examples $S = \{\mathbf{z}_1, \ldots, \mathbf{z}_d\}$ as defined above with $\|\mathbf{z}_i\| = 1, \forall i$, we define $V = \mathrm{span}(S)$ as the (possibly not unique) span of this set, and $V^\perp$ as its orthogonal complement. Suppose we have a (weight) vector $\mathbf{w}$ in the space $\mathrm{span}(\mathbf{z})$ also assumed to be normalised ($\|\mathbf{w}\| = 1$). Let $\mathbf{w}_{in}$ be the part of $\mathbf{w}$ that lies in V, and $\mathbf{w}_{out}$ be the part of $\mathbf{w}$ that lies in $V^\perp$. By definition $\mathbf{w}_{in} \perp \mathbf{w}_{out}$ and $\mathbf{w} = \mathbf{w}_{in} + \mathbf{w}_{out}$.

We need to make the following definitions:

1. $\mathbf{w}_{out}$ is *large* if $\Pr_{\mathbf{z}}(\langle \mathbf{w}_{out}, \mathbf{z} \rangle > \gamma/2) \geq \epsilon$, and
2. $\mathbf{w}_{out}$ is *small* if $\Pr_{\mathbf{z}}(\langle \mathbf{w}_{out}, \mathbf{z} \rangle > \gamma/2) < \epsilon$,

where we use $\Pr_{\mathbf{z}}(\cdot)$ to denote the probability over random sampling from the training set. If $\mathbf{w}_{out}$ is small, then as $\langle \mathbf{w}_{in}, \mathbf{z} \rangle = \langle \mathbf{w}, \mathbf{z} \rangle - \langle \mathbf{w}_{out}, \mathbf{z} \rangle$ and it was assumed that $\Pr_{\mathbf{z}}(\langle \mathbf{w}, \mathbf{z} \rangle > \gamma) = 1 - k/m$, it can be seen that $\Pr_{\mathbf{z}}(\langle \mathbf{w}_{in}, \mathbf{z} \rangle > \gamma/2) \geq 1 - \epsilon - k/m$ as required, and the proof would be complete. For the rest of the proof, we consider the situation where $\mathbf{w}_{out}$ is large, *i.e.* the set $\mathbf{z}$ has not yet been informative enough that the weight vector enabling separation lies sufficiently within its span.

For $\mathbf{w}_{out}$ that is large, we consider what happens when a new (random) point $\tilde{\mathbf{z}} = \mathbf{z}_{d+1}, \|\tilde{\mathbf{z}}\| = 1$ is added to the set, with the resulting induced space $\tilde{S} = S \cup \{\tilde{\mathbf{z}}\}$. Consider the case that $\tilde{\mathbf{z}} \notin V$ (*i.e.* $\tilde{\mathbf{z}} \in \tilde{V^\perp}$ where $\tilde{V^\perp} = \mathrm{span}(\tilde{S})^\perp = V^\perp \cup \{\tilde{\mathbf{z}}\}$). We can by the definition of $\mathbf{w}$ and $\mathbf{w}_{out}$ deduce that $\langle \mathbf{w}_{out}, \tilde{\mathbf{z}} \rangle > \gamma/2$. Let $\tilde{\mathbf{z}}_{in}$ and $\tilde{\mathbf{z}}_{out}$ be the normalised projections of $\tilde{\mathbf{z}}$ onto V and $V^\perp$ respectively. Similarly let $\tilde{\mathbf{w}}_{in} = \mathrm{proj}(\mathbf{w}_{in}, \tilde{V^\perp})$ and $\tilde{\mathbf{w}}_{out} = \mathrm{proj}(\mathbf{w}_{out}, \tilde{V^\perp})$ be the projections of $\mathbf{w}_{in}$ and $\mathbf{w}_{out}$ onto $\tilde{V^\perp}$ respectively. Observe that,

$$\begin{aligned}
\tilde{\mathbf{w}}_{in} &= \mathrm{proj}(\mathbf{w}, \tilde{V}), \\
&= \mathrm{proj}(\mathbf{w}, V) + \mathrm{proj}(\mathbf{w}, \tilde{\mathbf{z}}), \\
&= \mathbf{w}_{in} + \mathrm{proj}(\mathbf{w}, \tilde{\mathbf{z}}). \qquad (3.18)
\end{aligned}$$

Since $\tilde{\mathbf{w}}_{in} \perp \tilde{\mathbf{w}}_{out}$, $\tilde{\mathbf{w}}_{out}$ must shrink by a concordant amount,

$$\begin{aligned}\tilde{\mathbf{w}}_{out} &= \mathbf{w}_{out} - \mathrm{proj}(\mathbf{w}, \tilde{\mathbf{z}}),\\ &= \mathbf{w}_{out} - \mathrm{proj}(\mathbf{w}_{out}, \tilde{\mathbf{z}}),\\ &= \mathbf{w}_{out} - \langle \mathbf{w}_{out}, \tilde{\mathbf{z}} \rangle \tilde{\mathbf{z}}. \end{aligned} \tag{3.19}$$

Since $\tilde{\mathbf{z}} = \mathrm{proj}(\mathbf{z}, \tilde{S})$, and by definition $\tilde{\mathbf{z}} \perp V$, we have

$$\begin{aligned}\|\tilde{\mathbf{w}}_{in}\|^2 &= \langle \mathbf{w}_{in} + \mathrm{proj}(\mathbf{w}, \tilde{\mathbf{z}}), \mathbf{w}_{in} + \mathrm{proj}(\mathbf{w}, \tilde{\mathbf{z}}) \rangle ,\\ &= \|\mathbf{w}_{in}\|^2 + (\mathrm{proj}(\mathbf{w}, \tilde{\mathbf{z}}))^2 + \langle \mathbf{w}_{in}, \mathrm{proj}(\mathbf{w}, \tilde{\mathbf{z}}) \rangle ,\\ &= \|\mathbf{w}_{in}\|^2 + (\mathrm{proj}(\mathbf{w}, \tilde{\mathbf{z}}))^2 ,\\ &= \|\mathbf{w}_{in}\|^2 + (\langle \mathbf{w}, \tilde{\mathbf{z}} \rangle \tilde{\mathbf{z}})^2 ,\\ &= \|\mathbf{w}_{in}\|^2 + (\langle \mathbf{w}, \tilde{\mathbf{z}}_{in} \rangle \tilde{\mathbf{z}}_{in})^2 . \end{aligned} \tag{3.20}$$

and as before the corresponding norm of the orthogonal complement must shrink by a concordant amount,

$$\begin{aligned}\|\tilde{\mathbf{w}}_{out}\|^2 &= \|\mathbf{w}_{out}\|^2 - (\langle \mathbf{w}, \tilde{\mathbf{z}} \rangle \tilde{\mathbf{z}})^2 ,\\ &= \|\mathbf{w}_{out}\|^2 - (\langle \mathbf{w}, \tilde{\mathbf{z}}_{out} \rangle \tilde{\mathbf{z}}_{out})^2 . \end{aligned} \tag{3.21}$$

Using that,

$$\begin{aligned}\langle \mathbf{w}_{out}, \tilde{\mathbf{z}} \rangle &\leq \langle \mathbf{w}_{out}, \tilde{\mathbf{z}}_{out} \rangle ,\\ &= \langle \mathbf{w}_{out}, \mathbf{z} \rangle , \end{aligned} \tag{3.22}$$

and by definition of $\mathbf{z}$, we have,

$$\begin{aligned}\|\tilde{\mathbf{w}}_{out}\|^2 &= \|\mathbf{w}_{out}\|^2 - (\langle \mathbf{w}, \tilde{\mathbf{z}}_{out} \rangle \tilde{\mathbf{z}}_{out})^2 ,\\ &< \|\mathbf{w}_{out}\|^2 - (\gamma/2)^2 . \end{aligned} \tag{3.23}$$

We have therefore shown that the new point $\tilde{\mathbf{z}}$ has at least an ϵ chance of significantly improving the set S by a factor of at least $\gamma^2/4$, under the assumption that $\mathbf{w}_{out}$ is large. Since $\|\mathbf{w}\|^2 = \|\mathrm{proj}(\mathbf{w}, \emptyset)\|^2 = 1$, this can happen at most $4/\gamma^2$ times.

Under the assumptions above, and due to the strict inequality in Equation 3.23, we can then use Chernoff bounds to determine the number of projections d that are needed. The bounds in the multiplicative form state that the probability of independent random events $X_1, X_2, \ldots, X_n$ taking the values

0 or 1,

$$\Pr\left(X \geq (1+\zeta)\,\mathbb{E}[X]\right) < \left(\frac{\exp(\zeta)}{(1+\delta)^{1+\zeta}}\right)^{\mathbb{E}[X]}. \tag{3.24}$$

To use this form we need to switch round the statement above such that our random event is the chance that S will not be improved (*i.e.* $1-(\epsilon/2)$), and we are bounding the probability ζ that over n instantiations the mean value of the random events are larger than $n - n\epsilon/2)$. In this case we have that $\mathbb{E}[X] = n(1-\epsilon)$, and this means that,

$$\begin{aligned}(1+\zeta)n(1-\epsilon) &= n - n\epsilon/2,\\ \Rightarrow \zeta &= \frac{\epsilon/2}{1-\epsilon}.\end{aligned} \tag{3.25}$$

Substituting into Equation (3.24) leads to,

$$\Pr\left(X \geq n\left(1-\frac{\epsilon}{2}\right)\right) \leq \left(\frac{\exp(\zeta)}{(1+\zeta)^{1+\zeta}}\right)^{n(1-\epsilon)} \doteq \delta. \tag{3.26}$$

We now rearrange for n,

$$\begin{aligned}\delta &= \left(\frac{\exp(\zeta)}{(1+\zeta)^{1+\zeta}}\right)^{n(1-\epsilon)},\\ \ln(\delta) &= n(1-\epsilon)\ln\left(\frac{\exp(\zeta)}{(1+\zeta)^{1+\zeta}}\right),\\ \ln\frac{1}{\delta} &= n(1-\epsilon)\ln\left(\frac{(1+\zeta)^{1+\zeta}}{\exp(\zeta)}\right),\\ \ln\frac{1}{\delta} &= n(1-\epsilon)\left[(1+\zeta)\ln(1+\zeta)-\zeta\right],\\ n &= \frac{1}{1-\epsilon}\frac{\ln\frac{1}{\delta}}{\left[(1+\zeta)\ln(1+\zeta)-\zeta\right]}.\end{aligned} \tag{3.27}$$

Substituting (3.25) into (3.27) gives us,

$$\begin{aligned}n &= \frac{1}{1-\epsilon}\frac{\ln\frac{1}{\delta}}{\left[(1+\frac{\epsilon/2}{1-\epsilon}\ln(1+\frac{\epsilon/2}{1-\epsilon}) - \frac{\epsilon/2}{1-\epsilon}\right]},\\ &= \frac{1}{1-\epsilon}\frac{(1-\epsilon)\ln\frac{1}{\delta}}{\left[(1-\epsilon-\epsilon/2)\ln(1+\frac{\epsilon/2}{1-\epsilon}-\epsilon/2\right]},\\ &= \frac{\ln\frac{1}{\delta}}{(1-\epsilon/2)\ln(1+\frac{\epsilon/2}{1-\epsilon})-\epsilon/2}.\end{aligned} \tag{3.28}$$

We will now pull together the result from (3.23) with the above to lower bound the number of projection dimensions n. Setting τ to be the denominator in Equation (3.28), we can use the fact that $\tau \leq \frac{8}{\epsilon}$ for $\epsilon \in [0, 0.5]$, as shown in Figure 3.6, together with the consequence from (3.23) that is that with

probability $1-\delta$ we will have at least $n\epsilon/2 \geq 4/\gamma^2$ heads (implying that $n \geq \frac{8}{\gamma^2\epsilon}$) as follows,

$$\begin{aligned} n &\geq \max\left\{\frac{8}{\epsilon}\ln\frac{1}{\delta}, \frac{8}{\gamma^2\epsilon}\right\}, \\ &\geq \frac{8}{\epsilon}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]. \end{aligned} \tag{3.29}$$

□

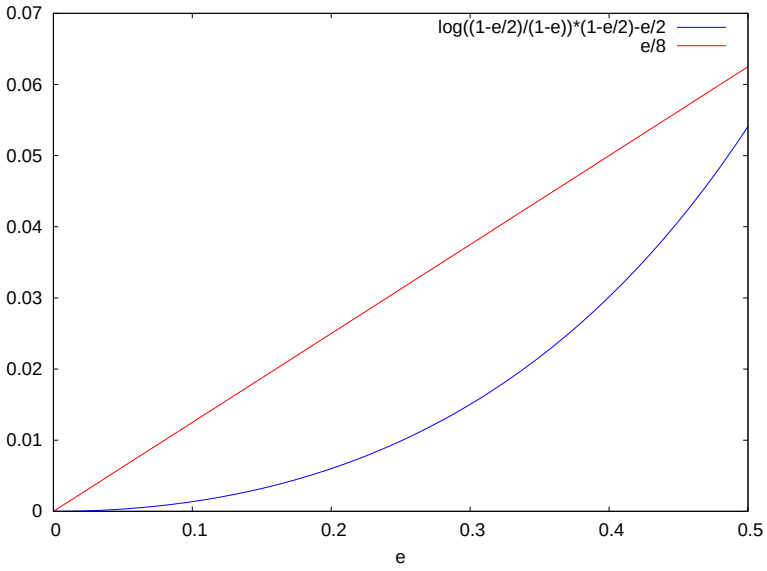


Figure 3.6: *Plot of* $f(\epsilon) = (1-\epsilon/2)\ln(1+\frac{\epsilon/2}{1-\epsilon}) - \epsilon/2$ *and* $f(\epsilon) = \frac{8}{\epsilon}$ *for* $\epsilon \in \{0, 0.5\}$

We now extend this to the soft margin case with the following corollary. We use the fact that the analysis still holds if some of the points fail to attain the margin.

Corollary 3.4.3. *Given a soft margin seperator characterised by a margin* γ *and slack variables* $\boldsymbol{\xi}$ *in the original space, where* $|\boldsymbol{\xi}| = k$*, then a Nyström projection of dimension*

$$d \geq \frac{8}{\epsilon - k/m}\left[\frac{1}{\gamma^2} + \ln\frac{1}{\delta}\right]$$

will with probability $1-\delta$ *over the selection of the* d *points defining the projection create a margin of at least* $\gamma/2$ *for all but at most an* $\epsilon + k/m$ *fraction of the training data. In particular, the objective of a support vector optimisation in the Nyström space is bounded by*

$$\frac{4}{\gamma^2} + \frac{2(k+\epsilon m)}{\gamma}.$$

If we now minimise in the new space the objective $\|\mathbf{w}\|_2^2 + C\sum_i \xi_i$ can only increase.

In the following Section the proposed methodology will be explored empirically for both classification and regression.

3.4.2 Experiments: Classification

Firstly, a comparison on 13 benchmark datasets derived from the UCI, DELVE and STATLOG benchmark repositories is presented, which are all binary classification problems or converted such that they are. The performance of KFDA, SVM, Nyström KFDA (NFDA) and Nyström SVM (NSVM) are compared. Results are also included for Matching Pursuit Kernel Fisher Discriminant Analysis (MPKFDA) as presented earlier in Section 3.2.1 [15], which was trained on the same benchmark datasets using the same splits. Radial Basis Function (RBF) kernels were used for all experiments. The data comes in 100 predefined splits into training and test sets (20 in the case of the image and splice datasets) as described in [34] [9]. For each of the datasets two rounds of cross-validation (CV) were used to select the optimal parameters (the RBF kernel width parameter, the C parameter in the SVM, and k the number of iterations in NFDA). For the first round a coarse range of parameter values was evaluated on the first 5 splits of the training set, with the parameter value corresponding to the median of the lowest error of the five splits being chosen for the second round. A fine range of parameters was constructed around this value, from which the optimal value was chosen using 5-fold CV over all splits of the training set.This way of estimating the parameters leads to more robust comparisons between the methods.

The sparsity parameter k for both NFDA and NSVM were set to $2\sqrt{n_{trn}}$ as this is justified by the upper bound. Previously [100] had selected k as 20% of the dataset but in cases of large m, this could result in a complexity worse than the SVM (which is $\mathcal{O}(n^2)$). The means and Standard Deviations (SDs) of the generalisation error for each method and dataset are given in Tables 3.5 for the SVM and 3.6 for FDA.

From casual examination of the data, it can be seen that although the SVM performs best in most situations (followed by KFDA), the differences are not large. Additionally, the differences between NFDA and MPKFDA are even smaller. This is somewhat surprising, as MPKFDA is much more expensive to compute ($\mathcal{O}(kn^3)$), and at each step is supposedly finding an "optimal" basis (according to the Fisher ratio). Two-sided heteroscedastic t-tests were performed to test whether the null hypothesis that the results for the SVM versus the NSVM, KFDA versus NFDA, and MPKFDA versus NFDA were drawn from the same normal distributions. All of these tests were insignificant ($p = 0.37$, $p = 0.39$ and $p = 0.42$ respectively) which means that under the assumptions of the test the null hypothesis cannot be rejected. This means that the differences between the results are not significant. Note also that the solutions given by the NSVM are much more sparse (in the dual sense) than the SVM solutions, and that the solutions given by NFDA have a comparable degree of sparsity with those given by MPKFDA.

Furthermore we compare in figures 3.7 and 3.8, for the Breast Cancer data set and Flare Solar dataset respectively, the error and computational cost as a function of k for Nyström as compared with KFDA.

[9] Available to download from: `http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm`

	SVM			NSVM		
Dataset	error	SD	SVs	error	SD	k
Banana	0.1061	0.01	76.4	0.1195	0.01	40
Breast Cancer	0.2584	0.05	58.1	0.2684	0.04	28
Diabetes	0.2367	0.02	168.8	0.2350	0.02	43
Flare Solar	0.3334	0.02	338.9	0.3361	0.02	52
German	0.2365	0.02	208.2	0.2415	0.02	54
Heart	0.1564	0.03	68.5	0.1677	0.03	26
Image	0.0061	0.00	216	0.0536	0.02	72
Ringnorm	0.0176	0.00	67.7	0.0190	0.00	40
Splice	0.1102	0.01	336.6	0.1618	0.01	63
Thyroid	0.0415	0.02	7.6	0.0532	0.03	24
Titanic	0.2243	0.01	48.3	0.2346	0.02	24
Twonorm	0.0275	0.00	48.7	0.0296	0.00	40
Waveform	0.0999	0.00	112.5	0.1070	0.00	40
Overall:	0.1426	0.01	146.3	0.1559	0.01	42

Table 3.5: *Generalization error estimates and Standard Deviations (SDs) for 13 benchmark datasets for the SVM, Nyström SVM (NSVM)*

	KFDA		NFDA			MPKFDA		
Dataset	error	SD	error	SD	k	error	SD	k
Banana	0.1056	0.00	0.1072	0.01	40	0.1101	0.01	31
Breast Cancer	0.2892	0.04	0.3104	0.11	28	0.3174	0.04	19
Diabetes	0.2505	0.02	0.2548	0.02	43	0.2543	0.02	18
Flare Solar	0.3423	0.02	0.3471	0.03	52	0.3457	0.02	19
German	0.2643	0.01	0.2784	0.02	54	0.2808	0.02	27
Heart	0.1638	0.03	0.1613	0.03	26	0.1599	0.03	13
Image	0.0273	0.01	0.0571	0.01	72	0.0136	0.03	39
Ringnorm	0.0152	0.00	0.0179	0.00	40	0.0573	0.03	15
Splice	0.1203	0.01	0.1710	0.03	63	0.0314	0.06	37
Thyroid	0.0483	0.02	0.0600	0.03	24	0.0699	0.03	29
Titanic	0.2319	0.01	0.2478	0.02	24	0.2468	0.05	7
Twonorm	0.0261	0.00	0.0260	0.00	40	0.0253	0.00	14
Waveform	0.0983	0.00	0.1042	0.01	40	0.1027	0.00	13
Overall:	0.1525	0.01	0.1648	0.02	42	0.1550	0.02	21.61

Table 3.6: *Generalization error estimates and Standard Deviations (SDs) for 13 benchmark datasets for the KFDA, NFDA, and MPKFDA*

Note that in these two examples, as with all of the other datasets we tested, a very small proportion of basis vectors is required for good generalisation error, and that the computational cost for these values of k is of an order of magnitude less than standard KFDA.

In the experiments the Nyström classifiers were roughly an order of magnitude faster than the kernel equivalents during training for the smaller datasets, and roughly two orders of magnitude faster for the larger datasets. This is born out by the fact that the complexity of both algorithms was $\mathcal{O}({n_{trn}}^{1.5})$ due to the method for choosing k that was used.

3.4.3 Experiments: Regression

Next, results on 7 benchmark regression datasets derived from the UCI, StatLib, and DELVE benchmark repositories will be presented. The performance of KRR and Nyström KRR (NRR) along with KMP

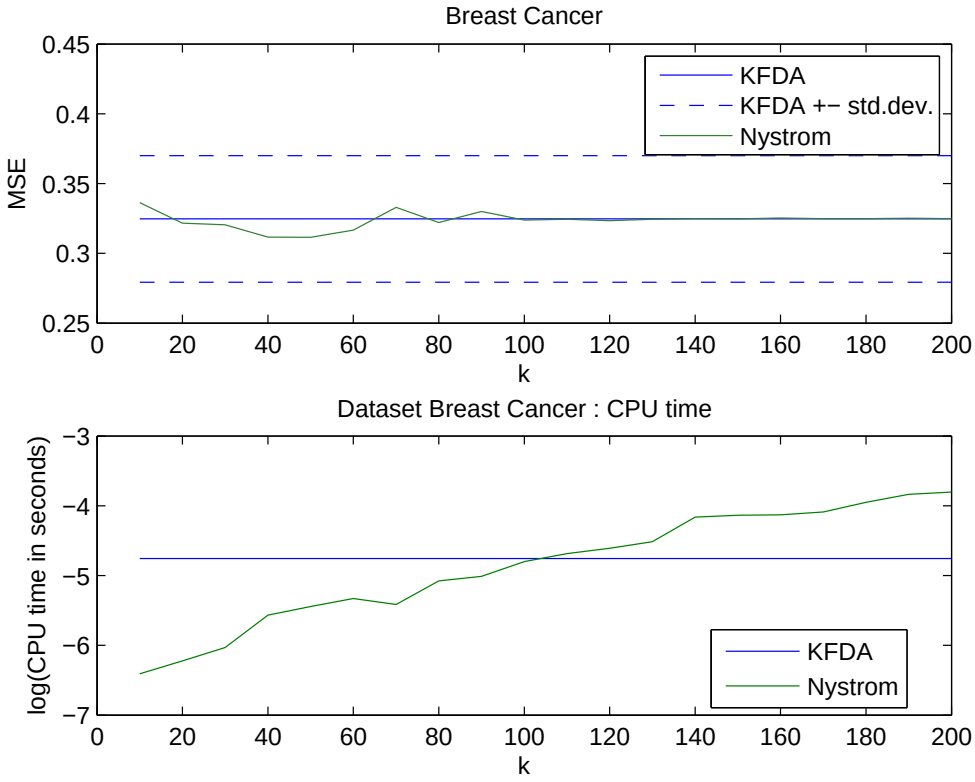


Figure 3.7: *Classification error (and log run-time) as a function of k for the 'Breast Cancer' dataset as achieved by NFDA, KFDA.*

				KRR		NRR		KMP	
Dataset	# ex	# dim	k	MMSE	SD	MMSE	SD	MMSE	SD
bodyfat	252	14	30	0.0000	0.0000	0.0000	0.0000	0.0156	0.0012
housing	506	13	43	11.0323	4.8159	24.2497	9.6876	82.9434	29.2603
mpg	392	7	38	7.3085	2.9387	10.0276	2.6627	47.5519	12.8119
mg	1385	6	71	0.0144	0.0008	0.0177	0.0030	0.0467	0.0159
pyrim	74	27	16	0.0057	0.0124	0.0124	0.0130	0.0514	0.0130
space_ga	3107	6	106	0.0107	0.0030	0.0100	0.0039	0.0261	0.0026
triazines	186	60	26	0.0202	0.0094	0.0242	0.0103	0.0308	0.0073

Table 3.7: *(Mean) Mean Squared Error (MMSE) and Standard Deviation (SD) for 7 benchmark datasets for Kernel Ridge Regression (KRR), Nyström KRR (NRR), and KMP.*

were analysed again using RBF kernels. The comparison against KMP was included as it is a state-of-the-art method for greedily selecting basis functions. 10 randomized splits into training and test sets were used. For each of the datasets two rounds of CV were again used to select the optimal RBF kernel width parameter for each of the algorithms and the regularization parameter μ in KRR and NRR. For both KMP and KPFP the sparsity parameter k was set using the same method as for the classification experiments, *i.e.* $k = 2\sqrt{n_{trn}}$. Note that this method of choosing k is by no means optimal for KMP (or NRR for that matter), but in the absence of a more robust heuristic this avoids costly CV (as with MPKFDA, the complexity of KMP is $\mathcal{O}(km^3)$). The means and standard deviations of the MMSE for each method and dataset are given in Table 3.7.

The results show that although NRR does not perform as well as KRR, for the same choice of k

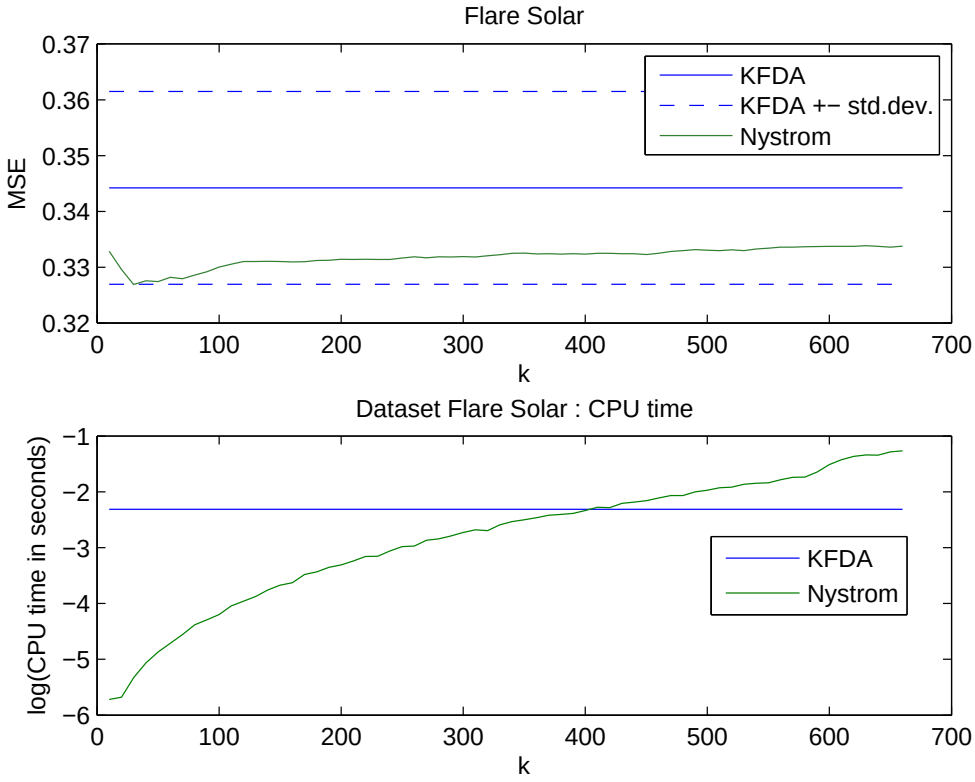


Figure 3.8: *Classification error (and log run-time) as a function of k for the 'Flare Solar' dataset as achieved by NFDA, KFDA.*

it comfortably outperforms KMP. Our observations were that poor performances of NRR and KMP on $housing$ and mpg were caused by overfitting, indicating that the heuristic method for choosing k should not be relied upon. Overall the results demonstrate that the Nyström method can be successfully applied to regression as well as classification.

Remark: Greedy versus random sampling. The theoretical and empirical analyses given above serve to demonstrate that greedy methods for sparse selection of basis vectors are extremely powerful and can often outperform standard ℓ_1 methods for enforcing sparsity, both in terms of generalisation error and also in terms of the sparsity of solutions. However it is also clear that by simply choosing basis vectors at random it is still possible to learn effectively, whilst of course this method is significantly cheaper. It therefore comes down to a trade-off between exactness of solutions and computational resources. If a slightly sub-optimal solution is sufficient for the application, then the Nyström method provides a simple way of providing sparse solutions in a computationally efficient way. However if the best possible sparse solution is sought, greedy methods such as OMP and PFP provide solutions that closely approximate (and in many cases achieve) the best possible ℓ_0 pseudo-norm solutions (as introduced in Section 3.5.2).

In the next Section the attention is turned to the problem of learning from multiple data sources or views (MSL and MVL respectively). There is certainly opportunity for a synthesis between the methods presented above and those presented below, but this is outside of the present scope. A discussion of possibilities for such a synthesis will be presented in Chapter 6.

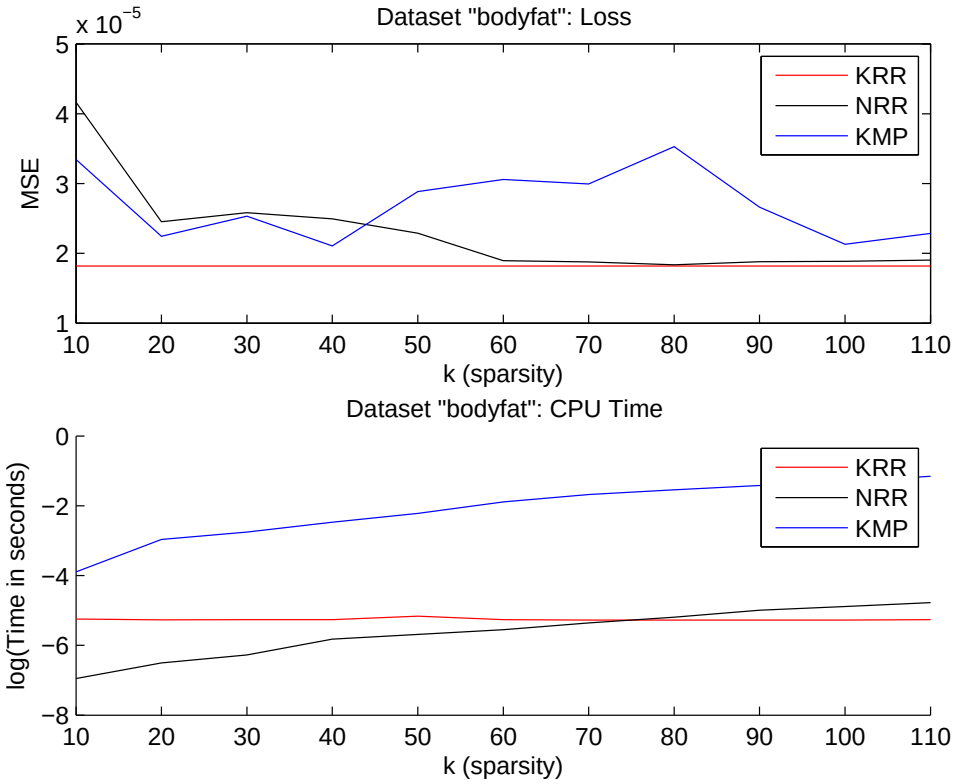


Figure 3.9: *Regression error (and log run-time) as a function of k for the Bodyfat dataset as achieved by KRR, NRR and KRR.*

3.5 Multi-View Learning

In the canonical form two or more "views" of the same data source are given, which are representations of the same underlying semantic object. Multi-View Learning (MVL) seeks to use information from both views in order to improve learning. Given two sets of signals which are in some way related, it would stand to reason that by making use of both signals the predictive power of the learned models can be improved.

Although often used interchangeably, it can be useful for both clarity of exposition and theoretical arguments to differentiate between Multi-Source Learning (MSL), MVL and Multiple Kernel Learning (MKL). The key differences are whether or not there are truly separate sources of information (MSL), or whether these are simply views of the same underlying semantic object (MVL), or whether different kernels are created given a single view of a data source (MKL). Whilst this might seem like splitting hairs, it can be an important distinction. Although in principle any algorithm developed for MVL can be used for MKL and *vice-versa*, the way in which data is amalgamated may be suboptimal. For example, a typical MKL will involve minimising over a convex set of kernels, but this assumes that the kernels are in the same family and is particularly sensitive to normalisation etc. MVL algorithms such as Kernel Canonical Correlation Analysis (KCCA), are designed to take advantage of correlations between views, but would perform poorly for standard MKL applications.

For example, MKL algorithms do not make any attempt to integrate the sources of information

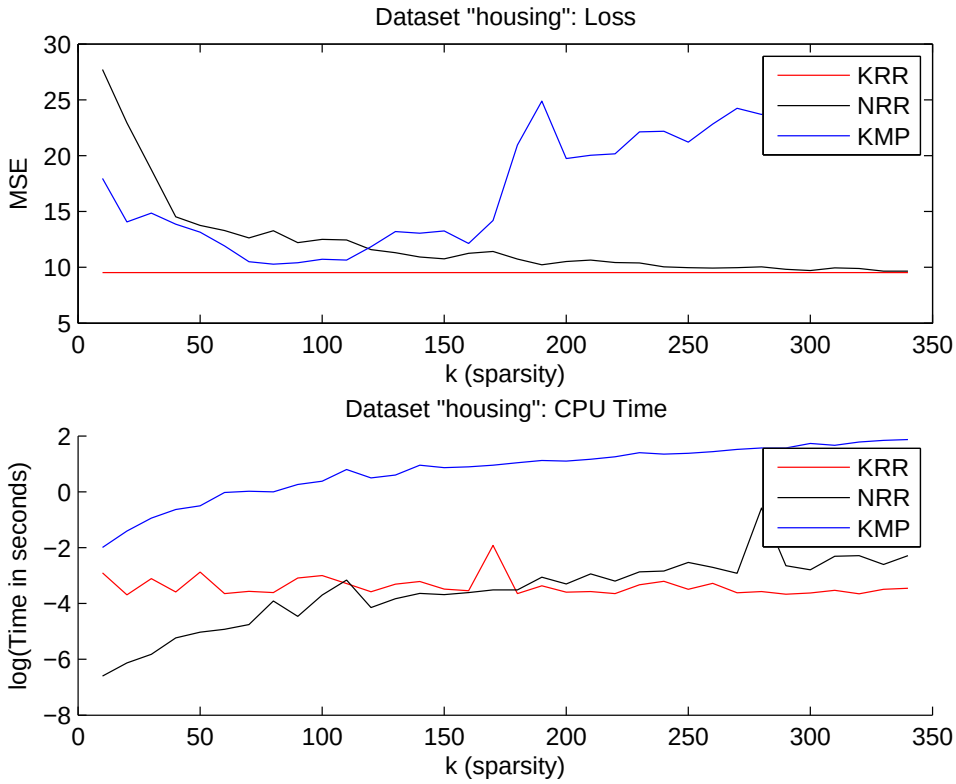


Figure 3.10: *Regression error (and log run-time) as a function of k for the Housing dataset as achieved by KRR, NRR and KRR.*

from each view, and work by simply placing weights over the kernels [105]. Anecdotally, it seems that in many practical situations in which the number of kernels is small, the performance of MKL algorithms can actually be worse than simply choosing the best kernel through a heuristic method such as CV[10]. In the MVL or MSL paradigm, we are assuming that the number of views or sources is typically small (*i.e.* $2 \rightarrow 10$), and hence another viewpoint is needed in which the sources are combined more usefully. The basic idea of MVL is to introduce one function per view which only uses the features from that view, and then jointly optimize these functions such that learning is enhanced. In MVL, we are also usually interested in having weight vectors and loadings for each of the views, which we do not have when we concatenate features (or equivalently sum kernel matrices), or take convex combinations of kernels as in the MKL setting.

The distinction between MSL and MVL is more subtle, and hence, most often confused. It is also, however, less important. Generally the distinction between single *versus* separate sources typically does not affect the modelling process. For the rest of this chapter, it will be assumed that the canonical paradigm is MVL, although the applications may be to both MVL and MSL. A diagrammatic view of this distinction is included in Figure 3.11.

Firstly KCCA is reintroduced, followed by an algorithmic development that allows it to be extended to the classification in an efficient way.

[10] Amongst others, this topic was discussed at the NIPS 2009 Workshop "Understanding Multiple Kernel Learning Methods"

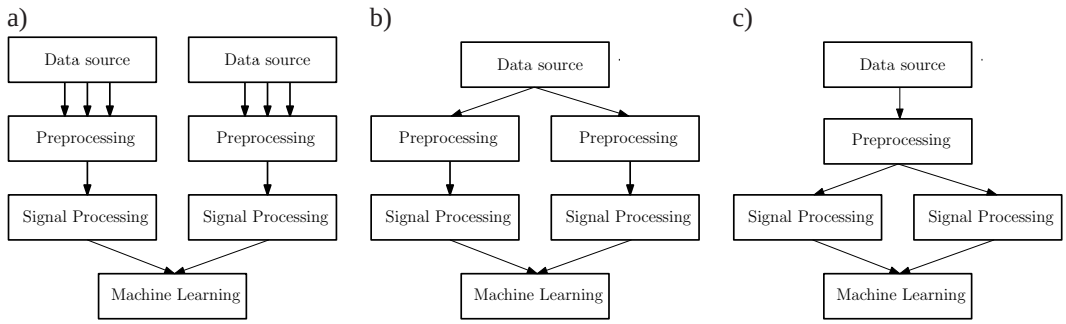


Figure 3.11: *Diagrammatic view of the process of a) Multi-Source Learning (MSL), b) Multi-View Learning (MVL) and c) Multiple Kernel Learning (MKL)*

Kernel Canonical Correlation Analysis (KCCA) was introduced in the previous Chapter in Section 2.1.13. KCCA finds basis vectors for two sets of variables such that the correlations between the projections onto these basis vectors are mutually maximised. The optimisation is given by,

$$\begin{aligned} \max_{\boldsymbol{\alpha}_a, \boldsymbol{\alpha}_b} \quad & \boldsymbol{\alpha}_a' \mathbf{K}_a \mathbf{K}_b \boldsymbol{\alpha}_b \\ \text{s.t.} \quad & \boldsymbol{\alpha}_a' \mathbf{K}_a^2 \boldsymbol{\alpha}_a = 1, \\ & \boldsymbol{\alpha}_b' \mathbf{K}_b^2 \boldsymbol{\alpha}_b = 1, \end{aligned} \tag{3.30}$$

where $\mathbf{K}_a$ and $\mathbf{K}_b$ are the kernel matrices of the two views.

3.5.1 Kernel Canonical Correlation Analysis with Projected Nearest Neighbours

In order to perform classification, typically the test data from one of the views (*e.g.* $\mathbf{K}_a$) is projected into the shared feature space (using $\boldsymbol{\alpha}_a$), and then a linear classification algorithm such as a primal SVM is then trained on this new feature space. However there is a way in which the projections can be used directly for classification, without incurring this additional computational cost. By using *e.g.* the 100 largest correlation values and the corresponding projections, the labels given by the corresponding example in the training set kernel from the other view are used as the classification. The reported errors are then the mean of the differences between these labels and the true test labels. This method is an extension of mate-based retrieval [106], and is given in Algorithm 5. It is non-parameteric and essentially *free* once the KCCA directions have been learnt. Because this algorithm is searching for the nearest neighbour in the shared semantic space defined by KCCA of the projection of test point into this space, we have called this algorithm Projected Nearest Neighbours (PNN).

A natural extension to this is to try to incorporate the classification and the subspace learning into a single optimisation routine. This was the motivation for MFDA and its variants [13], which are presented in the following Section, along with some experimental results on toy data and benchmark datasets.

Algorithm 5 Projected Nearest Neighbours (PNN) Classification

1: Given Kernels from each view $\mathbf{K}_a$ and $\mathbf{K}_b$, dual weight vectors $\boldsymbol{\alpha}_a$ and $\boldsymbol{\alpha}_b$ from KCCA, training labels $\mathbf{y}$, and vectors of train and test indices $\mathbf{i}$ and $\mathbf{j}$ respectively

2: Compute the projection of the training kernel of the first view

$$P_a \leftarrow \mathbf{K}_a[\mathbf{i}, \mathbf{i}]\boldsymbol{\alpha}_a$$

3: Compute the projection of the train-test kernel of the second view:

$$P_b \leftarrow \mathbf{K}_b[\mathbf{j}, \mathbf{i}]\boldsymbol{\alpha}_b$$

4: Compute the covariance matrix of the projections:

$$\boldsymbol{\Sigma}_{ab} \leftarrow P_a P_b'$$

5: Find the indices of the maximal values of each column:

$$\mathbf{k}[j] = \arg\max_{i \in \mathbf{i}} \left(\boldsymbol{\Sigma}_{ab}[i, j]\right) \quad \text{for } j \in \mathbf{j}$$

6: Select the labels of the training examples of those indices as the predictions:

$$\hat{\mathbf{y}} \leftarrow \mathbf{y}[\mathbf{k}]$$

3.5.2 Convex Multi-View Fisher Discriminant Analysis

As discussed in the previous Section, CCA and KCCA [52] attempt to integrate two sources of information by maximising the correlations between projections of each view. They are unsupervised techniques, and as such are not ideally suited to a classification setting. A common way of performing classification on two-view data using KCCA is to use the projected data from one of the views as input to a standard classification algorithm, such as a SVM, or to use the PNN method described above. However, the subspace that is learnt through such unsupervised methods may not always align well with the label space.

SVM-2K [107] was an attempt to take this to its logical conclusion by combining this two stage learning into a single optimisation. The algorithm introduces the constraint of similarity between two 1-dimensional projections which identify two distinct SVMs in the two feature spaces. However SVM-2K requires extra parameters (the C-parameter for each SVM, and another mixing parameter, along with any kernel parameters) that the methods presented here will not require. In addition, it is not easy to see how the SVM-2K formulation can be generalised to more than two views. There has been one related approach that tries to find the optimum combination of Fisher classifiers [108] using the MKL architecture [105]. In its initial form this problem is non-convex, although the authors do recast the problem in terms of a Semi-Definite Programme (SDP), at the expensive of an increase in the problem scale. In addition, the MKL architecture means that the output of the algorithm is a single weight vector for the convex combination of kernels. The formulation presented here has some similarities to that of [108], except cast here in the MVL framework and also providing additional modelling flexibility.

Here the convex formulation for FDA that was presented in the previous Chapter in Section 2.1.9 will be extended to multiple views. Given p "views" of the same data source, or alternatively p aligned

data sources, to form an $m-$sample S with input output $p+1$ tuples $(\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(p)}, y)$. It is assumed that each view has already been projected into a feature space $\mathcal{F}_d$, so that the kernel matrix $\mathbf{K}_d$ for that view has entries $\mathbf{K}_d[i,j] = \left\langle \mathbf{x}_{(d)i}, \mathbf{x}_{(d)j} \right\rangle$. Given matrices of inputs $\mathbf{X}_d = [\mathbf{x}_{(d)1}, \ldots, \mathbf{x}_{(d)m}]'$, the formulation (2.40) is extended to find p dual weight vectors $\boldsymbol{\alpha}_d, d = 1, \ldots, p$. The concatenation of these weight vectors will be denoted by $\tilde{\boldsymbol{\alpha}} = [\boldsymbol{\alpha}_1', \ldots, \boldsymbol{\alpha}_p']'$. The convex form of Multiview Fisher Discriminant Analysis (MFDA) is given in equation (3.31) below. The goal is now to minimise the variance of the data along the projection whilst maximising the distance between the average outputs for each class over all of the views.

$$\begin{aligned} \min_{\boldsymbol{\alpha}_d, b, \boldsymbol{\xi}} \quad & \mathcal{L}(\boldsymbol{\xi}) + \mu \mathcal{P}(\tilde{\boldsymbol{\alpha}}), \\ \text{s.t.} \quad & \sum_{d=1}^{p} (\mathbf{K}_d \boldsymbol{\alpha}_d + \mathbf{1} b_d) = \mathbf{y} + \boldsymbol{\xi}, \qquad\qquad d = 1, \ldots, p \\ & \boldsymbol{\xi}' \mathbf{e}^c = 0 \;\text{ for }\; c = 1, 2, \end{aligned} \tag{3.31}$$

where $\mathcal{L}(\cdot)$ is the loss function as before (2.41),

$$\mathcal{L}(\boldsymbol{\xi}) = \|\boldsymbol{\xi}\|_2^2,$$

and the regularisation function $\mathcal{P}(\cdot)$ is as follows,

$$\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \sum_{d=1}^{p} (\boldsymbol{\alpha}_d' \mathbf{K}_d \boldsymbol{\alpha}_d). \tag{3.32}$$

The first constraint in 3.31 ensures that the average loss between the output and its class label is minimised. The second constraint ensures that the average output for each class is each label. The classification function on a set of examples $\mathbf{x}_{(d),i}$ from views $d = 1, \ldots, p$ now becomes,

$$f(\mathbf{x}_{(d),i}) = \operatorname{sgn}\left(\sum_{d=1}^{p} f(\mathbf{x}_{(d)i}) \right) \tag{3.33}$$

$$= \operatorname{sgn}\left(\sum_{d=1}^{p} \mathbf{K}_d[:, i]' \boldsymbol{\alpha}_d + b_d \right). \tag{3.34}$$

Observe that the solutions given will be equivalent to summing kernels (as justified by the probabilistic interpretation). Meaning that viewed in the primal form, the result is the standard criterion in the space defined by the concatenation of the features, and the norm of the full weight vector is given by 3.32. However this formulation leads to two main advantages. Firstly, it provides a flexible framework that allows for different noise models and regularisation functions. Secondly, explicit weight vectors are available for each view, which allows the calculation of implicit weightings over the views (see Section 3.5.2 below).

Further intuition on the operation of the algorithm is as follows. Given two views $\mathbf{x}_{(a)}$ and $\mathbf{x}_{(b)}$, and using the standard ℓ_2 loss function, MFDA is trying to minimise the summed errors committed:

$\left\| f_a(\mathbf{x}_{(a)}) + f(\mathbf{x}_{(b)}) - \mathbf{y} \right\|_2^2$. So if some slack is added to one of the examples, *e.g.* $\mathbf{x}_{(a)i}$, then the algorithm will try to push the corresponding example $\mathbf{x}_{(b)i}$ the other way to try to minimise the overall slack. This can be seen as "view disagreement" which means that the algorithm tries to use information from both views to aid the classification. However of course the algorithm can "give up" and allow the slack to be big for that example, meaning that $\mathbf{x}_{(a)}$ and $\mathbf{x}_{(b)}$ can be pushed the same way.

It is actually possible to state the problem as the reverse - saying that normally in MVL the goal is to search for view agreement, which would be minimising $\left\| f(\mathbf{x}_{(a)}) - f(\mathbf{x}_{(b)}) \right\|_2^2$ (ignoring the labels). This is one particular form of the so-called "Co-Training" problem, which in order to work requires that each of the views are *sufficient* for classification, and methods that use this break down when there is significant view disagreement. A recent paper tried to get around this by learning separate classifiers and then looking for view agreement/disagreement between them, before combining them into a final classifier (a form of bootstrapping)[109]. MFDA should have an advantage over this as it is directly optimising the combined classifier. However, the alternative 'Private' method Private Multiview Fisher Discriminant Analysis (PMFDA) has separate slacks for each view as well as the overall slacks (see Section 3.5.2 to follow). This should allow the problem to flip around in some cases. Basically, if there is a "trouble" point in view $\mathbf{x}_{(a)}$, but not in view $\mathbf{x}_{(b)}$, the disagreement can be soaked up by the private slack, allowing the two views to move into agreement with zero shared slack.

Probabilistic Interpretation

Following the analysis of [35], it is possible to view the KFDA algorithm from a probabilistic point of view. It is known that FDA is Bayes optimal for two Gaussian distributions with equal covariance in the input space. The data may not fall naturally into this model, but it may be the case that for certain feature spaces (*e.g.* the space defined by the RBF kernel), the examples projected into a manifold in this space may be well approximated by Gaussian distributions with diagonal covariance. In this case KFDA would be Bayes optimal in the feature space.

If one considers KFDA as regression on to the labels, then a Gaussian noise model (as defined in Section 2.1.4) with known variance σ would result in the following expression for the likelihood $\Pr(\mathbf{y}|\boldsymbol{\alpha}) = \exp(-\left\|\boldsymbol{\xi}\right\|_2^2)$. If a prior over the weights with hyperparameters μ is used, the log of the posterior is simply $\log(\Pr(\mathbf{y}|\boldsymbol{\alpha})\Pr(\boldsymbol{\alpha}|\mu)) = -\left\|\boldsymbol{\xi}\right\|_2^2 - \log(\Pr(\boldsymbol{\alpha}|\mu))$. The choice of prior then becomes equivalent to the choice of regularisation function, which will be discussed in Section 3.5.2. When viewed in this way the outputs produced by KFDA can be interpreted as probabilities, which in turn makes it possible to assign confidence to the final classifications.

This view of KFDA also motivates the Multiview extension of the algorithm. We can extend and combine the graphical interpretations of [110] and [111] using the above definitions as seen in Figure 3.12. Note that explicit mixing weights $\boldsymbol{\beta}$ paramaterised by ρ are shown (dotted). Note that due to the optimisation (which constrains the functions over each feature space with the shared slack variable) and the fact that we have separate $\boldsymbol{\alpha}$ vectors for each view, we are able to drop the mixing weights $\boldsymbol{\beta}$ from our formulation. Under the assumption that the kernels are normalised, we can calculate these weigths

post-hoc as will be shown in Section 3.5.2. Taking the approach of Naïve Bayes Probabilistic Label

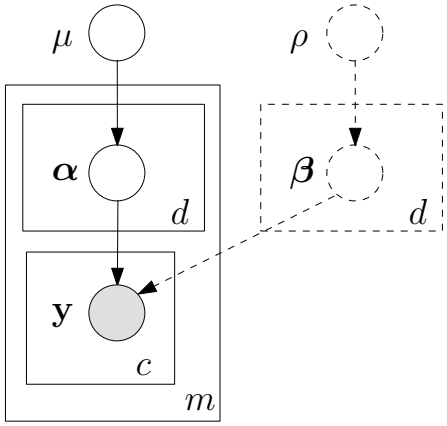


Figure 3.12: *Plates diagram showing the hierarchical Bayesian interpretation of MFDA. $\boldsymbol{\beta}$ are the hypothetical mixing parameters with prior weights ρ if an explicit mixing was used - in the case of MFDA these are fixed and hence can be removed, but can be calculated post-hoc.*

Fusion (NBF) [112], the first step is to assume conditional independence between classifiers given a class label. Suppose the set of labels $\mathbf{s} = \{s_1, \ldots, s_p\}$ are given from p classifiers for a given point $\mathbf{x}_i$. Denoting $\Pr(s_d)$ as the probability that classifier D_d labels an example $\mathbf{x}_i$ in class $\omega_c \in \Omega$, (in this case $\Omega = \{-1, +1\}$), then the likelihood of the classifiers given a label is,

$$\begin{aligned} \Pr(\mathbf{s}|\omega_c) &= \Pr(s_1, \ldots, s_p|\omega_c) \\ &= \prod_{d=1}^{p} \Pr(s_d|\omega_c). \end{aligned} \tag{3.35}$$

The posterior probability needed to label $\mathbf{x}_i$ is then given by,

$$\begin{aligned} \Pr(\omega_c|\mathbf{s}) &= \frac{\Pr(\omega_c)\Pr(\mathbf{s}|\omega_c)}{\Pr(\mathbf{s})} \\ &= \frac{1}{Z}\Pr(\omega_c) \prod_{d=1}^{p} \Pr(s_d|\omega_c), \end{aligned} \tag{3.36}$$

where Z is a normalisation constant. Assume a uniform prior over labels, the log posterior is then given by,

$$\log(\Pr(\omega_c|\mathbf{s})) \propto \sum_{d=1}^{p} \log(\Pr(s_d|\omega_c)). \tag{3.37}$$

This implies that by directly optimising this sum, we are optimising the NBF over KFDA classifiers, which is precisely the motivation for both the objective function and the classification function for MFDA, both of which will be described in the next Section. At first glance it seems that this conditional independence assumption could be problematic, as this assumption is seldom true. However, Kuncheva made the point that despite this NBF is experimentally observed to be surprisingly accurate and efficient [112]. However, it does open the door to further possibilities for combining KFDA classifiers, but this is

outside the scope of the present work.

Implicit Weighting

In order to determine the importance of each of the views after training, following [113] it is possible to calculate the implicit weighting of each view simply through the weighted sum of the absolute values of the classification functions. This is justified by the intuition made in Section 3.5.2 that the outputs of each classifier can be interpreted as probabilities, with the assumption that each kernel is normalised as per [3], *i.e.* $\mathrm{trace}(\mathbf{K}_d) = m$. This in turn means that the overall confidence of the classifier can be calculated as the sum of the log probabilities that the function $f(\mathbf{x}_{(d)i})$ for classifier d on example i give the class label ω_c.

$$\begin{aligned} u_d &\approx \frac{1}{Z}\sum_{c\in\Omega}\log(p(s_d|\omega_c)) \\ &= \frac{\sum_{i=1}^{m}|\mathbf{K}_d[:,i]'\boldsymbol{\alpha}_d + b_d|}{\sum_{i=1}^{m}\sum_{d=1}^{p}|\mathbf{K}_d[:,i]'\boldsymbol{\alpha}_d + b_d|}. \end{aligned} \tag{3.38}$$

Regularisation and Loss Functions

The natural choices for the regularisation function $\mathcal{P}(\tilde{\boldsymbol{\alpha}})$ would either be the sum of the ℓ_2-norms of the primal weight vectors (as in (3.32)), or the sum of the ℓ_2-norms of the dual weight vector $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \sum_{d=1}^{p}\|\boldsymbol{\alpha}_d\|_2^2$. However more interesting is the ℓ_1-norm of the dual weight vector, $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \sum_{d=1}^{p}\|\boldsymbol{\alpha}_d\|_1$, as this choice leads to sparse solutions (as previously discussed) due to the fact that the ℓ_1-norm can be seen as an approximation to the (pseudo) ℓ_0-norm. In the rest of the chapter the ℓ_1-norm regularisation method is denoted as Sparse Multiview Fisher Discriminant Analysis (SMFDA).

In some situations these regularisation functions $\mathcal{P}(\cdot)$ may be too simplistic, in which case additional domain knowledge can be incorporated into the function. For example, there is reason to believe *a-priori* that most of the views are likely not to be useful, but the individual weights in that view are, then $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \|\mathbf{A}\|_{2,1}$ could be used where $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p]$ is $\tilde{\boldsymbol{\alpha}}$ reshaped as a matrix of weights and the block (r,p)-norm of $\mathbf{A}$ is defined as $\|\mathbf{A}\|_{r,p} = (\sum_{i=1}^{m}\|\boldsymbol{\alpha}_i\|_p^r)^{1/p}$. Another example would be a situation it may be desirable to impose sparsity on some views but not others. For two views, this would simply be $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \|\boldsymbol{\alpha}_1\|_2^2 + \|\boldsymbol{\alpha}_2\|_1$ in order to promote sparsity in the second view but not the first. One could also promote sparsity in the primal version of one view by passing in the explicit features for that view (if available) and penalising $\mathbf{X}_d'\boldsymbol{\alpha}_d$. In this way any mixture of linear with nonlinear features and primal with dual sparsity can be combined across the views, all in a single optimisation framework. One can also pre-specify the weights of views by parameterising them, if one has a strong prior belief that a view will be more or less useful, but it in general it is not necessary or helpful to do this.

Following [114] the assumption of a Gaussian noise model can also be removed, resulting in different loss functions on the slacks $\boldsymbol{\xi}$. For example, if a Laplacian noise model is chosen $\|\boldsymbol{\xi}\|_2^2$ can be replaced with $\|\boldsymbol{\xi}\|_1$ in the objective function. The advantage of this is if the ℓ_1-norm regulariser from above is chosen, the resulting optimisation is a linear programme, which can be solved efficiently using

methods such as column generation. From a modelling perspective, it may be advantageous to choose a noise model that is robust to outliers, such as Huber's Robust loss, which can easily be used in the framework presented here[11].

Incorporating Private Directions

The above formulations seek to find the projection that is maximally discriminative averaged across views. However these problems are very tightly constrained, and optimisation may be difficult in situations where one or more of the views is not informative of the labels (*i.e.* is essentially noise). This leads to considering the allowance of some extra slack $\boldsymbol{\zeta}_d$ that is private to each view, which is similar in vein to the approach taken by [83] to Multi-Task learning (MTL) and [115] to probabilistic latent space modelling. This leads to the following formulation which we term PMFDA,

$$\begin{aligned} \min_{\boldsymbol{\alpha}_d, b, \boldsymbol{\xi}, \boldsymbol{\zeta}_d} \quad & H(\boldsymbol{\xi}, \tilde{\boldsymbol{\zeta}}, \tau) + \mu \mathcal{P}(\boldsymbol{\alpha}_d), && d = 1, \ldots, p \\ \text{s.t.} \quad & \mathbf{K}_d \boldsymbol{\alpha}_d + \mathbf{1} b = \mathbf{y} + \boldsymbol{\xi} + \boldsymbol{\zeta}_d && d = 1, \ldots, p \\ & \mathbf{1}_i' \boldsymbol{\xi} = 0 && i = 1, 2, \end{aligned} \tag{3.39}$$

with $\tilde{\boldsymbol{\zeta}} = [\boldsymbol{\zeta}_1', \ldots, \boldsymbol{\zeta}_p']'$. The regularisation function $\mathcal{P}(\cdot)$ is as before (3.32), and the loss function is updated to incorporate $\boldsymbol{\zeta}_d$ as follows,

$$H(\boldsymbol{\xi}, \tilde{\boldsymbol{\zeta}}, \tau) = \|\boldsymbol{\xi}\|_2^2 + \tau \sum_{d=1}^{p} \|\boldsymbol{\zeta}_d\|_2^2. \tag{3.40}$$

Note the extra parameter τ which enables the tuning of the relative importance of private or shared slacks. If $\tau = 1$ the penalties of the private slack for an example i are proportional to $\boldsymbol{\xi}_i / p$, which means that the more views that are added, the less each view is allowed to dominate. In the experiments conducted here this was simply set heuristically to 0.1 to allow a reasonable amount of leeway for each view.

Generalisation Error Bound for MFDA

We now construct a generalisation error bound for MFDA by applying the following results from [85] and [86] and extending to the Multiview case. The first bounds the difference between the empirical and true means (Theorem 3 in [85]).

Theorem 3.5.1 (Bound on the true and empirical means)**.** *Let S_d be a view of a sample of m points drawn independently according to a probability distribution P_d. Consider the mean vector $\boldsymbol{\mu}_d$ and the empirical estimate $\hat{\boldsymbol{\mu}}_d$ defined as*

$$\begin{aligned} \boldsymbol{\mu}_d &= \mathbb{E}_{P_d} \left[\phi(\mathbf{x}_d) \right], \\ \hat{\boldsymbol{\mu}}_d &= \hat{\mathbb{E}}_{\mathbf{x}_d} \left[\phi(\mathbf{x}_d) \right] = \frac{1}{p} \sum_{d=1}^{p} \phi(\mathbf{x}_d). \end{aligned} \tag{3.41}$$

[11] See Section 2.1.9 in the previous Chapter for an outline of some loss functions for classification

Then with probability at least $1-\delta$ *over the choice of* S_d*, we have*

$$\|\hat{\boldsymbol{\mu}}_d - \mathbb{E}_{\mathbf{x}_d}[\phi(\mathbf{x}_d)]\| \leq \frac{R}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{1}{\delta}}\right). \tag{3.42}$$

Consider the covariance matrix $\boldsymbol{\Sigma}_d$ and the empirical estimate $\hat{\boldsymbol{\Sigma}}_d$ defined as

$$\begin{aligned}\boldsymbol{\Sigma}_d &= \mathbb{E}\left[(\phi(\mathbf{x}_d)-\boldsymbol{\mu}_d)(\phi(\mathbf{x}_d)-\boldsymbol{\mu}_d)'\right],\\ \hat{\boldsymbol{\Sigma}}_d &= \hat{\mathbb{E}}\left[(\phi(\mathbf{x}_d)-\hat{\boldsymbol{\mu}}_d)(\phi(\mathbf{x}_d)-\hat{\boldsymbol{\mu}}_d)'\right].\end{aligned} \tag{3.43}$$

The following corollary bounds the difference between the empirical and true covariance (Corollary 6 in [85]).

Corollary 3.5.2 (Bound on the true and empirical covariances)**.** *Let* S_d *be an* m *sample from* P_d *as above, where* R_d *is the radius of the ball in the feature space* $\mathcal{F}_d$ *containing the support of the distribution. Provided* $m \geq (2+\sqrt{2\ln 2/\delta})^2$*, we have*

$$\left\|\hat{\boldsymbol{\Sigma}}_d - \boldsymbol{\Sigma}_d\right\|_F \leq \frac{2R_d^2}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{2}{\delta}}\right), \tag{3.44}$$

The following Lemma is connected with the classification algorithm "Robust Minimax Classification" developed by [86], adapted here for MFDA.

Lemma 3.5.3. *Let* $\boldsymbol{\mu}_d$ *be the mean of a distribution and* $\boldsymbol{\Sigma}_d$ *its covariance matrix,* $\mathbf{w}_d \neq 0$*,* b *given, such that* $\mathbf{w}_d'\boldsymbol{\mu}_d + b \leq 0$ *and* $\Delta \in [0,1)$*, then if*

$$-(\mathbf{w}_d'\boldsymbol{\mu}_d + b) \geq \kappa(\Delta)\sqrt{\mathbf{w}_d'\boldsymbol{\Sigma}_d\mathbf{w}_d},$$

where $\kappa(\Delta) = \sqrt{\frac{\Delta}{1-\Delta}}$*, then*

$$\Pr\left(\mathbf{w}_d'\phi(\mathbf{x}_d) + b \leq 0\right) \geq \Delta$$

In order to provide a true error bound we must bound the difference between this estimate and the value that would have been obtained had the true mean and covariance been used.

Theorem 3.5.4 (Main)**.** *Let* S_d *be a view of a sample of* m *points drawn from* P_d *as above, where* R_d *is the radius of the ball in the feature space* $\mathcal{F}_d$ *containing the support of the distribution. Let* $\hat{\boldsymbol{\mu}}_d$ *(*$\boldsymbol{\mu}_d$*) be the empirical (true) mean of a sample of* m *points from the view* S_d*,* $\hat{\boldsymbol{\Sigma}}_d$ *(*$\boldsymbol{\Sigma}_d$*) its empirical (true) covariance matrix,* $\mathbf{w}_d \neq \mathbf{0}$*,* $\|\mathbf{w}\|_2 = 1$*, and* b *given, such that* $\mathbf{w}_d'\boldsymbol{\mu}_d + b \leq 0$ *and* $\Delta \in [0,1)$*. Then with probability* $1-\delta$ *over the draw of the random sample, if*

$$-(\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d + b) \geq \kappa(\Delta)\sqrt{\mathbf{w}_d'\hat{\boldsymbol{\Sigma}}_d\mathbf{w}_d} \qquad d = 1,\ldots,p,$$

then

$$\Pr\left((\mathbf{w}_d'\phi_d(\mathbf{x}_d)+b)>0\right)<1-\Delta,$$

where

$$\Delta=\frac{\left(\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d+b-A_d\right)^2}{\mathbf{w}_d'\hat{\boldsymbol{\Sigma}}_d\mathbf{w}_d+B_d+\left(\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d+b-A_d\right)^2},$$

such that $\|\hat{\boldsymbol{\mu}}_d-\boldsymbol{\mu}_d\|\leq A_d$ *where* $A_d=\frac{R_d}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{2m}{\delta}}\right)$,
and $\left\|\hat{\boldsymbol{\Sigma}}_d-\boldsymbol{\Sigma}_d\right\|_F\leq B_d$ *where* $B_d=\frac{2R_d^2}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{4m}{\delta}}\right)$.

Proof. (sketch). First we re-arrange $\mathbf{w}_d'\boldsymbol{\mu}_d+b\geq\kappa(\Delta)\sqrt{\mathbf{w}_d'\boldsymbol{\Sigma}_d\mathbf{w}_d}$ from Lemma 3.5.3 for each view in terms of $\kappa(\Delta)$:

$$\kappa(\Delta)=\frac{\mathbf{w}_d'\boldsymbol{\mu}_d+b}{\sqrt{\mathbf{w}_d'\boldsymbol{\Sigma}\mathbf{w}_d}}. \tag{3.45}$$

These quantities are in terms of the true means and covariances. In order to achieve an upper bound we need the following sample compressed results for the true and empirical means (Theorem 3.5.1) and covariances (Corollary 3.5.2):

$$\|\hat{\boldsymbol{\mu}}_d-\mathbb{E}_{\mathbf{x}_d}[\hat{\mu}_d(\mathbf{x}_d)]\|\leq A_d=\frac{R_d}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{2m}{\delta}}\right),$$
$$\left\|\hat{\boldsymbol{\Sigma}}_d-\boldsymbol{\Sigma}_d\right\|_F\leq B_d=\frac{2R_d^2}{\sqrt{m}}\left(2+\sqrt{2\ln\frac{4m}{\delta}}\right).$$

Given equation (3.45) we can use the empirical quantities for the means and covariances in place of the true quantities. However, in order to derive a genuine upper bound we also need to take into account the upper bounds between the empirical and true means. Including these in the expression above for $\kappa(\Delta)$ by replacing δ with $\delta/2$, to derive a lower bound, we get:

$$\kappa(\Delta)=\frac{\mathbf{w}_d'\hat{\boldsymbol{\mu}}_{dS_d}+b-A_d}{\sqrt{\mathbf{w}_d'\hat{\boldsymbol{\Sigma}}_d\mathbf{w}_d+B_d}}.$$

Finally, making the substitution $\kappa(\Delta)=\sqrt{\frac{\Delta}{1-\Delta}}$ and solving for Δ yields the result. □ □

The following Proposition upper bounds the generalisation error of Multiview Fisher Discriminant Analysis (MFDA).

Proposition 3.5.5. *Let* $\mathbf{w}_d$, b, *be the (normalised) weight vector and associated threshold returned by the Multiview Fisher Discriminant Analysis (MFDA) when presented with a view of the training set* S_d. *Furthermore, let* $\hat{\boldsymbol{\Sigma}}_d^+$ *(*$\hat{\boldsymbol{\Sigma}}_d^-$*) be the empirical covariance matrices associated with the positive (negative) examples of the* m *training samples from* S_d *projected using* $\mathbf{w}_d$. *Then with probability at least* $1-\delta$ *over the draw of all the views of the random training set* $S_d,\ d=1,\ldots,p$ *of* m *training examples, the*

generalisation error $\mathcal{R}$ is bounded by

$$\mathcal{R} \leq \max(1 - \Delta^+, 1 - \Delta^-)$$

where Δ^j, $j = +, -$ such that

$$\Delta^j = \frac{j\left(\left(\sum_{d=1}^{p}(\mathbf{w}_d'\hat{\boldsymbol{\mu}}_{S_d}^j + b) - C^j\right)^2\right)}{\left(\sum_{d=1}^{p}\mathbf{w}_d'\hat{\boldsymbol{\Sigma}}_d^j\mathbf{w}_d\right) + D^j + \left(j(\sum_{d=1}^{p}\mathbf{w}_d'\hat{\boldsymbol{\mu}}_{S_d}^j + b) - C^j\right)^2},$$

where $C^j = \frac{\sum_{d=1}^{p} R_d}{\sqrt{m^j}}\left(2 + \sqrt{2\ln\frac{4mp}{\delta}}\right)$, $\quad D^j = \frac{2\sum_{d=1}^{p} R_d^2}{\sqrt{m^j}}\left(2 + \sqrt{2\ln\frac{8mp}{\delta}}\right)$.

Proof. For the negative part of the proof we require $\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d^- + b \geq \kappa(\Delta)\sqrt{\mathbf{w}_d'\hat{\boldsymbol{\Sigma}}_d^-\mathbf{w}_d}$ which is a straight forward application of Theorem 3.5.4 with δ replaced with $\delta/2$. For the positive part, observe that we require $\mathbf{w}_d'\hat{\boldsymbol{\mu}}_d^+ - b \geq \kappa(\Delta)\sqrt{\mathbf{w}_d'\hat{\boldsymbol{\Sigma}}_d^+\mathbf{w}_d}$, hence, a further application of Theorem 3.5.4 with δ replaced by $\delta/2$ suffices. Finally, we take a union bound over the p views such that m is replaced by mp. □ □

Experiments: Toy Data

In order to validate the outlined methods, experiments were first conducted with simulated toy data. A data source S was created by taking two $1-$dimensional Gaussian distributions (S^+, S^-) which were well separated, which was then split into 100 train and 50 test points. The source S was embedded into $2-$dimensional views through complementary linear projections (ϕ_1, ϕ_2) to give new "views" $\mathbf{X}_1, \mathbf{X}_2$. Differing levels of independent "measurement noise" were added to each view (n_1, n_2), and identical "system noise" was added to both views (n_S). A third view was constructed of pure noise to simulate a faulty sensor ($\mathbf{X}_3$). The labels $\mathbf{y}$ were calculated as the sign of the original data source.

$S = \{S^+, S^-\}$	(source)
$S^+ \sim \mathcal{N}(5,1), S^- \sim \mathcal{N}(-5,1)$	
$\mathbf{y} = \text{sgn}(S)$	(labels)
$\phi_1 = [1,-1], \phi_2 = [-1,1]$	(projections)
$n_1 \sim \mathcal{N}(0,5)^2, n_2 \sim \mathcal{N}(0,3)^2$	(meas. noise)
$n_S \sim \mathcal{N}(0,2)^2$	(system noise)
$\mathbf{X}_1 = \phi_1' S + n_1 + n_S$	(view 1)
$\mathbf{X}_2 = \phi_2' S + n_2 + n_S$	(view 2)
$\mathbf{X}_3 = n_S$	(view 3)

$\mathbf{X}_1$ and $\mathbf{X}_2$ are noisy views of the same signal, with correlated noise, which can be a typical problem in multivariate signal processing (*e.g.* sensors in close proximity). Linear kernels were used for each view. A small value for the regularisation parameter $\mu = 10^{-3}$ was chosen heuristically for all the experiments. Table 3.8 gives an overview of the results on the toy dataset. Comparisons were made against:

KFDA on each of the views (denoted as $f(1)$, $f(2)$ and $f(3)$ respectively);

summing the classification functions of these ($fsum$);

summing the kernels of each view ($ksum$);

followed by MFDA, PMFDA and SMFDA.

Note that an unweighted sum of kernels is equivalent to concatenating the features before creating a single kernel. The table shows the test error over 10 random repeats of the experiment in first column, followed by the implicit weightings for each of the algorithms calculated via (3.38). Note that the $ksum$ method returns single $m-$dimensional weight vector, and unless a kernel with an explicit feature space is used it is not possible to recalculate the implicit weightings over the features. In this case, since linear kernels are used the weightings have been calculated. For the three methods outlined in this paper (MFDA, PMFDA, SMFDA), as expected the performance is roughly equivalent to the $ksum$ method. The last row in the table (actual) is the empirical Signal to Noise Ratio (SNR) calculated as $SNR_d = \sum(\mathbf{X}_d'\mathbf{X}_d)/\mathrm{var}(S - \mathbf{X}_d)$ for view d, which as can be seen is closely matched by the weightings given.

The sparsity of SMFDA can be seen in figure 3.13. The sparsity level quoted in the figure is the proportion of the weights below 10^{-5}.

Method	**Test error**	$W(1)$	$W(2)$	$W(3)$
$f(a)$	0.19	1.00	0.00	0.00
$f(b)$	0.10	0.00	1.00	0.00
$f(c)$	0.49	0.00	0.00	1.00
$fsum$	0.39	0.33	0.33	0.33
$ksum$	0.04	0.29	0.66	0.05
MFDA	0.04	0.29	0.66	0.05
PMFDA	0.04	0.29	0.66	0.05
SMFDA	0.04	0.29	0.66	0.05
Actual		0.35	0.65	0.00

Table 3.8: *Test errors over ten runs on the toy dataset. Methods described in the text. $W(\cdot)$ refers to the implicit weightings given by each algorithm for each of the views. Note that the weightings closely match the actual SNR.*

Experiments: VOC 2007 DATASET

The sets of features ("views") used can be found in [116], with an extra feature extraction method known as Scale Invariant Feature Transformation (SIFT) [117]. RBF kernels were constructed for each of these feature sets, the RBF width parameter was set using a heuristic method [12]. The Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) Visual Object Classes (VOC) 2007 challenge database was used which contains 9963 images, each with at least 1 object. The number of objects in each image ranges from 1 to 20, with, for instance, objects of people, sheep, horses, cats, dogs etc. For a complete list of the objects, and description of the data set see the VOC 2007 challenge website[13].

Figure 3.14 shows Recall-Precision curves for SMFDA with 1, 2, 3 or 11 kernels and PicSOM

[12] For each setting of the width parameter, histograms of the kernel values were created. The chosen kernel was the one whose histogram peak was closest to 0.5 (*i.e.* furthest from 0 and 1).

[13] http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

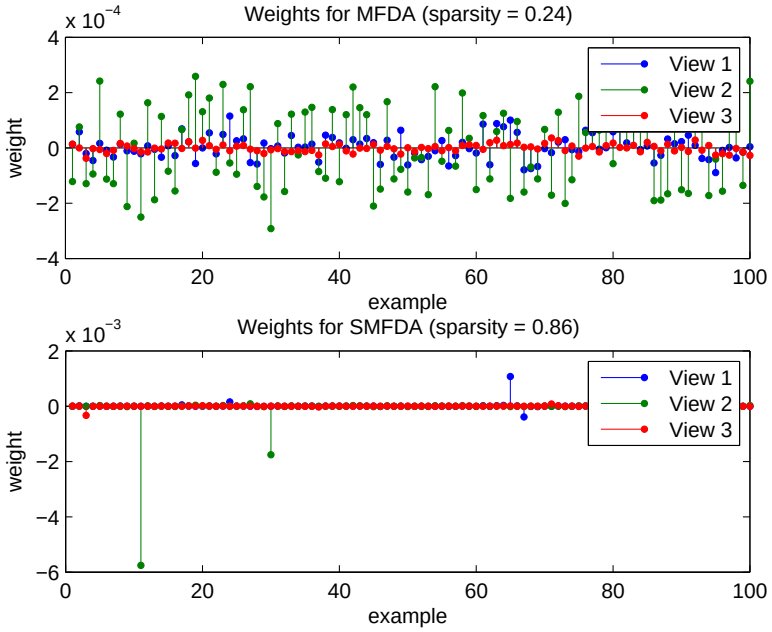


Figure 3.13: *Weights given by MFDA and SMFDA on the toy dataset. Notice that many of the weights for SMFDA are close to zero, indicating sparse solutions. Also notice that most of the weights for view 3 (pure noise) are close to zero.*

[116], and Table 3.9 shows the balanced error rate (the average of the errors on each class) and overall average precision for the PicSOM, KFDA using cross-validation to choose the best single kernel, KFDA using an unweighted sum of kernels, and MFDA. For the purposes of training, a random subset of 200 irrelevant images was used rather than the full training set. Results for three of the object classes (cat, cow, dog) are presented. The results show that, in general, adding more kernels into the optimisation can assist in recall performance. For each object class, the subsets of kernels (*i.e.* 1,2, or 3) were chosen by the weights given by SMFDA on the 11 kernels. The best single kernel (based on SIFT features) performs well alone, yet the improvement in some cases is quite marked. Results are competitive with the PicSOM algorithm, which uses all 11 feature extraction methods, and all of the irrelevant images.

Dataset →	**Cat**		**Cow**		**Horse**	
Method ↓	BER	AP	BER	AP	BER	AP
PicSOM	n/a	0.18	n/a	0.12	n/a	0.48
KFDA CV	0.26	0.36	0.32	0.14	0.22	0.51
MFDA	0.26	0.36	**0.27**	**0.15**	**0.19**	0.58

Table 3.9: *Balanced Error Rate (BER) and Average Precision (AP) for four of the VOC challenge datasets, for four different methods: PicSOM, KFDA with cross validation (KFDA CV), KFDA using a sum of kernels ($ksum$) and MFDA*

Experiments: Neuroimaging Dataset

This section describes analysis of fMRI data[14] that was acquired from 16 right handed healthy US college male students aged 20-25 which, according to a self report, did not have any history of neurological or

[14] Data kindly donated by Mourão-Miranda *et. al.* [118].

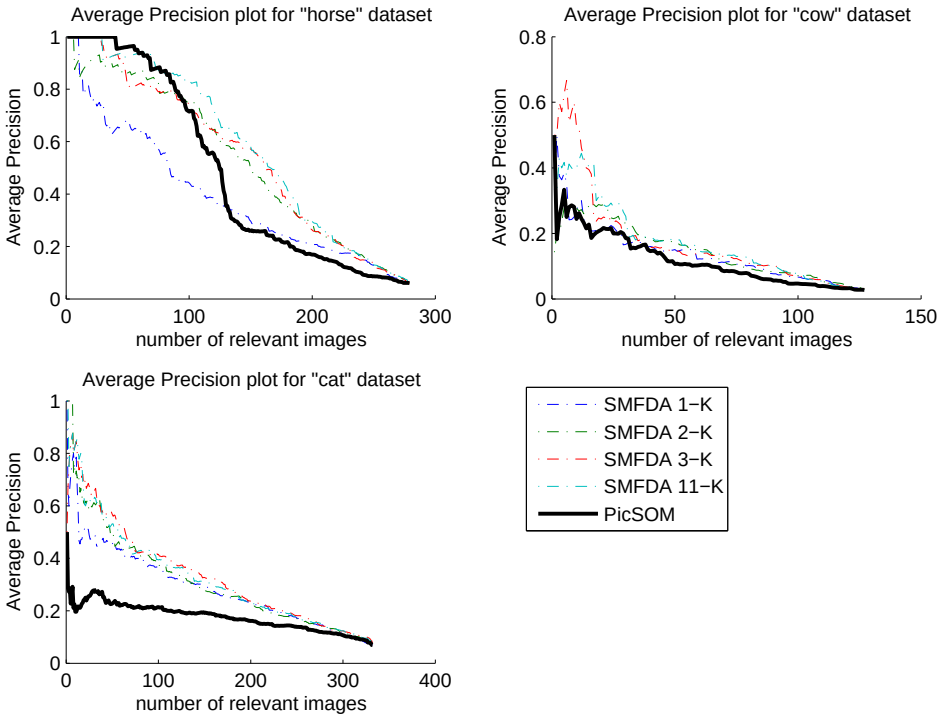


Figure 3.14: *Average precision recall curves for 3 VOC 2007 datasets for SMFDA plotted against PicSOM results*

psychiatric illness. The subjects viewed image stimuli of three different active conditions: viewing unpleasant (dermatologic diseases), neutral (people), pleasant images (female models in swimsuits and lingerie), and a control condition (fixation). In these experiments only unpleasant and pleasant image categories are used. The image-stimuli were presented in a block fashion and consisted of 42 images per category. During the experiment, there were 6 blocks of each active condition (each consisting of 7 image volumes) alternating with control blocks (fixation) of 7 images volumes.

In a similar fashion to the study in [53], pleasant images are given positive labels and unpleasant negative labels, the image stimuli are represented using SIFT features [117]. Conventional pre-processing was applied to the fMRI data. A detailed description of the fMRI pre-processing procedure and image-stimuli representation is given in [53]. The experiments were run in a leave-subject-out fashion where 15 subjects are combined for training and a single subject is withheld for testing. This gave a sum total of $42 \times 2 \times 15 = 1260$ training and $42 \times 2 = 84$ testing fMRI volumes and paired image stimuli. The analysis was repeated for each participant (hence 16 times) using linear kernels. In the following experiment, the following comparisons were made:

- An SVM on the fMRI data (single view)
- KCCA on the fMRI + Image Stimuli (two views) followed with an SVM trained on the fMRI data projected into the learnt KCCA semantic space
- SMFDA on the fMRI + Image Stimuli (two views)

The results are given in Table 3.10 where it can be observed that on average MFDA performs better than both the SVM (which is a single view approach), and the KCCA/SVM which similarly to MFDA

Sub.	SVM	KCCA/SVM	MFDA
1	0.1310	0.1667	**0.1071**
2	0.1905	0.2739	**0.1429**
3	0.2024	**0.1786**	0.1905
4	0.1667	0.2125	**0.1548**
5	**0.1905**	0.2977	0.2024
6	0.1667	0.1548	**0.1429**
7	**0.1786**	0.2262	0.1905
8	0.2381	0.2858	**0.2143**
9	0.3096	0.3334	**0.2619**
10	0.2977	0.3096	**0.2262**
11	**0.1191**	0.1786	0.1429
12	0.1786	0.2262	**0.1667**
13	0.2500	0.2381	**0.0714**
14	0.4405	0.4405	**0.2619**
15	**0.2500**	0.2977	0.2738
16	**0.1429**	0.1905	0.1860
Mean:	0.2158±0.08	0.2508±0.08	**0.1860±0.06**

Table 3.10: *In the table above the leave-one-out errors for each subject are presented. The following methods are compared: SVM on the fMRI data alone; KCCA analysis on the two views fMRI and Image Stimuli followed by an SVM on the projected fMRI data; the proposed MFDA on the two views fMRI+Image. Numbers in bold indicate the best performing algorithm for a particular subject.*

incorporates two views into the learning process. In this case the label space is clearly not well aligned with the KCCA projections, whereas a supervised method such as MFDA is able to find this alignment

3.6 Conclusions and Further Work

This goal of this Chapter was to present a unified general framework for the application of sparse ML methods to multivariate signal processing. The methods presented can be seen as modular building blocks that can be applied to a variety of applications. To begin with, the focus was on greedy methods for sparse classification and regression, specifically Matching Pursuit Kernel Fisher Discriminant Analysis (MPKFDA) and Kernel Polytope Faces Pursuit (KPFP). This was followed by a presentation of methods that take advantage of the Nyström method for low-rank kernel approximation for large scale data, including Nyström KRR (NRR), Nyström KFDA (NFDA), and Nyström SVM (NSVM). For the rest of the Chapter the attention was turned to the problem of learning from multiple data sources or views (MSL and MVL respectively), with the development of Multiview Fisher Discriminant Analysis (MFDA), Sparse Multiview Fisher Discriminant Analysis (SMFDA) and Private Multiview Fisher Discriminant Analysis (PMFDA). Detailed conclusions for each of the methods presented can be found in 6.

Chapter 4

Applications I

Abstract

Styles of Music. *The first application area for the "LeStruM" project*[1] *was the classification of musical genre from polyphonic audio files. This is a task that tests the application of Machine Learning (ML) methods to Digital Signal Processing (DSP), albeit in the univariate domain. It is also potentially an area in which sparsity can be exploited, as we are given prior knowledge that the signal was created by a finite set of instruments, be they physical or electronic, and that the degrees of freedom at any one time are far less than the sampling rate of the audio files.* ***Radar*** *The next application area was a study of how the Analogue to Digital Conversion (ADC) sampling rate in a digital radar can be reduced—without reduction in waveform bandwidth—through the use of Compressed Sensing (CS). Real radar data is used to show that through use of chirp or Gabor dictionaries and Basis Pursuit (BP) the Analogue to Digital Conversion (ADC) sampling frequency can be reduced by a factor of 128, to under 1 mega sample per second, while the waveform bandwidth remains 40* MHz*. The error on the reconstructed fast-time samples is small enough that accurate range-profiles and range-frequency surfaces can be produced.*

4.1 Introduction

Before moving on to multivariate signal processing (see Chapter 5), a natural stepping stone is to test some of the ML methods described to this point on univariate signals. By this it is meant that the signal of interest is characterised by a single variable that is varying through time. This variable may come from a sensor or be a direct digital instantiation of a signal. It is important to distinguish the terms univariate and multivariate with respect to signals with the same terms as they are used in general mathematical (and indeed ML) nomenclature. The processing and analysis of the signals will certainly be multidimensional, and hence multivariate, even though the originating signal was univariate. Throughout most of the Chapter, the signals will be treated as if they can be broken down into small enough segments

[1] EPSRC ICT project reference: EP/D063612/1

such that the temporal shift from one segment to the next is small in comparison with the variation within the signal. This allows the signals to be modelled using descriptions based on short-term features.

The first part of the Chapter examines the classification of musical genre from raw audio files. Although most musical files are produced in stereo format (hence bivariate), for the purposes of this study the files were downsampled to a mono format (univariate). This is justifiable in this setting as it is clear that humans do not require stereo information to differentiate betweeen genres. It will be shown that sparse ML methods are advantageous in this setting. The rest of the Chapter examines the application of CS to conventional radar. Again the signals are univariate, but in this case with a much higher frequency. Here the focus is on DSP, although the methods used are directly applicable in ML settings as well, and there is scope for further analysis of this data in an ML setting.

4.2 Genre Classification

To begin, an analysis was performed of the state of the art in feature extraction from polyphonic music through the use of DSP techniques. To this end, classification of musical genre from raw audio files (MPEG-1 Audio Layer 3 (MP3) format), as a fairly well researched area of music research, provided a good starting point. The Music Information Retrieval Evaluation eXchange (MIREX) is a yearly competition in a wide range of machine learning applications in music, and in 2005 included a genre classification task, the winner of which [75] was an application of the multiclass boosting algorithm AdaBoost .MH [42]. The method was duplicated, and then modified through the use of LPBoost [5]. The hypothesis is that LPBoost is a more appropriate algorithm for this application due to the higher degree of sparsity in the solutions. The aim was to improve on the [75] result by using a similar feature set and the multiclass boosting algorithm LPBoost .MH. This work was presented at the Neural Information Processing Systems (NIPS) 2007 Workshop “Music, Brain and Cognition” [11].

A music genre is a categorisation of pieces of music that share a certain style. Music is also categorised by non-musical criteria such as geographical origin, though a single geographical region will normally include a wide variety of sub-genres. Any given music genre or sub-genre could be defined by the musical instruments used, techniques, styles, context or structural themes.

The groupings of musical genres and sub-genres leads naturally to the idea of a genre hierarchy. However, the distinctions both between individual sub-genres and also between sub-genres and their parent genres are not always clear-cut. While attempts have been made to automatically construct genre hierarchies (*e.g.* [119, 120, 121]), the performance of such systems do not appear to warrant the additional complexity they entail. In addition, the MIREX set-up uses only flat classifications, and for simplicity and comparability of results the focus of the current research is also flat classification.

One of the problems with the grouping of musical pieces into genres is that the process is subjective and is directly influenced by the individual’s musical background. This is especially true in sub-genres. Another difficulty is that a single artist or band will often span multiple genres or sub-genres (sometimes intentionally), often within the space of a single album (and in some cases a single song!). It becomes

virtually impossible to classify the artist or the album into a single genre. Further confusing the matter is that some genre labels are quite vague and non-descriptive. For example, the genres *world* and *easy listening* are often used a catch-all for music that does not fit naturally into more common genres such as *rock* or *classical* (which are themselves extremely broad and rather vague!). There are additional problems that have been noted, such as the "producer effect" or "album effect" [122], where all of the songs from a single album share overall spectral characteristics much more than from other albums from the same artist. This can even extend to greater similarities between artists sharing the same producer than between the artist's albums. Despite these issues, the automatic classification of new material into existing genres is of interest for commercial and marketing reasons, as well as generally for ML researchers.

The performance of humans in classifying musical genre has been investigated in [123]. In this study participants were trained using representative samples from each of ten genres, and then tested using a ten-way forced-choice paradigm. Participants achieved an accuracy of 53% correct after listening to only 250ms samples and 70% correct after listening to 3s samples. Another study by [124] reports similar results. Although direct comparison of these results with the automatic musical genre classification results of various studies is not possible due to different genre labels and datasets, it is notable that human performance and the automatic retrieval system performance are broadly similar. Moreover, these results indicate the fuzzy nature of musical genre boundaries. It also indicates the difficulty of gathering ground truth annotations, and explains why some datasets appear to be afflicted with particularly poor annotations.

However, probably the main practical problem for research in the field of automatic music classification is the lack of a freely available high quality dataset. Due to legal obstacles it is not possible to publish datasets of popular music in the way that is possible in other fields, such as text recognition. As a result the datasets that are publicly available consist of "white label" recordings which are ostensibly of poorer quality than mainstream recordings (subjectively in terms of musical quality, but objectively in terms of production quality). The present study uses one publicly available dataset (Magnatune) and one provided by a fellow researcher (Anders Meng, see [124]). The former has been used for the MIREX competition on more than one occasion, and the latter has been used in studies [125, 124], which will be used for comparison.

4.2.1 MIREX

The Music Information Retrieval Evaluation eXchange (MIREX) is part of the annual International Conference on Music Information Retrieval (ISMIR). It takes the form of a series of competitions that have been running since 2004. The 2005 competition included an Audio Genre Classification task, in which the task was classification of polyphonic musical audio into a single high-level genre per example. The audio format for the task was MP3, CD-quality (PCM, 16-bit, 44100 Hz), mono. Full files were used, with segmentation being done at authors' discretion.

Although the categories were organised hierarchically, submitted software was only required to

produce classifications of leaf categories. This means that entrants did not implement hierarchical classification and could treat the problem as a flat classification, effectively ignoring the hierarchy. The hierarchical structure was suggested because this reflects the natural way in which humans appear to organise genre classifications, and it allows hierarchical classification techniques if desired. The approach taken at MIREX had the advantage of allowing entrants to treat the problem as either a flat or hierarchical classification problem. In addition all of the recordings used belong to one and only one category.

Two sets of data were used, 'Magnatune'[2] and 'USPOP'[3]. The Magnatune dataset has a hierarchical genre taxonomy, while the USPOP categories are at a single level. The audio sampling rates used were either 44.1 KHz or 22.05 KHz (mono). More data information is in the following table:

The results for MIREX 2005 are summarised in table 4.1 below (see the contest wiki [4] for full results). It should be noted that the statistical validity of the results of the MIREX competitions have recently been called into question [126], due to the testing methods employed. The result is that the reported test accuracies are artificially high, so care must be taken when making direct comparisons.

Participant	Algorithm	Features	Score
Bergstra et al.	AdaBoost	Aggregated features	82.23%
Mandel & Ellis	SVM	KL-Divergence	78.81%
West	Trees,LDA	Spectral & Rhythmic	75.29%
Lidy & Rauber	SVM	Spectral & Rhythmic	75.27%
Pampalk et al.	1-NN	MFCC	75.14%
Scaringella	SVM	Texture & Rhythmic	73.11%
Ahrendt & Meng	SVM	Auto-Regression	71.55%
Burred	GMM/ML	Aggregated features	62.63%
Soares	GMM	Aggregated features	60.98%
Tzanetakis	LSVM	FFT/MFCC	60.72%

Table 4.1: *Summary of results of the Audio Genre Classification task from MIREX 2005 (Mean of Magnatune Hierarchical Classification Accuracy and USPOP Raw Classification Accuracy)*

4.2.2 Feature Selection

The various methods for classifying musical genre generally differ in the way that acoustic features are selected, how sub-song level features are aggregated into song-level features, and the machine learning techniques used to classify based on the features. This Section describes briefly some different approaches to feature selection, followed by a more detailed examination of the approach taken by [75]. The techniques that are employed for extracting acoustic features from musical pieces are inspired by speech perception, signal processing theory, and music theory. In most cases the audio waveform is broken into short frames (in the case of [75] these were 46.44ms in length, or 1024 samples of audio at 22050Hz), and then frame level features are constructed. These frames are then assumed to be independent draws from a Gaussian distribution over features. Whilst this assumption is clearly false, it is a simplifying assumption that allows a range of ML methods to be applied, such as the Support Vector Machine (SVM) or AdaBoost .

[2] http://www.magnatune.com
[3] http://www.ee.columbia.edu/ dpwe/research/musicsim/uspop2002.html
[4] http://www.music-ir.org/evaluation/mirex-results/audio-genre/index.html

4.2.3 Frame level features

The frame level features that are used to describe the audio signal are described below.

Discrete Fourier Transform (DFT)

The DFT is an application of the Fourier Transform (see 2.70 in Section 2.2.1) on digitised data. Fourier analysis is used to analyse the spectral composition of the frames. Given a signal of length T, the DFT and the inverse operation (Inverse Fourier Transform (IFT)) are defined as,

$$\hat{f}(d) = \sum_{t=0}^{T-1} f(t) \exp \frac{-i2\pi dt}{T}, \quad d = 1, \ldots, T \quad \text{(DFT)}, \tag{4.1}$$

$$f(t) = \frac{1}{T} \sum_{d=0}^{T-1} \hat{f}(d) \exp \frac{i2\pi dt}{T}, \quad t = 1, \ldots, T \quad \text{(IFT)}. \tag{4.2}$$

A 512-point transform of each frame was performed, of which the lowest 32 coefficients were retained during experiments. In practice a Fast Fourier Transform (FFT) is used, which is a reorganisation of the calculation that involves $\mathcal{O}(T \log_2 T)$ calculations instead of $\mathcal{O}(T^2)$ [127].

Real Cepstral Coefficients (RCC)

The motivation behind 'cepstral' analysis is the source/filter model used in speech processing. It is used to separate the source (the voicing) from the filter (the vocal tract). In musical instruments the source would be the excitation impulse caused by for example plucking a string, and the filter would be the reverberations from the body of the instrument. In general, a spectrum can be seen as having two components - a slowly varying part (the filter or spectral envelope) - and a rapidly varying part (the source or harmonic structure). These can be separated by taking a further Fourier Transform of the spectrum. This is known as the 'cepstrum' (which is an anagram of spectrum), and is said to be in the 'quefrency' domain (an anagram of frequency). Formally, the real cepstrum of a signal is defined as:

$$z^{RCC} = \text{real}\left(f\left(\log\left(|\hat{f}(t)|\right)\right)\right) \tag{4.3}$$

where $\hat{f}(\cdot)$ is the Fourier transform and $f(\cdot)$ is the inverse Fourier transform.

Mel Frequency Cepstral Coefficients (MFCC)

The MFCC is a measure of the perceived harmonic structure of the sound. It is similar to the RCC, except that the input x is first projected according to the Mel-scale [128]. The name Mel comes from the word melody to indicate that the scale is based on pitch comparisons. A Mel is a psychoacoustic unit of frequency which relates to human perception, the Mel scale can be approximated from the frequency q in Hz by,

$$m(q) = 1127.01048 \log\left(\frac{1+q}{700}\right). \tag{4.4}$$

Zero Crossing Rate (ZCR)

The ZCR of a signal is the rate of sign changes along the signal. This is a measure which for a single instrument is correlated with dominant frequency [129] (*i.e.* it is a primitive pitch detection routine). The meaning of this measure is less clear for polyphonic music, but it is included for completeness. Defining the indicator variable $v(t)$ as

$$v(t) = \begin{cases} 1, \ f(t) \geq 0, \\ 0, \ f(t) < 0 \end{cases} \tag{4.5}$$

and the squared difference $g(t_1, t_2) = (v(t_1) - v(t_2))^2$ then the ZCR over a frame is calculated as

$$z^{ZCR} = \frac{1}{T-1} \sum_{t=1}^{T-1} g(t, t-1). \tag{4.6}$$

The complexity of the ZCR amounts to $\mathcal{O}(T)$ and is the cheapest of the features discussed to extract.

Spectral Centroid

The spectral centroid describes the center of gravity of the octave spaced power spectrum and indicates whether the spectrum is dominated by low or high frequencies. It is related to the perceptual dimension of timbre. Given the Fourier transform $\hat{f}(t)$, the spectral centroid is formulated as,

$$z^{ASC} = \frac{\sum_{t=0}^{T-1} t|\hat{f}(t)|^2}{\sum_{t=0}^{T-1} |\hat{f}(t)|^2} \tag{4.7}$$

Spectral Spread

The audio spectrum spread describes the second moment of the log-frequency power spectrum. It indicates whether the power is concentrated near the centroid, or if it is spread out in the spectrum. A large spread could indicate how noisy the signal is, whereas a small spread could indicate if a signal is dominated by a single tone. The spectral spread is formulated as,

$$z^{ASS} = \frac{\sum_{t=0}^{T-1} (t - z^{ASC})^2 |\hat{f}(t)|^2}{\sum_{t=0}^{T-1} |\hat{f}(t)|^2} \tag{4.8}$$

Spectral Roll-off

Spectral roll-off is defined as the a-quantile of the total energy in $\hat{f}(t)$. In other words, it is the frequency under which a fraction of a of the total energy is found, and is defined as

$$z^{RO} = \max \left\{ z : \sum_{t=0}^{z} |\hat{f}(t)|^2 \leq a \sum_{t=0}^{T} |\hat{f}(t)|^2 \right\} \tag{4.9}$$

The spectral roll-off was calculated at 16 equally spaced thresholds in the interval $[0, 1]$.

Autocorrelation

The ℓ Linear Predictive Coefficients (LPC) and the Correlation Coefficient (LPCE) of the (original) signal x are defined as:

$$z^{LPC} = \arg\min_{a} \sum_{t=1}^{T} (x_t - \sum_{i=1}^{\ell} a_i x_{t-i}) \tag{4.10}$$

$$z^{LPCE} = \min_{a} \sum_{t=1}^{T} (x_t - \sum_{i=1}^{\ell} a_i x_{t-i}) \tag{4.11}$$

which is equivalent to an autoregressive compression of spectral envelope. The LPC can be efficiently computed using Levinson-Durbin recursion.

4.2.4 Feature Aggregation

In order to convert the sub-song level feature sets into a manageable feature set for statistical pattern analysis, some form of aggregation of sub-song level features into a single song-level feature set is required.

Gaussian Features

Possibly the simplest approach is to calculate the mean and standard deviation over segments, which amounts to fitting a single Gaussian distribution with diagonal covariance over the features of the data. The resulting full feature vector is created by concatenating the means and variances of 256 RCC, 64 MFCC, 32 LPC, 1 LPCE, 32 FFT, 16 roll-off, and 1 ZCR. This leads to $402 \times 2 = 804$ parameters for each song.

Autoregression (AR) Features

Another idea is to try to incorporate some of the temporal information over the length of each song into the feature aggregation. Genres may, for example, be defined more by changes in their spectral qualities than the average of those given by the Gaussian fitting. Autoregression (AR) coefficients can be calculated with an all-pole model using the Yule-Walker method. This method uses Levinson-Durbin recursions on the biased estimate of the sample autocorrelation sequence to compute the coefficients [130]. Using a 10^{th} order model and ignoring the zeroth order component results in $402 \times 10 = 4020$ parameters for each song. This method was used on the smaller of the two datasets presented here in combination with the Gaussian feature aggregation.

4.2.5 Algorithms

The empirical testing here will follow [75] by using multiclass AdaBoost (AdaBoost .MH), as was introduced in Section 2.1.11, in combination with aggregated features. However as the number of features is already large before the creation of weak learners, which will result in a larger number of weak classifiers

for the boosting algorithm to choose from, it may be the case that an algorithm that enforces sparsity in the solutions would be preferable. The natural extension is therefore to use the LPBoost algorithm, as introduced in Section 2.1.11.

Multiclass

Both AdaBoost and LPBoost must be extended to cope with the multiclass setting presented here. Any binary classifier can be turned into a multiclass classifier using the "one-versus-rest" approach, where binary classifiers are built for each class versus the rest, and the classifier that gives the most positive decision value (or least negative in the case that all are negative) is the class label given. This is the first approach taken for LPBoost , and AdaBoost is extended in a similar manner to give the algorithm AdaBoost .MH (see Algorithm 1).

Uneven loss function

Multiclass classification problems in the one-vs-rest framework are inherently unbalanced, as the class which is being classified will tend to have far fewer members than the rest of the dataset. Both AdaBoost and LPBoost can be modified with uneven loss functions to try to mitigate against this problem. This involves increasing the weight of false negatives more than false positives, and decreasing the weight of true positives less than true negatives. The result of this is that positive examples maintain higher weight (misclassification cost). This leads to two new algorithms known as AdaUBoost and LPUBoost [131].

Another approach to Multiclass classification is to map the outputs to binary codes using Error-Correcting Output Codes (ECOC) [132]. This theoretically should aid classification as it overcomes the standard one-versus-rest imbalance. Experiments were conducted using this method, but it was found that due to the small number of classes in the present experiment (4, 6, or 11), no difference in performance was observed. In fact for the smallest number of classes (4), the performance was actually worse. This is most likely due to the artificial way in which the ECOC encoding partitions the data.

4.2.6 Multiclass LPBoost Formulation (LPMBoost)

This section details the formulation of a new multiclass extension, to be called LPMBoost , of the LPBoost algorithm in which the original objective function is such that the margin between the correct class and each of the incorrect classes is maximised. It is similar in flavour to the multiclass extension of the SVM [133], and also resembles the linear programming formulation of structured output learning over a path [134]. However to the author's knowledge this extension of LPBoost to the multiclass setting is novel. Let k be the number of classes, where $k > 2$. Let $y^k \in \{-1, 1\}$ be the vector of labels for the one versus rest classification for class j where $j = 1, \ldots, k$. $\tilde{y} = [y^1; \ldots; y^k]$ is the vertical concatenation of these vectors into a column vector of length mk. The goal is then to maximise the margin γ (or minimise the negative margin) between the output of the correct class and that of the other classes,

i.e. $\forall i, \mathbf{H}_{i,\cdot}(\mathbf{w}_s - \mathbf{w}_{\hat{s}}) \geq \gamma, \hat{s} \neq s = y_i$. This is done by replacing matrix $\mathbf{H}$ (where $\mathbf{H} = \sum_i y_i h(x_i, \cdot)$) in the first constraint of the primal (2.59) and dual (2.60) formulations with another matrix $\mathbf{M}$.

The matrix $\mathbf{M}$ is formed by augmenting the hypothesis matrix into a large matrix with all of the necessary comparisons (*i.e.* the hypotheses for the correct class for each example versus the negative of the hypotheses for every other class). The rows are generated that correspond to an example with a negative label, of which there are $k-1$ for each example, and a zero row for the comparison of the hypothesis with itself. The zero row will create a constraint that can't be satisfied, so although this could be mopped up by the slack variable, it is better to remove it from the matrix, giving a total of $m(k-1)$ rows. The weak learners correspond to a weak learner for a particular class, and as such there are nk columns. An example matrix is given in Table 4.2. Learning is then performed using the standard LPBoost algorithm. At the testing stage, given a matrix M_{test} containing one row for each test point and one column for each weak learner, and the set of chosen weak learners $\mathbf{i}$ and primal weights $\mathbf{w}^j, j = 1, \ldots, k$ (the Lagrange multipliers from the final step of the dual optimisation for each class k), the decision function is now simply,

$$\hat{f}^j = \mathbf{M}_{test}[:, \mathbf{i}]\mathbf{w}^j, \quad j = 1, \ldots, k, \tag{4.12}$$

and the classification is then given by,

$$\hat{y} = \arg\max_{j=1,\ldots,k} \left\{ \hat{f}^j \right\}. \tag{4.13}$$

Example	Comparison	y_i	Class 1	Class 2	Class 3
1	1v2	1	$\mathbf{h}_1$	$-\mathbf{h}_1$	0
	1v3		$\mathbf{h}_1$	0	$-\mathbf{h}_1$
2	1v2	2	$-\mathbf{h}_2$	$\mathbf{h}_2$	0
	2v3		0	$\mathbf{h}_2$	$-\mathbf{h}_2$
3	1v3		$-\mathbf{h}_3$	0	$\mathbf{h}_3$
	2v3		0	$-\mathbf{h}_3$	$\mathbf{h}_3$
4	1v2	2	$-\mathbf{h}_4$	$\mathbf{h}_4$	0
	2v3		0	$\mathbf{h}_4$	$-\mathbf{h}_5$

Table 4.2: *An example of the augmented hypothesis matrix* $\mathbf{M}$*. In this example there are four examples with class labels* $\mathbf{y} = \{1, 2, 3, 2\}'$ *and corresponding weak learner vectors* $\mathbf{h}_1, \ldots, \mathbf{h}_4$*, which are row vectors of weak learners* $\mathbf{h}_i = \left\{h_{i(1)}, \ldots, h_{i(n)}\right\}$

4.2.7 Experiments

The dataset used in the MIREX 2005 genre classification task is not freely available due to licensing issues. Experiments were run using two datasets: an older Magnatune 2004 dataset which is publicly available and a dataset provided by Anders Meng [124]. These are described below.

Magnatune 2004

The RWC Magnatune database used for the MIREX 2004 Audio description contest is still available (see [135]). Whilst this suffers from many of the problems discussed at the beginning of this chapter, it has the advantage of being released under the slightly more lenient framework of the "Creative Commons". The dataset is split into 6 genres (*classical*, *electronic*, *jazz & blues*, *metal & punk*, *rock & pop*, and *world*).

Anders Meng dataset d004

Dataset consisting of 11 genres, with 1100 training examples and 220 test examples. The integrity of the data-set has been evaluated by humans (experts and non-experts) at a decision time horizon of 30 seconds [124]. It is interesting to note that human performance on this dataset is only at 57.2% in a 11-way forced choice paradigm (see 4.1). This suggests that either the ground truth annotations are inaccurate or that the genre labels are not very descriptive. The genres in the dataset are *alternative*, *country*, *easy listening*, *electronica*, *jazz*, *latin*, *pop/dance*, *rap/hip-hop*, *R&B/soul*, *reggae*, *rock*. However, the dataset was used with some success in previous studies [136, 137]. During the evaluation of this method, the full dataset

	alternative	country	easy-listening	electronica	jazz	latin	pop&dance	rap&hiphop	rb&soul	reggae	rock
alternative	16.0	2.7	9.3	9.3	1.3	0.0	32.0	0.0	4.0	2.7	22.7
country	5.3	54.7	9.3	0.0	4.0	1.3	9.3	0.0	4.0	0.0	12.0
easy-listening	17.3	0.0	34.7	8.0	12.0	0.0	13.3	5.3	2.7	0.0	6.7
electronica	5.3	0.0	0.0	54.7	1.3	0.0	32.0	1.3	4.0	1.3	0.0
jazz	5.3	0.0	5.3	4.0	70.7	6.7	2.7	1.3	4.0	0.0	0.0
latin	2.7	0.0	8.0	5.3	5.3	56.0	14.7	0.0	5.3	2.7	0.0
pop&Dance	4.0	1.3	10.7	10.7	0.0	1.3	62.7	0.0	5.3	1.3	2.7
rap&hiphop	1.3	0.0	5.3	1.3	1.3	1.3	1.3	80.0	6.7	0.0	1.3
rb&soul	2.7	1.3	13.3	1.3	2.7	0.0	14.7	0.0	57.3	2.7	4.0
reggae	5.3	0.0	0.0	4.0	0.0	0.0	1.3	5.3	2.7	81.3	0.0
rock	12.0	1.3	9.3	0.0	1.3	2.7	8.0	1.3	2.7	0.0	61.3

Figure 4.1: *Confusion Matrix of human performance on Anders Meng dataset d004*

of all 11 genres was used along with a subset of this containing the 4 genres that had the highest rate of accuracy for human performance (*jazz*, *pop/dance*, *rap/hip-hop*, and *reggae*). The reasoning behind this was that if the main problems encountered with this dataset were based on inaccuracies or vagaries of the ground truth labelling, these would be reduced by taking the most consistent results from human evaluation.

4.2.8 Results

In all the experiments the AdaBoost stopping parameter was selected by 5-fold CV. The average classification accuracies of the different algorithms on the datasets are shown in Tables 4.3 and 4.4. The labels for the datasets are as follows: MAGNA6 refers to the Magnatune database (6 classes); MENG4 refers to the reduced Anders Meng dataset, where the 4 classes with the highest accuracy of human performance were chosen.

MAGNA6 (6 classes)	
Algorithm	**Accuracy**
AdaBoost	59.3%
AdaUBoost	59.8%
LPBoost	55.1%
LPUBoost	57.8%
LPMBoost	**60.9%**

Table 4.3: *Average 6-class classification accuracy on Magnatune 2004 dataset using AdaBoost , LPBoost , and LPMBoost classifiers*

Due to the large size of the MAGNA6 dataset only the Gaussian feature aggregation method was used. The results show that, somewhat against expectations, the performance of LPBoost is actually worse than that of AdaBoost . The modifications for the uneven nature of the dataset due to the one-versus-rest classification, LPUBoost and AdaUBoost , both resulted in slight improvements in classification accuracy, and narrowed the difference between the two algorithms. However the best performance on the dataset was obtained by the LPMBoost algorithm, which directly optimised the margin between the multiple classes whilst enforcing sparsity.

	MENG4 (4 classes)		
Algorithm	**Gaussian features**	**AR features**	**All features**
AdaBoost	41.2%	35.0%	46.2%
AdaUBoost	42.5%	35.0%	50.0%
LPBoost	46.2%	35.0%	43.8%
LPUBoost	46.2%	35.0%	47.5%
LPMBoost	43.8%	38.7%	**53.8%**

Table 4.4: *Average 4-class classification accuracy on MENG(4) dataset using AdaBoost , LPBoost , and LPMBoost classifiers*

For the MENG4 dataset both the Gaussian feature aggregation and the Autoregressive feature aggregation were used individually, and also together. For the Gaussian feature aggregation method, the LPBoost algorithm performed better than the AdaBoost algorithm, and in this case with only 4 classes the uneven modifications AdaUBoost and LPUBoost made little or no difference. In this case the LPMBoost algorithm performed slightly worse than the standard LPUBoost algorithm. For the Autoregressive feature aggregation method the overall classification accuracy was somewhat lower than for the Gaussian feature aggregation method in all cases, with the LPMBoost algorithm performing the best in this case. Interestingly, by combining the two feature extraction methods together, the performance of the algorithms was improved in nearly all cases. As with the MAGNA6 dataset, when using all features the AdaBoost algorithm initially outperformed the LPBoost algorithm, and again the AdaUBoost and LPU-

Boost modifications improved classification accuracy. Once again, however, the LPMBoost algorithm gives the best overall classification accuracy, which demonstrates the efficacy of this method. In general, the performance of all of the algorithms on this dataset is lower than may be expected. However, results of human performance cited in [124] suggest that the dataset is extremely difficult to classify - possibly indicating that the ground truth labelling is inaccurate, or that there are other confounding factors.

4.3 Compressed Sensing for Radar

This Section presents a study of how the Analogue to Digital Conversion (ADC) sampling rate in a digital radar can be reduced—without reduction in waveform bandwidth—through the use of Compressed Sensing (CS). Real radar data is used to show that through use of chirp or Gabor dictionaries and Basis Pursuit (BP) the ADC sampling frequency can be reduced by a factor of 128, to under 1 mega sample per second, while the waveform bandwidth remains 40 MHz. The error on the reconstructed fast-time samples is small enough that accurate range-profiles and range-frequency surfaces can be produced.

CS is a new paradigm in Digital Signal Processing (DSP) that trades sampling frequency for computing power and allows accurate reconstruction of signal sampled at rates many times less than the conventional Nyquist frequency [59, 69]. This new technique has been applied successfully in Synthetic Aperture Radar (SAR) to both achieve higher resolution images [138, 139] and to reduce the number of measurements made of the backscatter signal, which in turn reduces data transfer and storage requirements [140, 141]. Additionally there have been studies made of how CS can be used to reduce the sampling requirements of Ultra Wide Band (UWB) radar systems [142, 143] although the latter of these did not consider the impact of the Doppler shift on the CS algorithm, and both have been conducted entirely with simulated data.

In this work, the CS approach used in [143, 10] will be extended to include processing of data that includes Doppler shifts. Additionally, data from a real radar system will be used that includes noise and non ideal measurement conditions, such as the presence of clutter, small amounts of interference and clipping of the signal at the ADC. The form of CS being employed is AIC [9, 10] that reduces the sampling frequency from the traditional Nyquist rate by sampling at the information rate, rather than the rate required to accurately reproduce the baseband signal.

Conventional sampling theory requires that digital samples of an analogue signal be measured at a rate sufficient for the signal to be reproduced without aliasing, this is the Nyquist frequency. Sampling in this way is concerned purely with accurate reconstruction of the signal and does not consider that the information contained within the signal that is really important. It is likely that the true information rate is much lower than the Nyquist frequency, and so long as the sampling approach captures this information then the original signal can be reconstructed. It is important to realize that while the sampling frequency has been reduced, the computational overhead has increased since it is now required that the original signal be reconstructed. Such a trade may be desirable in radar applications to allow relaxation of the sampling requirements to reduce cost or to permit gaps to be left in the radar bandwidth [144] that might

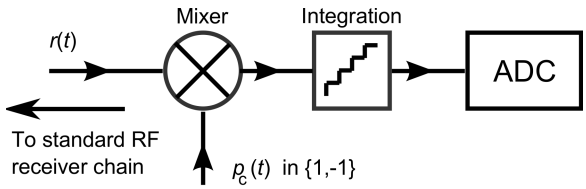


Figure 4.2: *The modified receiver chain for CS radar.*

then be used in other applications. These possibilities make the study of CS for regular radar applications attractive.

The principal contributions of this study are the use of real radar data in a CS study and the consideration how the Doppler shift affects reconstruction in the AIC approach.

4.3.1 Review of Compressive Sampling

This section provides a brief review of the theory of Compressed Sensing (CS) as first introduced in Section 2.2.4, a technique that allows signals to be acquired or reconstructed sparsely, by using prior knowledge that the signal is sparse in a given basis [59, 69]. The principal result is that signals can be reconstructed exactly even with data deemed insufficient by the Nyquist criterion. Formally, given a signal $x \in \mathbb{R}^n$ and a dictionary $\Psi \in \mathbb{R}^{n \times d}$ which forms an orthonormal basis, x is said to be sparse if x can be represented as a linear combination of k atoms from Ψ, *i.e.* $x = \sum_{i=1}^{k} \alpha_i \Psi_{.,i}$ where $k \ll d$. According to the CS theory it is possible to construct a measurement matrix $\Phi \in \mathbb{R}^{m \times n}$ with $m \ll n$, and perform stable reconstructions of the signal from measurements y, where $y = \Phi\Psi\boldsymbol{\alpha}$, if the measurement matrix is incoherent with the dictionary.

This principle of incoherence extends the duality between the time and frequency domains. For CS we need a stable measurement matrix Φ and a reconstruction algorithm to recover x from y. The Restricted Isometry Property (RIP) describes a sufficient condition for a stable solution for both k-sparse and compressible signals [59]. It has been shown that i.i.d. random Gaussian and Bernoulli matrices satisfy both the RIP and incoherence conditions with high probability [59] (see also Section 2.2.5).

This study used a form of ℓ_1-penalised least squares known as BP, which has been shown to approximate the $k-$sparse ℓ_0 solution [31].

$$\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \Phi\Psi\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 , \tag{4.14}$$

BP can be solved using the LARS [32]. LARS computes the full regularisation path, which is a piecewise linear function between $\lambda = 0$ and $\lambda = \infty$ (as described in Section 2.1.7).

Details of the dictionaries and measurement matrices used are given in Section 4.3.3.

4.3.2 Application of CS To Radar

To allow the ADC to run at a sub-Nyquist rate, the radar receiver chain must be modified to allow CS. Figure 4.2 shows the additional components required for an AIC receiver. After the standard filters,

downconverters and amplifiers, but before the ADC, two new components are added—another mixer and an integration filter. The first input of the mixer, $r(t)$, is the baseband signal. The second input is a pseudo-random signal, $p_c(t)$, that can take a value of either 1 or -1. Such a signal can be readily generated using direct digital synthesis. Following the mixer is an integration filter that sums the output of the mixer over an interval, T_{CS}:

$$T_{CS} = NT_{sample} \tag{4.15}$$

where T_{sample} is the Nyquist sampling interval and N the undersampling factor. This process of mixing and then summing the signal constitutes the projection of the received backscatter signal onto the measurement basis, Φ, that is defined by $p_c(t)$, see Section 4.3.1. The algorithm, and seed, of the random number generator used to create $p_c(t)$ must be known, since a replica of the signal is needed during the reconstruction of $r(t)$.

Each output of the AIC is a projection of the baseband signal received during the interval T_{CS} on to the measurement basis. The AIC samples emerge at a rate of $\frac{1}{T_{CS}}$. These slower-rate samples cannot be used in the conventional processing that may follow digitisation, such as matched filtering and Fourier analysis, as they stand. Instead, the fast-time samples must be reconstructed using CS. To achieve this, multiple observations of the target area are required. Fortunately, the radar already gathers these observations since in pulsed, or Frequency Modulated Continuous Wave (FMCW) radar, the same waveform is transmitted repeatedly. Only one set of fast-time samples will be reconstructed from the multiple observations: so while the radar operates with one Pulse Repetition Frequency (PRF) the emerging range profiles have a different, lower, PRF. The ratio of the two PRFs will be the number of pulses used to reconstruct the fast-time samples. This reduction in the PRF will ultimately reduce the range of Doppler frequencies that may be detected.

It is possible to synthesise the AIC approach to radar processing using data gathered with a conventional digital radar. During data collection, the baseband signal is digitised with an ADC that runs at the Nyquist frequency. Once the samples have been stored, mixing with the signal $p_c(t)$ and the subsequent integration are performed digitally. The output of this pre-processing will produce samples comparable to those that would be output by a true AIC receiver. This was the approach taken for this study.

4.3.3 Experimental Approach

The Radar Dataset

Data was gathered using a single node of University College London (UCL)'s NetRAD radar [145]. The radar had a 2.4 GHz carrier frequency and was set to transmit a linear Frequency Modulated (FM) pulse, with width 0.6 μs and a 40 MHz bandwidth, and to use a PRF of 20 kHz. The ADC digitised the baseband signal at 100 mega-samples per second, *i.e.* $f_s = 100\ MHz$, and 128 samples were collected per pulse. There was a delay in starting the ADC, so that the transmitted signal would not be recorded, resulting in ranges between $90m$ and $280.5m$ being measured. The targets were placed at range $120m$. When moving, the velocity of the target was along the radar Line Of Sight (LOS) and always towards the

radar. Three targets were used: a stationary flat metal plate; a running person; and a transit van travelling at $15mph$. For the flat plate, 40,000 pulses were recorded while for the moving targets the number was increased to 60,000.

Specific CS Implementation

The AIC was implemented entirely in post processing, as described in Section 4.3.2. The 128 fast-time samples collected during each pulse were compressed into a single sample *i.e.* the integration duration was $128 \times f_{sample}$, and the under sampling factor, N, was 128. This meant that if the AIC had been implemented in hardware, rather than software, the ADC would have needed a sampling rate of under 1 mega sample per second, a substantial reduction over the data capture card used in NetRAD. The random Bernouilli signal $p_c(t)$ was generated using the Matlab functions `randn` and `sign`.

The fast-time samples were reconstructed using BP, see Section 4.3.1. The reconstruction was performed based on 60 compressed samples, or radar pulses, leading to the PRF of the reconstructed data being $333Hz$, one sixtieth of NetRAD's original $20kHz$. Within the BP algorithm the regularisation parameter, λ, was set by taking the value that minimised the reconstruction error on the calibration set.

Chirp atoms were introduced to deal with the nonstationary behavior of the instantaneous frequency of some signals, and shown to form an orthonormal basis [57]. Further, it is clear that the domain in which a radar signal should be most sparse is that composed of delayed and frequency shifted versions of the transmitted signal [143]. A real chirp atom is given by

$$g_{\gamma,\phi,c}(t) = \frac{1}{Z} \cdot g\left(\frac{t-u}{s}\right) \cdot \cos\left(\xi(t-u) + \frac{c}{2}(t-u)^2 + \phi\right) \tag{4.16}$$

where Z is a normalisation factor (to ensure that for each atom $\|g_{\gamma,\phi}\| = 1$), $\gamma_n = (s_n, u_n, \xi_n)$ denotes the series of parameters of the functions of the dictionary, and $g(t) = \exp^{-\pi t^2}$ is the Gaussian window, and c is the chirp rate. The chirp atom has an instantaneous frequency $\omega(t) = \xi + c(t-u)$ that varies linearly with time. For the construction of Φ the parameters of the atoms were chosen from dyadic sequences of integers with the octave parameter $j = 1$ [56]. The Gabor dictionary is constructed in the same way, except that the chirp rate $c = 0$.

Testing Strategy

Each target dataset was processed using the simulated AIC radar system, described in 4.3.2, and both the Gabor and chirp sparse basis. Once the reconstructed fast-time samples had been formed the normalised error between the reconstruction and the actual data could be calculated according to:

$$\epsilon = \frac{\|\mathbf{x}_{orig} - \mathbf{x}_{CS}\|_2}{\|\mathbf{x}_{orig}\|_2} \tag{4.17}$$

where $\mathbf{x}_{orig}$ is the original signal before projection onto the measurement basis and $\mathbf{x}_{CS}$ is the reconstructed signal. In this study, the reconstructed signal was formed from sixty original signals, but for the calculation of ϵ only first signal was used. The use of a mean signal was considered, but averaging

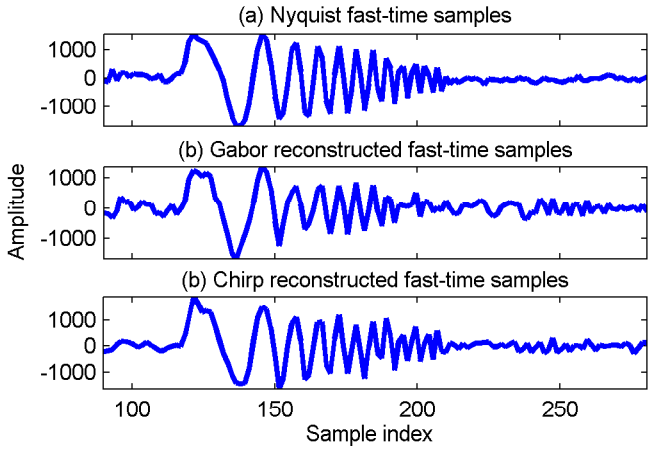


Figure 4.3: *Fast-time samples of the stationary target.*

radar signals is a form of integration that would improve the SNR. This improvement would not be in the reconstructed signal making the comparison unfavourable.

The reconstructed fast-time samples were processed using a conventional matched filter to obtain range profiles, and the DFT was then used to produce range-frequency surfaces.

4.3.4 Results And Analysis

Stationary Target

Initial testing was conducted using the data for the stationary flat-plate target. Measurement of the received signal power indicated that the SNR for the target was $\approx 22dB$. Simulation of the AIC was performed using the Gabor dictionary and the chirp dictionary as the sparse basis for reconstruction. From the original 40,000 pulses 666 reconstructed sets of fast-time samples were reconstructed. During reconstruction the normalised error, see (4.17), had a mean value of 0.70 with a standard deviation of 0.18 for the Gabor dictionary, and 0.58 mean with 0.23 standard deviation for the chirp dictionary. Figure 4.3 shows the reconstructed fast-time samples using the Gabor and chirp dictionaries in parts (b) and (c) respectively, with the samples from the first pulse in the batch of sixty used for reconstruction in part (a) for comparison. In this case, the normalised error was 0.42 for the Gabor dictionary and 0.28 for the chirp. Visual inspection of the figure shows the reflection of the transmitted chirp at a range of $120m$ and both the Gabor and chirp dictionaries appear to reconstruct this part of the curve well (seen as the peaks in Figure 4.3). Conversely, beyond the limits of the reflected chirp the reconstruction appears poor, and it is thought that the majority of the normalised error comes from these regions. Application of a matched filter to the samples resulted in the range profiles shown in Figure 4.4. Again, both the Gabor and chirp dictionary reconstructions, parts (b) and (c) of the figure, are a good match with the Nyquist sampled data, part (a). It was observed that the square root of the peak intensity for the Gabor reconstruction was $\approx 10,000$ less than the actual data, and that for both types of reconstructions the noise regions were much more pronounced.

Observation of the atoms from the two dictionaries used during the reconstruction indicated why

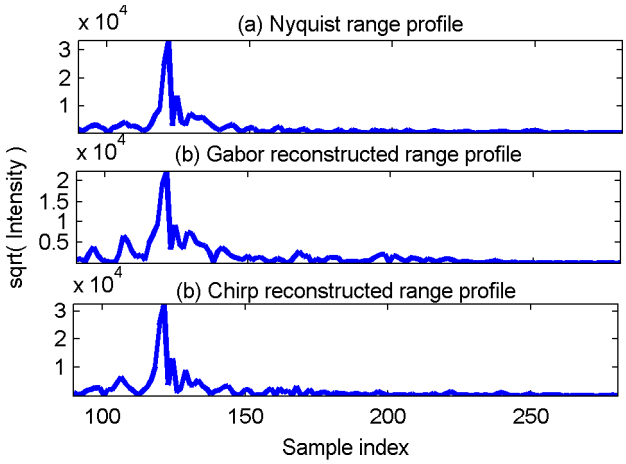


Figure 4.4: *Range profiles of the stationary target.*

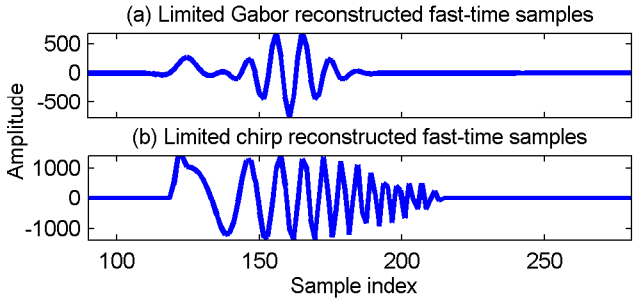


Figure 4.5: *Fast-time samples constructed from largest three coefficients.*

the noise parts of the reconstructed range profiles contained more energy than the original data. In the case of the chirp dictionary it was clear that the most significant atoms used related to the target. Since each atom was a delayed chirp it was straightforward to understand why the BP algorithm had selected it. The most significant atom was at a delay corresponding to the target range. After that there were several atoms, with much smaller amplitude coefficients, distributed throughout the fast-time samples. It was thought these atoms were being used to approximate the thermal noise. In the case of the Gabor dictionary comprehension of the BP process was less certain since the atoms did not correspond directly to the transmitted waveform. There was a series of significant atoms, with narrow scale, that appeared to represent the reflected chirp at the target range. In addition there was a series of atoms with long scale but coefficients indicating a small amplitude; these were attributed to an attempt to reconstruct the thermal noise. Figures 4.5 and 4.6 show the reconstructed fast-time samples and range profiles, respectively, when only the three most significant atoms are used during reconstruction. In both figures part (a) shows the Gabor result and part (b) the chirp. It was observed that the chirp result is almost identical to the full reconstruction, but with less energy in the noise regions, while the limited Gabor reconstruction had not been successful. The ability to reconstruct with fewer atoms in the chirp case suggests a larger regularisation parameter, λ in (4.14), could have been used. In this case the effect of increased sparsity would be that automatic denoising of the signal would be performed during reconstruction.

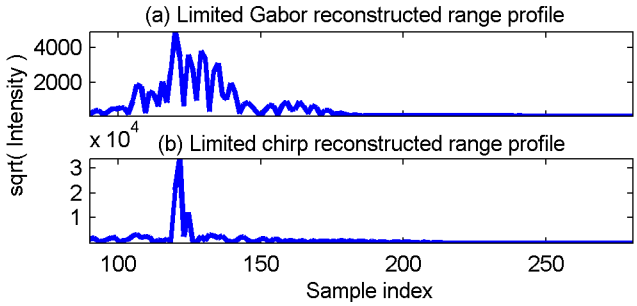


Figure 4.6: *Range profiles constructed from largest three coefficients.*

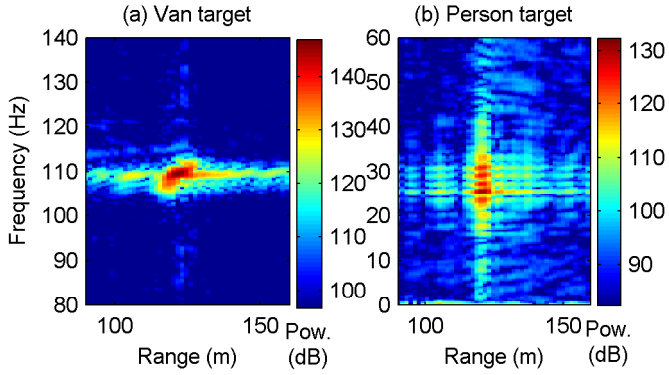


Figure 4.7: *The range-frequency surfaces for the moving targets.*

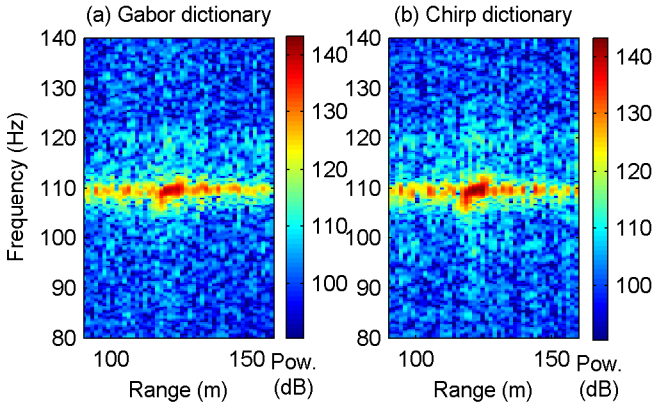


Figure 4.8: *Range-frequency surfaces for van target using CS.*

Moving Targets

When considering moving targets, it is the range-frequency surface that is of interest, rather than the range profile, since it provides information on the target's Doppler shift as well as its range. The surface is calculated by first performing matched filtering of the fast-time samples and then performing a Fourier transform over the pulses in each range-bin. Figure 4.7 shows the range-frequency surfaces for the two moving targets when no CS was employed.

The results for processing the van target data with the simulated AIC are shown in Figure 4.8. It is apparent that there is very little difference between using the Gabor and chirp dictionaries, shown in parts (a) and (b) respectively. Close inspection of the surfaces indicate that the shape of the main peak

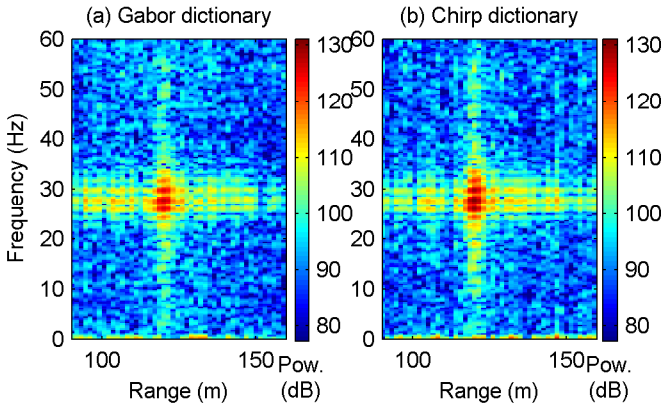


Figure 4.9: *Range-frequency surfaces for person target using CS.*

Table 4.5: *The normalized errors for the moving targets*

Dictionary	Van		Person	
	Av. Error	Std Dev	Av. Error	Std Dev
Gabor	1.094	0.109	0.784	0.143
Gabor top 10	1.051	0.095	0.892	0.133
Chirp	1.153	0.145	0.733	0.189
Chirp top 10	1.120	0.516	0.970	1.197

from the chirp dictionary gave a slightly better match with the original surface (Figure 4.7 part (a)), but the improvement over the Gabor dictionary was only slight. It was also observed that the noise floor for the CS results was higher than in the Nyquist sampled data. This can be seen by comparing the figures.

The running person results are shown in Figure 4.9, again the Gabor dictionary is in part (a) and the chirp, part (b). In this instance it was not possible to discern any difference between the two dictionaries by inspection of the range-frequency surfaces. Both were observed to be a good match with the Nyquist data, although again the surfaces contained more noise than when CS was not used.

The mean normalised errors, and their standard deviations, between the reconstructed fast-time samples and the original Nyquist versions are shown in Table 4.5. The table details the errors for both targets and both dictionaries as well as the cases when reconstruction was performed using only the ten largest coefficients. It was observed that in this instance there was little difference between the two choices of dictionary. For the van target the Gabor dictionary had the lowest error while the chirp was superior for the person. In both instances, however, the difference between errors was in the second decimal place. Furthermore, reducing the number of atoms used in reconstruction did not have an appreciable affect on the error.

Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) is an algorithm for measuring similarity between two sequences which have different temporal extent [146]. DTW has been applied to many different signal processing applications including video, audio, and graphics. A well known application has been automatic speech recognition, where it used to align the signals from speakers with different cadences and inflections (see

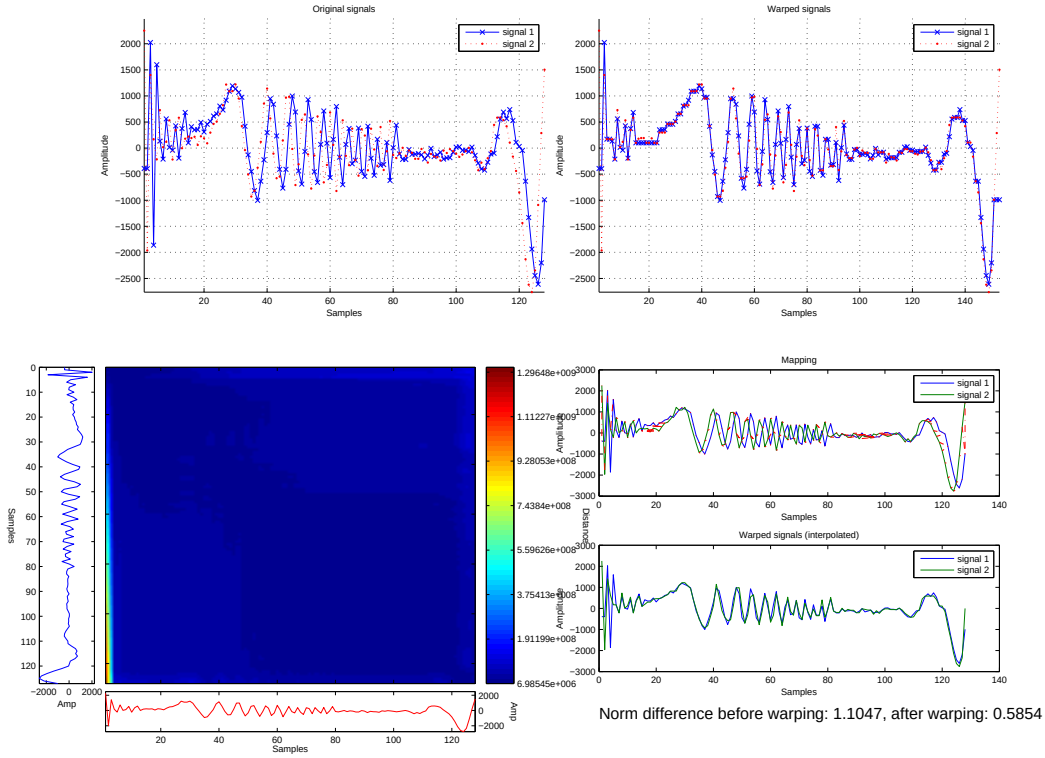


Figure 4.10: *DTW applied to the person target. In this instance the warping has little effect as target is moving slowly, meaning that the warping is minimal. There is, however, still an improvement in the resulting reconstruction (bottom right).*

Chapter 4 of [147]). For targets such as the van target in the present dataset, the deviation between successive fast time samples may become quite large, with an accompanying phase shift due to the Doppler effect. DTW is one possible way of dealing with this. Results of applying the DTW algorithm to successive samples for firstly the person target and then the van target are presented in Figures 4.10 and 4.11 respectively. It can be seen that for the person target, which is slow moving and therefore results in little phase shift or signal offset, the effect of DTW is modest. However for the van target, the effect is much more pronounced. The resulting signal has been realigned such that it is in phase, and the resulting reconstruction is greatly improved. This is demonstrated in Table 4.6, where the results of the improved reconstructions can be seen by the effect they have on outputs of the matched filtering.

Dataset	Original	DTW
Calibration	0.1118	0.0997
Person	0.1237	0.0986
Van	0.1628	0.1138

Table 4.6: *Effect of Dynamic Time Warping (DTW). The figures quoted are the normalised ℓ_2 distances between the results of the matched filter with and without CS. Note that in every case the DTW improves the reconstructions (and hence range-profiles) made by CS*

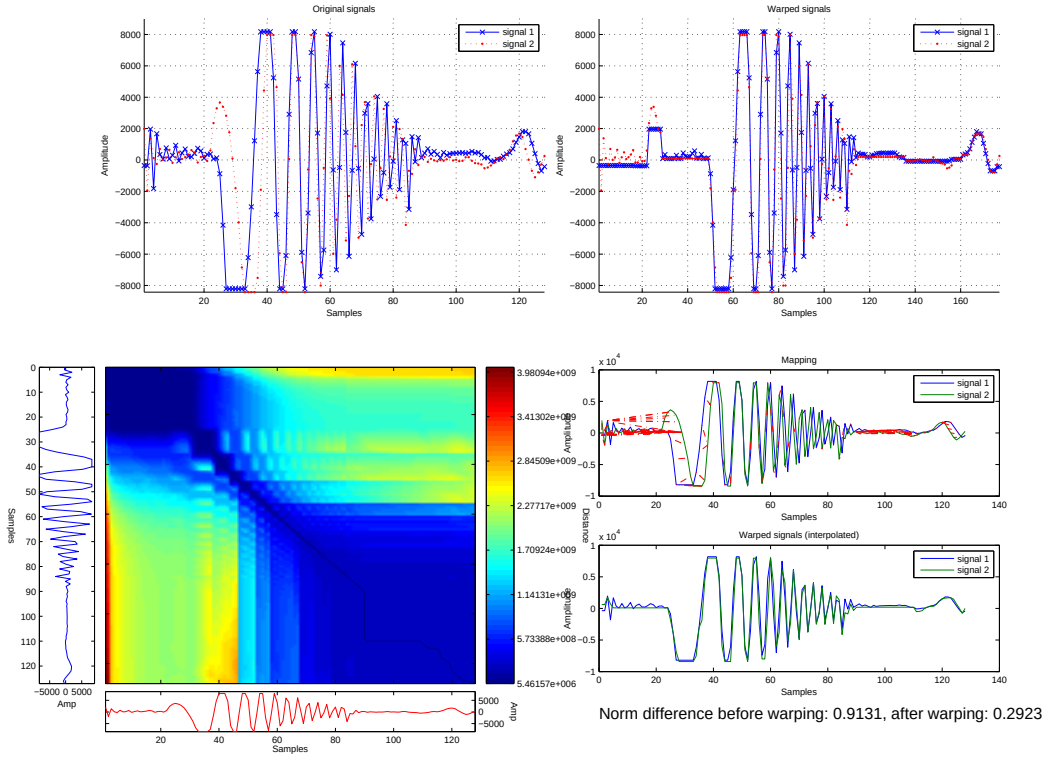


Figure 4.11: *DTW applied to the van target. In this instance the warping has a much greater effect as target is moving more quickly resulting in a bigger deviation between the two signals. The warping here has the effect of realigning the signals such that they are in phase, and the resulting reconstructing is improved greatly (bottom right).*

4.4 Conclusions

The first part of the Chapter examined the classification of musical genre from raw audio files. This was demonstrated through the use of DSP for feature generation and aggregation, and the ML algorithms LPBoost and a novel multiclass extension LPMBoost . It was therefore demonstrated that sparse ML methods are advantageous in this setting.

The rest of the Chapter examined the application of CS to conventional radar. As with the genre classification task, the signals are univariate in the sense of a single sensor or time series, but in this case with a recording frequency orders of magnitude higher. Here the focus is on DSP, although the methods used are directly applicable in ML settings as well, and there is scope for further analysis of this data in an ML setting.

Chapter 5

Applications II

Abstract

This Chapter presents the core application area of the methods described in Chapter 3: Multivariate signal processing. Signals recorded from the brain activity of participants via Electroencephalography (EEG) and Magnetoencephalography (MEG) are both multivariate (there are many sensors) and high frequency (up to 100Hz). As such they present interesting challenges for the application of ML and DSP methods. Additionally, information contained in the stimuli presented to the participant may itself be useful for classification purposes, rather than simple labels. In this situation Multiview methods are required. Two experimental studies will be described: ***Tonality*** *The first is concerned with the task of distinguishing between tonal and atonal musical sequences stimuli through EEG recordings;* ***Genres*** *In the second experiment we seek to detect the genre of music that a listener is attending to from MEG recordings.*

5.1 Introduction

When sensory stimulation reaches the brain, the summed electrical activity of populations of neurons results in characteristic sequences of waves which can be observed in Electroencephalography (EEG) signals. These are known as sensory evoked potentials. One can also measure the corresponding magnetic fields associated with these electrical fields using Magnetoencephalography (MEG). The evoked potentials differ in each sensory modality and also depend on the intensity of the stimulus. They have a very reliable temporal relation to the stimulus onset. Evoked potentials have very low amplitude and are drowned by the ordinary EEG/MEG rhythms. In order to see them, a large number of identical stimuli must be presented and averages taken over all the signals. There are also motor evoked potentials, related to the brain activity preceding movements. Event-Related Potential (ERP) analysis has been primarily used for vision research (*e.g.* [148]) and auditory research (*e.g.* [149]).

However, ERP analysis is not well suited for examining the effects of music, due to the way which

we process musical structures. By definition, a piece of music develops over time and thus engages both short-term and long-term memory systems. The individual responses to particular stimuli (*i.e.* notes or chords) play only a small part in the cognition of a musical piece. Secondly, ERP analysis requires many repetitions of identical stimuli with identical properties (duration, inter stimulus interval, envelope, timbre), which when applied to musical sequences leads to distinctly unmusical sets of stimuli! The first experiment that will be described, conducted in the Leibniz Institute for Neurobiology, suffers from this problem somewhat, as the experimental design was intended for both ERP analysis and the analysis described in this Chapter.

The analysis of brain scans with a view to accurately identifying the semantic processing of the subject has received increasing attention recently [150]. Analysis of subjects listening to music has also received some attention [151] though in some cases this has caused some controversy [152]. This chapter will focus on two experiments: the first is an EEG experiment to examine the brain activity related to the tonal processing of music, and the second is an MEG experiment to examine the brain activity related to the processing of musical genre. In both experiments we will be performing single trial classification. In both cases a similar approach to the classification of time series data will be taken as in the previous Chapter: each "example" will be a segment of data corresponding to a specific musical stimulus (*e.g.* of duration 8 seconds) and features will be calculated for each example using multivariate DSP with feature aggregation. However the major difference is that we will now be attempting to use information from the stimuli themselves to improve the quality of the classifiers using the Multivew methods described in Chapter 3 (Section 3.5).

5.2 Experiment 1: Classification of tonality from EEG recordings

A common structural element of Western tonal music is the change of key within a melodic sequence. The present Section, based on [12] examines data from a set of experiments that were conducted to analyse human perception of different modulations of key. Electroencephalography (EEG) recordings were taken of participants who were given melodic sequences containing changes in key of varying distances, as well as atonal sequences, with a behavioural task of identifying the change in key. Analysis of EEG involved derivation of 122120 separate dependent variables (features), including measures such as inter-electrode spectral power, coherence, and phase. We present a novel method of performing semantic dimension reduction that produces a representation enabling high accuracy identification of out-of-subject tonal verses atonal sequences.

The present study is concerned with the task of distinguishing between tonal and atonal stimuli through the observed EEG recordings of the subjects. It should be stressed that EEG data is notoriously noisy and making reliable cross-subject predictions has proved difficult even for simple tasks. Indeed it will be seen that a naive application of SVMs to the collected signals is unable to make out-of-subject predictions much better than chance, although within-subject predictions were possible. The key contribution will be the demonstration of a novel semantic dimension reduction method that makes use of

a complex description of the stimuli to identify key dimensions in the space of signals that are highly correlated with the stimulus. Using even a simple nearest neighbour classifier in this semantic space can achieve very high accuracy in both within-subject and out-of-subject prediction.

The proposed analysis to discover statistical relationships between musical structure and EEG recordings of participants to the same music is based on the premise that the brain represents structural elements of the auditory signal that it receives through shifting patterns of activity. This activity may take many forms, ranging from generalised changes in activity in certain brain regions to more complex relationships. By taking a multivariate approach to the signal processing of the EEG signal, it is possible to analyse a wide range of such relationships. As such pairwise electrode comparisons, which provide an indication of communication between brain regions, are of paramount importance. The analysis to date has included pairwise statistics such as cross power and coherence. Cross phase is another interesting statistic that will be investigated, as it indicates that there may be an increase (or decrease) in synchrony between brain regions. The collection of statistics derived from the EEG analysis procedure will then be compared with the features derived from the audio recordings in order to seek common patterns.

The encoding of the information about the stimulus is through a kernel designed to capture the melodic and harmonic structure of a musical score available in a simple midi format.

The data under examination in this Section was produced by an EEG experiment conducted in partnership with the University of Magdeburg. The principal hypothesis was that neural patterns should reflect relative changes in the key of music that a listener is attending to. In order to examine this, a series of stimuli (chord sequences) were constructed and ordered such that there were the following five experimental conditions:

1. Distant key (two stimuli)
2. Close key (two stimuli)
3. Same key (two stimuli)
4. No key (one stimulus)
5. Initial (two stimuli)

Section 5.2.2 gives details of the setup and protocol of the experiment upon which the analysis was performed, including details of the EEG data preprocessing. Section 5.2.5 gives details about the process of the multivariate signal processing techniques used to extract features from the EEG data for classification. Section 5.2.6 describes the machine learning analysis approaches taken, including conventional SVM analysis as well as a semantic dimension reduction method based on KCCA.

5.2.1 Participants

16 right-handed participants (9 female, 7 male), aged 19 to 31, with normal hearing took part in the experiment. None had received any formal musical education. All participants gave written informed consent to the study, which was approved by the ethics committee of the University of Magdeburg.

5.2.2 Design

The stimuli consist of sequences of chords, with each stimulus in a single key (or no key). All sequences consist of 16 chords with onsets at 500ms intervals and with duration filling the entire 500ms, giving a total length of 8s. The experimental conditions are defined by contiguous stimulus triplets with changes in relative key (listed below). Relative key is established by tonal stimuli, and reset by atonal stimuli. Stimuli from the first three conditions are followed by a stimulus from condition four as a contrast and a reset of relative tonality. 48 stimuli required altogether, all chordal (in root position), of which 32 are tonal and 16 atonal. Tonal stimuli to be transposed as required to fulfill experimental role. First stimulus in each tonal pair is to be in C major, to eliminate any long-term tonality effects (or at least to take advantage of them); second is in either F# major (condition 1), G major (condition 2) or C major (condition 3). In total there were 48 initial, 48 atonal, 16 close and 16 distant trials per participant, giving a total of 144 trials.

Ordering Principles:

1. Each condition should appear an equal number of times
2. Each different melody type (a,b etc.) should appear an equal number of times
3. The three conditions should appear in each permutation (to minimise condition order effects)
4. Each different melody type should be used once for each of the three main conditions (to minimise individual melody effects)
5. Each tonal pair in the conditions should use the same stimulus
6. Each tonal pair should be followed by a unique atonal stimulus to reset tonality (and provide a control condition)
7. Same order for each run and for each subject (for direct comparison in subsequent analysis)

5.2.3 EEG Measurements

EEG recordings were acquired at the Leibniz Institute for Neurobiology (Magdeburg, Germany). 64 unipolar channels, including 2 Electrooculogram (EOG) channels and one nose reference electrode were recorded at a sampling of $500Hz$ and a resolution of $0.1\mu V$. Across all participants the voltage range was $3.2767mV$ and the impedance was less than $5k\Omega$. The music was played to the participants using a Terratec EWX 24/96 soundcard, Black Cube Linear Science amplifier by Lehmann Audio (www.lehmannaudio.de), and Eartone 3A Insert Earphones 50Ω using binaural presentation. The volume of the amplifier was at notch 6. Stimulus delivery and scanning coordination were controlled with $Presentation^{©}$ software (Neurobehavioural Systems Inc, Albany, USA) using a custom-written script.

5.2.4 Data Preprocessing

Muscular activity related to eye movements and eye blinks alter the electromagnetic fields around the eyes and typically introduce artefacts into the EEG, especially in frontal regions. A number of algorithms have been proposed to correct for EOG artefacts, which all correct for EOG artefacts by subtracting a

proportion of one or more EOG channels from the EEG channels. A study by [153] evaluated four correction techniques by correcting blinks, vertical and horizontal eye movements from 26 subjects. The study concluded that in the absence of specific calibration protocols, the method described by [154], based on multiple regression, was the best solution. The approach taken by [155] was based on the algorithm suggested by [154], with modifications described in [156]. This latter method was chosen for the present study.

Prior to time-frequency analysis, the data was filtered using two-way least-squares FIR filtering. Digital filters: $0.2Hz$ low pass filter. $100Hz$ high pass filter. The $50Hz$ component of the signal was removed using a notch filter between $49Hz$ and $51Hz$ due to AC mains signal.

The electrodes were then re-referenced using the nose electrode.

5.2.5 Feature Extraction

The data from the 64 channel EEG system at $500Hz$ sampling rate was imported as a single matrix such that the format was [channels x frames]. The data was segmented into 8 second epochs, giving 144 epochs per subject. These epochs have a one-to-one correspondence with the experimental stimuli. This results in a data matrix of shape [channels x frames x epochs].

Time-Frequency Analysis

The *time average* of a discrete-time random signal is defined as,

$$\langle \cdot \rangle \doteq \lim N \to \infty \frac{1}{2N+1} \sum_{t=-N}^{N} (\cdot). \tag{5.1}$$

We can then describe ensemble averages in terms of this time average, as follows:

$$\begin{aligned} \text{Mean value} \quad & \mu_x = \langle x(t) \rangle \\ \text{Variance} \quad & \sigma_x = \left\langle |x(t) - \mu_x|^2 \right\rangle \\ \text{Autocorrelation} \quad & r_x(l) = \langle x(t) x * (t-l) \rangle \\ \text{PSD} \quad & R_x = \textstyle\sum_{l=-\infty}^{\infty} r_x(l) \end{aligned} \tag{5.2}$$

Until now, the discussed estimation techniques for the computation of spectral properties of signals have all been *univariate* (*i.e.* those given in Section 4.2.3). In many applications we have two or more jointly stationary random processes and we wish to study relationships between them (as is the case for the class of signals in this Chapter). We will use multiple *bivariate* spectral estimations to perform *multivariate* analysis. Assume that $x(t)$ and $y(t)$ are two zero-mean, jointly stationary random processes. The following quantities can then be defined,

$$\begin{aligned} \text{Cross-correlation} \quad & r_{xy}(l) = \langle x(t) y * (t-l) \rangle \\ \text{Cross-PSD} \quad & R_{xy} = \textstyle\sum_{l=-\infty}^{\infty} r_{xy}(l) \exp(-i\omega l) \\ \text{Coherence} \quad & C_{xy} = \frac{|R_{xy}|^2}{R_x R_y} \end{aligned} \tag{5.3}$$

For the analysis of EEG, these bivariate estimations should in principle be more stable that the univariate estimations. Coherence between pairs of EEG signals recorded simultaneously from different scalp sites provides a high time resolution measure of the degree of dynamic connectivity between brain regions. Coherence measures the correlation between a pair of signals as a function of frequency. Thus it provides a means for identifying and isolating frequency bands at which the EEG displays between-channel synchronization (see *e.g.* [157] for a recent review). In addition, electrodes have a tendency to "drift" over time (in terms of both amplitude and mean amplitude), meaning that univariate estimations can become unstable. Bivariate estimation methods overcome this problem as electrodes that are spatially proximate tend to drift in a linearly dependent manner.

A multitaper spectrum is produced by averaging multiple windowed FFTs generated with a set of orthogonal data tapering windows known as Discrete Prolate Spheroidal Sequences (DPSS) or Slepian functions. Since each of the windows in a specific sequence is uncorrelated, an unbiased average spectrum can be produced. A multitaper spectrum offers no greater frequency resolution than a single tapered spectrum. In fact, the spectral peaks resulting from the algorithm have a flat-topped envelope shape which makes the central frequency determination more difficult. What is gained is a reduced-variance spectral estimator that retains a high dynamic range. [158]

Using DPSS, inter-channel coherence, cross phase and cross power were computed, for all pairwise combinations of channels, excluding the EOG electrodes and nose reference electrode. Cross power simply refers to the ratios of the power within each of the frequency bandwidths. The coherence function measures the correlation between two signals as a function of the frequency components they contain, and is therefore a correlation spectrum [159, 160]. It determines the likelihood of two stochastic signals arising from the same generating process.

This differs from the cross-correlation function, which involves calculating Pearson product-moment correlation coefficients for the two signals at various displacements of sampling interval. Quantitative analysis [160] has shown that the cross-correlation sometimes fails in situations where coherence does not, as well as being more expensive to compute. Complementary to the computation of the coherence spectrum is the phase spectrum, which indicates the phase relationship between two signals as a function of frequency - information that is lost using ordinary spectral methods. An important feature of all of these methods is that they are independent of amplitude, as the amplitudes of electrodes are known to vary greatly both within and between recording sessions.

The resulting 256 Fourier coefficients for each of the measures were divided into bands, providing estimates of spectral power within the following recognised frequency bandwidths:

- delta (0.3-3.9Hz)
- theta (4-7.9Hz)
- alpha (8-13Hz)
- beta1 (13-19Hz)
- beta2 (20-30Hz)

In addition,

- low gamma (30-42Hz)
- 40Hz (38-42Hz)
- mid gamma (43-63Hz)
- high gamma (64-100Hz)
- general gamma (30-100Hz)
- global (0.01-100Hz)

bandwidths were computed. The means and variances of each of the measures within each of the wavebands were computed. The data was then flattened in order to create a large feature vector of length 122120 for classification.

5.2.6 Results

SVM Analysis

Recall that we are aiming to predict whether the partcipants were attending to tonal or atonal sequences. The data was standardised across the features to obtain "standard normal" random variables with mean 0 and standard deviation 1. The data for each subject was split randomly into 75% train, 25% test[1] and then concatenated to form the full training and test sets. The same random split was applied for all of the analysis. Classification was performed using the SVM-Light Support Vector Machine implementation [161] with linear, RBF and laplace kernels (where the laplace kernel is the same as the RBF kernel except that the 2-norm is replaced with a 1-norm). 5-fold CV was performed on the training set to discover best setting of the C and sigma parameters. Table 5.1 shows the test errors for the SVM classifier on the split of the data described above. The significance of the classifier was evaluated using the upper bound of the cumulative distribution function (CDF) of the binomial distribution of a random classifier, calculated as follows:

$$p \leq \exp\left(-2\frac{(n\pi - k)^2}{n}\right) \tag{5.4}$$

where n is the number of trials (test examples), π is the probability of success (0.5 for a random classifier) and k is the test error of the classifier.

Test	# Train	# Test	Linear	RBF	Laplace
Tonal vs Atonal	1152	384	0.2298**	**0.1175****	0.2742**
Close vs Distant	384	128	0.3125**	**0.2422****	0.4375
Same vs Distant	384	128	0.2656**	0.2344**	**0.2109****
Same vs Close	384	128	0.2031**	**0.1641****	**0.1641****

Table 5.1: *Test errors for within-subject SVM classification. ** denotes significance at the $p < 0.001$ level (see text)*

Table 5.2 shows the leave-one-out test error for each of the participants using a linear kernel. In this test the data from 15 of the participants is used as the training set and the data from the remaining participant is used as the testing set. This is a much more difficult test, in the sense that the goal is now to

[1]Each trials were treated as a single example, and therefore with 16 participants and 96 trials each training set contained $16 \times 72 = 1152$ examples and each test set contained $16 \times 24 = 384$ examples

learn features that can generalise from one set of brains to a new brain. It is therefore not surprising that with a subject pool of only 16 participants the classification errors are close to chance for most subjects. Results (not given) for the RBF and Laplace kernels were not significantly different. It is interesting to note that the distinction between "close" and "distant" gives the best classification results rather than tonal vs atonal. As such it appears that conditions with key changes result in more consistent prediction across brains than those for processing atonal music.

Subject	Tonal v atonal (96)	Close v distant (32)	Same v distant (32)	Same v close (32)
1	0.4583	0.3438	0.3125	0.3750
2	0.4947	0.4688	0.3438	0.4375
3	0.4688	0.3438	0.3750	0.4062
4	0.4688	0.4375	0.5000	0.4062
5	0.4896	0.4688	0.5000	0.4062
6	0.5000	0.5000	0.4375	0.4688
7	0.4583	0.4688	0.4375	0.4375
8	0.4896	0.3750	0.3438	0.5000
9	0.4896	0.4375	0.5000	0.5000
10	0.4688	0.4062	0.3750	0.4688
11	0.4792	0.3438	0.5000	0.4688
12	0.4792	0.3125	0.4062	0.5000
13	0.4583	0.3750	0.4375	0.5000
14	0.5000	0.3750	0.5000	0.4062
15	0.4688	0.4375	0.3125	0.4688
16	0.5000	0.3125	0.3750	0.5000
mean	0.4795	0.4004	0.4160	0.4531
median	0.4792	0.3906	0.4219	0.4688

Table 5.2: *Test errors for leave-one-out SVM classification using linear kernels. The numbers in parentheses represent the number of test examples. None of the test errors reached significance at the* $p < 0.01$ *level*

KCCA Analysis

Various methods have been proposed for searching for common patterns between two sets of signals, including kernel canonical correlation analysis (KCCA), which can be viewed as a generalised form of kernel independent components analysis [162]. Canonical correlation analysis (CCA) is a technique to extract common features from paired multivariate data. Recall that KCCA is a nonlinear version of this technique which allows nonlinear relations to be found between multivariate variables effectively [52].

$$\rho = \max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{\boldsymbol{\alpha}'\mathbf{K}_x\mathbf{K}_y\boldsymbol{\beta}}{\sqrt{\boldsymbol{\alpha}'\mathbf{K}_x^2\boldsymbol{\alpha}\beta'\mathbf{K}_y^2\boldsymbol{\beta}}} \tag{5.5}$$

For this analysis it was necessary to calculate kernels on the musical stimuli. For simplicity of analysis, the only distinction being examined in this section is tonal vs atonal, as the experimental setup does not lead to a simple calculation of relative pitch for stimuli that were presented following silence. The midi audio files used to generate the experimental stimuli were first embedded into pitch class space. Pitch class space [163] is the circular (quotient) space with the result that differences between octave-related pitches are ignored. In this space, there is no distinction between tones that are separated by

an integral number of octaves. The pitch class vectors for each stimulus were then formed into kernels using a squared exponential kernel. As a sanity check, running an SVM on these gives a test error of 0.0261, showing that this kernel representation is valid. Perfect classification was not achieved as there appear to be outlier stimuli, *i.e.* atonal sequences that appear tonal in this representation.

For the purposes of the KCCA analysis, a linear kernel is used for the EEG, as the dimensionality of the RBF kernel in this case is too high. Both kernels were projected into Gram-Schmidt space using the partial Gram-Schmidt decomposition outlined in [52]. The precision parameter was set to 0.3 using a heuristic method. The use of this decomposition results in an implicit regularisation, and as such the KCCA regularisation parameter was set to zero. Experimentation with different values of this parameter did not show any improvement in results.

The kernels from each view were then projected into the shared feature space using the top 100 resulting KCCA directions. The test kernel for the EEG was also projected into this space, and then normalised such that the ℓ_2-norm of each vector was 1. Using the 100 largest correlation values with the corresponding projections of the training data, the most popular labels of the corresponding example in the music kernel were used as the classification. The reported errors are then the mean of the differences between these labels and the true test labels. This method is an extension of mate-based retrieval [106], was given algorithmically in Algorithm 5 in Chapter 3.

The classification results using the PNN classification approach are given in table 5.3. It can be seen that this method is able to classify between the tonal and atonal experimental conditions almost perfectly. As a comparison, an SVM was trained on the projection of the EEG data into the shared feature space, using a linear kernel and 5-fold CV to select the C parameter. The results show that the PNN method performs competitively with the SVM, whilst being essentially an unsupervised method. It is also much more computationally efficient as there are no parameters to tune.

Classifier	**# Train**	**# Test**	**Linear**
KCCA + PNN	1152	383	0.0183**
KCCA + SVM (linear)	1152	383	0.0157**

Table 5.3: *Test errors for within-subject classification for Tonal vs Atonal using KCCA with PNN and SVM classification. ** denotes significance at the $p < 0.001$ level*

5.2.7 Leave-one-out Analysis

We now present results for leave-one-out analysis of the data. This is the (much more difficult) classification task of taking each participants' data as the test set in turn, using only the data from the remaining participants as the training set. We therefore are given no prior knowledge of the unique physiology of the test participant, nor do we have any knowledge of the specifics of the particular recording (such as the raw electrode amplitudes). This means that the features used for classification must be robust across participants and recording sessions.

Table 5.4 shows the leave-one-out test error for each of the participants using the PNN classification approach, along with the SVM trained on the projection of the EEG data into the shared features space,

again using a linear kernel and 5-fold CV to select the C parameter. The results show that the PNN method performs competitively with the SVM, whilst both significantly outperform the naive SVM approach (see Table 5.2).

Participant	KCCA + PNN	KCCA + SVM (linear)
1	0.2708	0.1667**
2	0.2737	0.2421
3	0.3125	0.2500
4	0.2083*	0.1667**
5	0.4062	0.2500
6	0.2500	0.2500
7	0.5625	0.1667**
8	0.2500	0.2500
9	0.2708	0.2500
10	0.1667**	0.1667**
11	0.7396	0.2500
12	0.2500	0.2500
13	0.1562**	0.1667**
14	0.3542	0.2500
15	0.2500	0.2500
16	0.4688	0.1667**
mean	0.3244	0.2183
median	0.2708	0.2500

Table 5.4: *Test errors for leave-one-subject-out KCCA projected nearest neighbour classification. * and ** denote significance at the $p < 0.01$ and $p < 0.001$ level respectively, using the upper bound of the CDF of the binomial distribution of a random classifier as before*

5.3 Discussion

The results demonstrate that using standard modern Digital Signal Processing (DSP) and Machine Learning (ML) techniques with careful manipulation of the data can enable us to differentiate between certain patterns of brain activity. Coherence analysis and other types of cross-spectral analysis may be used to identify variations which have similar spectral properties (high power in the same spectral frequency bands) if the variability of two distinct time series is interrelated in the spectral domain. The results demonstrate that it is possible to reliably distinguish between whether a listener was attending to tonal or atonal music, including in the case when the test set was a "new brain" (leave-one-out analysis). This can be considered to be a task of high-order cognitive processing, rather than a simple sensory input task. As the differentiation was based on properties of the EEG over relatively long timespans (*i.e.* the length of an epoch, or 8 seconds), this is clearly not due to simple evoked potentials, but instead represents a more fundamental change in the pattern of processing over time.

Further analysis using KCCA demonstrated that through the use of unsupervised methods it is possible to significantly improve the classification accuracy. The new classification method defined in Section 3.5.1 using the shared semantic space given by projections from KCCA weight vectors together with a nearest neighbour method was applied. This was able to distinguish between the tonal and atonal experimental conditions with a high degree of accuracy. It was also shown that an SVM trained on projected

data performed extremely well. The success of both of these methods is due to the KCCA projections acting as a data cleaning step, in which a form of semantic dimensionality reduction is occurring. As the musical stimuli are sufficiently distinct between conditions, the additional information extracts the directions correlated with the differing experimental conditions. The key ingredient in the approach is the introduction of a clean source of data that encodes a complex description of the experience of the subject. It would seem that this approach to information extraction has enormous promise in a wide range of signal processing and time series data analysis tasks.

Subtler discriminations in the task of the listener were also reliably discriminated, such as distinguishing a move from one key to a close or distant key. However the results were not as convincing as for the tonal-atonal distinction. There are several possible reasons for this. Firstly, there were fewer examples of these events by a factor of 3, which on its own increases the difficulty in learning. Secondly, the cognitive task is clearly much more subtle than the tonal vs atonal case, and as such the changes in patterns of activity are likely to be much more subtle, although this is of course speculative. Finally, the type of relationship between the patterns of activity in this case may be too slight to detect, meaning that the DSP techniques employed were unable to detect them (as opposed to the learning algorithm). Further experiments with larger datasets (more repetitions or more participants) could provide the answers to these questions.

EEG data is notoriously noisy and unreliable, so it is extremely encouraging that it is possible to generate reliable discriminations using fully automatic procedures. It is usual to perform artefact rejection by hand during the preprocessing stage, as well as other manual techniques. The present study used automatic techniques at every stage of the process (preprocessing, feature extraction, data treatment, and classification). The methods presented demonstrate the ability to reliably discriminate between brain signals associated with different sequences of music in both within-subject and out-of-subject paradigms.

5.4 Experiment 2: Classification of genre from MEG recordings

Classification of musical genre from audio is a well-researched area of music research. However to our knowledge no studies have been performed that attempt to identify the genre of music a person is listening to from recordings of their brain activity. It is believed that with the appropriate choice of experimental stimuli and analysis procedures, this discrimination is possible. The main goal of this experiment is to see whether it is possible to detect the genre of music that a listener is attending to from brain signals. The present experiment focuses on Magnetoencephalography (MEG), which measures magnetic fields produced by electrical activity in the brain. It will be shown that classification of musical genre from brain signals alone is feasible, but unreliable. Through the use of sparse multiview methods, such as Sparse Multiview Fisher Discriminant Analysis (SMFDA), reliable discriminates between different genres are possible.

The motivation for this study came from the analysis presented in Section 4.2, with the same caveats regarding the task of genre classification applying here as well. As highlighted there and in [11], the

choice of an appropriate dataset was shown to be of great importance. This is interesting from a cognitive perspective, as genre classification may represent both low- and high-order cognitive processes. Using a combination brain recordings and carefully chosen stimuli allows us to analyse this question further.

The analysis procedures employed in this study are based on those used for fMRI using standard GLM and SVM/KCCA methods [164], and methods used for analysis of EEG using KCCA as a semantic dimensionality reduction method prior to classification [14]. The analysis begins with genre classification from the audio source only, as outlined in [11], except that in this study the features used are derived from the midi versions of the audio files rather than raw audio files. The reasons for this are twofold. Firstly, the features of interest are more readily available from the midi, as direct access to the pitch values and note durations of the musical sequences is given. Secondly, the nature of the stimuli means that there is no timbral information available. Most of the features used in previous studies such as [75, 11] are based on short-term spectral information, most of which are strongly picking out timbral features.

Following this, features are derived from the MEG data using spectral methods common to the neuropsychological literature, after which machine learning algorithms are used to classify these features according to genre. Multiview methods are then applied, following on from [164, 14], which attempt to use the stimuli themselves as another view of the phenomenon underlying the brain signals. These methods are improved upon through the use of Sparse Multiview Fisher Discriminant Analysis (SMFDA) [12]. The key difference between this and previous approaches is that SMFDA uses label information to find informative projections of each view into a shared space, which are more appropriate in supervised learning settings. In addition, SMFDA seeks to find sparse solutions by using ℓ_1 optimisation, which is known to approximate the optimally sparse ℓ_0 solution. This is also a form of regularisation that prevents overfitting in high dimensional feature spaces. Sparsity of solutions is important in this setting as the feature set constructed from the MEG data is extremely high dimensional, with a low signal-to-noise ratio.

From Chapter 3 and [13], the optimisation for MFDA is given by,

$$\begin{aligned} \min_{\boldsymbol{\alpha}_d, b, \boldsymbol{\xi}} \quad & \mathcal{L}(\boldsymbol{\xi}) + \mu \mathcal{P}(\tilde{\boldsymbol{\alpha}}), && d = 1, \ldots, p \\ \text{s.t.} \quad & \sum_{d=1}^{p} (\mathbf{K}_d \boldsymbol{\alpha}_d + \mathbf{1} b_d) = \mathbf{y} + \boldsymbol{\xi}, \\ & \boldsymbol{\xi}' \mathbf{e}^c = 0 \ \text{ for } \ c = 1, 2, \end{aligned}$$

The natural choices for the regularisation function $\mathcal{P}(\tilde{\boldsymbol{\alpha}})$ would either be the ℓ_2-norm of the dual weight vectors, *i.e.* $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \sum_{d=1}^{p} \|\boldsymbol{\alpha}_d\|_2^2$, or the ℓ_2-norm of the primal weight vector $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \sum_{d=1}^{p} \boldsymbol{\alpha}_d' \mathbf{K}_d \boldsymbol{\alpha}_d$. However more interesting is the ℓ_1-norm of the dual weight vector, $\mathcal{P}(\tilde{\boldsymbol{\alpha}}) = \sum_{d=1}^{p} \|\boldsymbol{\alpha}_d\|_1$, as this choice leads to sparse solutions due to the fact that the ℓ_1-norm can be seen as an approximation to the ℓ_0-norm. This version is SMFDA.

We can also follow [114] and remove the assumption of a Gaussian noise model, resulting in differ-

ent loss functions on the slacks $\boldsymbol{\xi}$. A noise model with longer tails, such as the Laplacian noise model, may be more appropriate for the class of signals under examination (see [165] for a recent review). In this case we can simply replace $\|\boldsymbol{\xi}\|_2^2$ with $\|\boldsymbol{\xi}\|_1$ in the objective function. The advantage of this is if the ℓ_1-norm regulariser from above is chosen, the resulting optimisation is a linear programme, which can be solved efficiently using methods such as column generation.

The main goal of this experiment is to see whether it is possible to detect the genre of music that a listener is attending to from brain signals. The present experiment uses MEG, which is an imaging technique used to measure the magnetic fields produced by electrical activity in the brain. The data is from an experiment conducted at the Functional Imaging Laboratory (FIL) of UCL.

5.4.1 Participants

MEG recordings from 2 participants are from a 275-channel CTF system with SQUID-based axial gradiometers at a sampling rate of $1200Hz$. Sensors were automatically rejected whose mean power were beyond a static threshold, and trials were rejected in which there was a "sensor jump". The data is filtered using least-squares FIR filters: low pass at $100Hz$; notch filter at 49-$51Hz$. The data is then split into epochs and then downsampled to $200Hz$.

5.4.2 Design

Stimuli were 9 seconds long, with an inter stimulus interval of 2 seconds during which behavioural responses were collected. The behavioural task was identification of genre. Participants were presented four blocks of 20 stimuli.

5.4.3 Procedure

The independent variable was the genre of the musical piece, with 4 levels. Each stimulus was 9 seconds in duration, with an inter-stimulus-interval of 2 seconds within which participants gave their responses for the behavioural task. The behavioural task was identification of genre. Participants were presented four blocks of 20 stimuli, with a break between each block. Blocks were randomized to ensure that practice and fatigue effects are accounted for.

The following genres were included in the experiment: *Classical, Jazz, Ragtime, Pop*. In order to avoid confounding factors of spectral or timbral properties of the pieces within each genre being the main criteria of discrimination, all pieces are based on a single instrument, the piano. The stimuli were sourced and selected as MIDI files from various sources, and then rendered to WAVE format using a single instrument and normalized according to peak amplitude. Most of the excerpts in the *Pop* category were solo piano introductions. The experimental stimuli were validated *a-priori* firstly by classification of genre from the MIDI files using the analysis procedures described by [75, 5] and secondly by examination of the behavioural results.

5.4.4 Feature Extraction

The following two subsections will describe the extraction of features from each of the sources of information. Recall that in addition to the MEG recordings from the participants, we will also be using the stimuli themselves (in the form of the original MIDI audio files) to generate a complementary set of features. These two sources of information will be combined together to build a stronger classifier than would be possible from the MEG alone. For testing purposes we will only use the MEG data (*i.e.* the weights found for the MEG kernel) to show that effect of the addition of information from the stimuli on classification accuracy.

Feature Extraction from Audio

Following [75, 11], the general approach to genre classification taken was to create a large set of features from the audio, and then use a sparse boosting algorithm (LPBoost) which effectively performs feature selection during the classification stage. Since midi files are being used rather than raw audio, it is possible to take advantage of a range of features that are readily derivable from the midi. The features used along with the dimensionality of each feature are given in Table 5.5.

Feature	**Dimensionality**
Meter features	
Tempo	1
Meter	1
Proportion of concurrent onsets	1
Note density	1
Autocorrelation of onset times	33
Melodic features	
Ambitus (melodic range)	1
Tonal features	
Pitch class profiles	12
Distribution of pitch classes (DPC)	12
Krumhansl-Kessler (KK) key estimation	1
Correlation of DPC to KK profiles	24
Mean & standard deviation of KK profiles	2
Statistical features	
Entropy	1
Distribution of note durations	9
of from Number of notes	1
Total	101

Table 5.5: *MIDI features used for genre classification*

For extraction of the features the midi Matlab toolbox of Eerola and Toiviainen was used [166]. These features are then concatenated to produce a single feature vector of length 101.

Feature Extraction from Brain Signals

After preprocessing, the data from each trial were split into 3 segments, representing the first, middle and last 3 seconds of each stimulus presentation. Each of these segments were then used as an example for classification. Dimensionality reduction was then performed using both Principal Components

Analysis (PCA) and Independent Components Analysis (ICA) over the channels, to create two sets of 10 "virtual electrodes". The segments were flattened to form a feature vector of length [20×1800] for each example.

5.4.5 Results

Classification of Genre by Participants

Table 5.6 shows the confusion matrix of the behavioural performance of the subjects. The order of the genres is *classical*, *jazz*, *pop*, *rag*. The true labels are on the rows. Firstly results of the behavioural task of the participants are presented. The overall error is 0.15 (*i.e.* 85% classification success). Note that for 4 classes a random classifier would achieve 0.25, so this is significantly better than chance. This appears to validate the stimuli, and is similar to (or above) levels of accuracy reported elsewhere (see [124] for a review). From the user experiments it can be seen that *pop* appears to be the hardest of the

	classical	jazz	pop	rag	Error
classical	48	1	5	6	0.20
jazz	2	51	4	3	0.15
pop	8	1	49	2	0.18
rag	2	2	1	55	0.08
average					0.15

Table 5.6: *Confusion matrix for classification of genre by participants. True labels are in rows, estimates in columns.*

genres to classify. This makes sense, given that **a)** *pop* as a genre is very derivative, and many themes are borrowed from other genres such as *classical* and *jazz* and **b)** *pop* pieces were chosen that had a solo piano part (*e.g.* as an introduction) meaning that to the uninitiated they may sound uncharacteristic.

Classification of Genre from Audio Features

Using the feature set generated from the midi stimuli, LPBoost [5] was applied using decision stumps as the weak learners as per [75], which results in 6262 weak learners for the algorithm. In order to boost classification performance we split the files into 3 parts, and then took the sum of the classification functions for each of the 3 parts before normalising and classifying. The overall 4-fold cross-validation (CV) error is 0.05 (*i.e.* 95% classification success). This further validates the stimuli, and shows that the methods are appropriate.

Tracing back from the chosen weak learners (of which there were 114/6262), it is possible to see which features were chosen. Interestingly a wide spread of the features were used (52 of the vector of length 101). The only blocks of features not used at all were: *KK key estimation*, *Mean of KK profile*, *Onset autocorrelation*. The key advantage of the LPBoost method is that you can throw as many features as possible at it and it will only pick the useful ones, as it is a sparse method. This means that the same method can be applied to a variety of classification tasks, the algorithm effectively performing feature selection and classification simultaneously.

Figure 5.1 shows a spider diagram of the overall confusion matrix resulting from classification of

genre using audio. This diagram demonstrates that the performance of the classification algorithm is similar across all four genres, with no particular bias towards confusion between any of the genres. The exception is *rag*, for which the performance is generally improved. This can be explained by the fact that the genre is generally more homogeneous, and also less derivative of the other genres. In each of the other genres examples can be found which are in some way similar to one of the other genres.

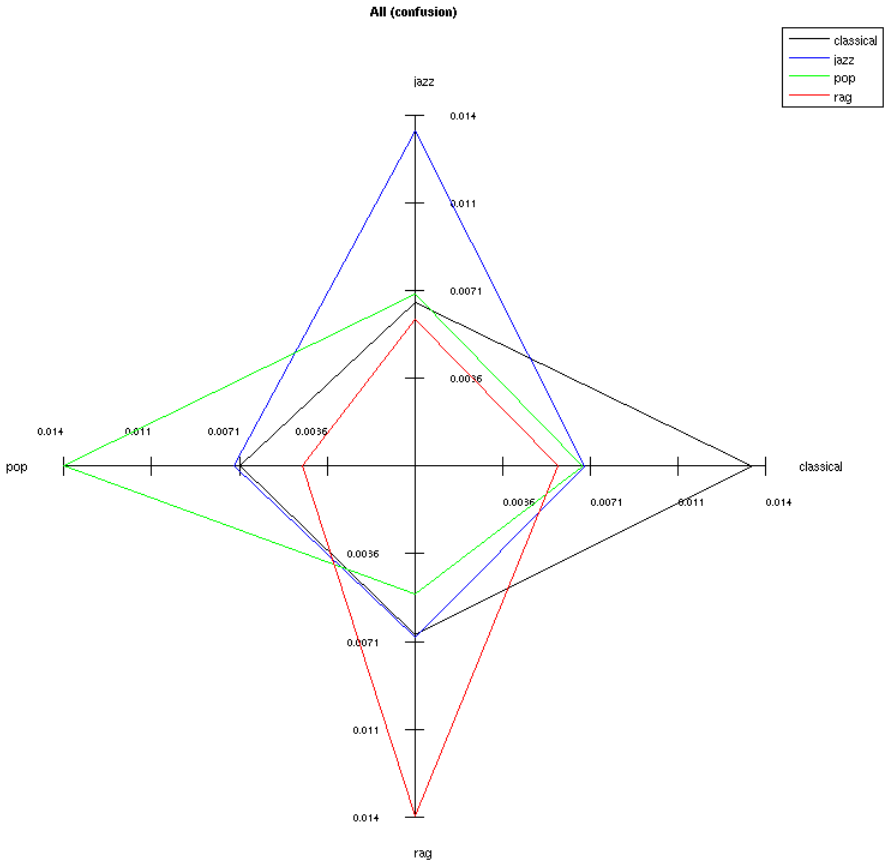


Figure 5.1: *Spider plot of the overall confusion matrix resulting from classification of genre using audio. This is a way of visualising the confusion matrix between classes. The true labels are the axes, and the lines denote the patterns of correct and incorrect classification by class. Note that* rag *(red) has the most "peaked" profile showing that the confusion between this and the other classes was smallest.*

Classification of Genre from MEG Features

Using the feature set generated from the MEG data, linear kernels were constructed used with KFDA [3]. As with the classification of genre from audio features, the files were split into 3 parts, and then the sum of the classification functions for each of the 3 parts were taken before normalising and classifying. The overall 4-fold cross-validation error is 0.71 for participant 1 and 0.70 for participant 2 (*i.e.* 29% and 30% classification success respectively). Note that this is still some way above chance level (25%) but far from reliable.

Classification of Genre using both Data Sources

Using the feature sets generated from the MIDI data and the MEG data, linear kernels were constructed and applied Sparse Multiview Fisher Discriminant Analysis (SMFDA) [13]. 4-fold CV was used for the selection of parameters. Since the sparse version of MFDA is being used, the regularisation parameter can be set using a heuristic method to a small value ($\exp(-3)$) as it has little effect. As with the classification of genre from audio features, the files were split into 3 parts, and then the sum of the classification functions were taken for each of the 3 parts before normalising and classifying. Note that in testing we use only the function learnt on the brain signals. In this way we can be sure that we are not simply classifying on the basis of the MIDI data alone. Furthermore, this is closer to the traditional supervised learning setting where the labels or other significant information regarding correct classification is not known.

The overall 4-fold CV error is 0.65 for participant 1 and 0.63 for participant 2 (*i.e.* 35% and 37% classification success respectively). In itself these classification results are not so impressive, but the side benefit is that the weights of the classifier over the MEG features can be used to then calculate the brain regions involved in classification of musical genre.

5.4.6 Discussion

In this study it was shown that classification of musical genre from brain signals alone is feasible, but unreliable. It was shown that through the use of sparse multiview methods, such as SMFDA, it was possible to improve the discrimination between different genres.

The procedures [164, 12] both incorporate information from the stimuli themselves to improve classification performance. These were extended through the use of Sparse Multiview Fisher Discriminant Analysis (SMFDA) [13]. The key difference is that SMFDA uses label information to find informative projections. It is also important that the method is sparse, as the MEG data is extremely high dimensional.

The key ingredient in the approach of this work is the introduction of a clean source of data that encodes a complex description of the experience of the subject. It seems that this approach has enormous promise in a wide range of signal processing and time series data analysis tasks.

Chapter 6

Conclusions

6.1 Conclusions

6.1.1 Greedy methods

The first part of Chapter 3 focussed on greedy methods for sparse classification and regression, firstly by applying Orthogonal Matching Pursuit (OMP) to KFDA to produce a novel sparse classification algorithm (Matching Pursuit Kernel Fisher Discriminant Analysis (MPKFDA)). Generalisation error bounds were provided that were analogous to that used in the Robust Minimax algorithm [86], together with a sample compression bounding technique. Experimental results on real world datasets were presented, which showed that MPKFDA is competitive with both KFDA and SVM, and additional experiments that showed that MPKFDA performs extremely well in high dimensional settings. In terms of computational complexity the demands of MPKFDA during training are higher, but during the evaluation on test points only k kernel evaluations are required compared to m needed for KFDA.

In a similar vein, the greedy algorithm Polytope Faces Pursuit (PFP) (which is based on the geometry of the polar polytope, where at each step a basis function is chosen by finding the maximal vertex using a path-following method) was applied to nonlinear regression using the *kernel trick*, resulting in KPFP. The utility of this algorithm was demonstrated by providing a novel generalisation error bound which used the natural regression loss and pseudo-dimension in order to upper bound its loss. The experimental results showed that KPFP was competitive against the KMP and KRR.

6.1.2 Low-rank approximation methods

Moving away from greedy methods, the following Section (3.4) constructed algorithms that took advantage of the Nyström method for low-rank kernel approximation for large-scale data. Recent work which empirically justifies using a uniform subsampling technique for the Nyström approximation [100] was theoretically extended. An upper bound on the SVM objective function solved in this subspace was given, followed by empirical validation for both classification and regression using the SVM, KFDA (classification) and KRR (regression) algorithms. The empirical results support the use of uniform

sampling to maintain good learnability in the Nyström subspace. The results show that in the case of MPKFDA it is possible to substantially improve on the complexity of $\mathcal{O}(n^3k)$ to a reduced complexity of $\mathcal{O}(k^3)$, and even improve generalisation performance on some of data sets. This is surprising and counter-intuitive, as MPKFDA selects projection directions that directly optimize the FDA quotient. The main conclusion from the performance of NFDA against MPKFDA and NRR against KMP is that the method by which basis functions are chosen (*i.e.* randomly or according to an objective function) is probably of secondary importance in most cases, unless the goal is for the best possible generalisation error. It seems that the power of these methods are in the projection into the Nyström approximated subspace.

6.1.3 Multiview methods

For the rest of Chapter 3 the attention was turned to the problem of learning from multiple data sources or views (MSL and MVL respectively).

To begin with a method was presented that extends the KCCA algorithm to the classification setting. This method (Projected Nearest Neighbours (PNN)) is an extension of mate-based retrieval [106], and is given in Algorithm 5. It is non-parameteric and essentially *free* once the KCCA directions have been learnt.

KFDA can be formulated as a disciplined convex optimisation problem, which was extended to the multi-view setting MFDA using justifications from a probabilistic point of view. A sparse version SMFDA was then introduced, and the optimisation problem further extended to account for directions unique to each view PMFDA. Experimental validation was shown on a toy dataset, followed by experimental results on part of the PASCAL 2007 VOC challenge dataset and a fMRI dataset, showing that the method is competitive with state-of-the-art methods whilst providing additional benefits.

Mika *et. al.* [35] demonstrate that their convex formulation of KFDA can easily be extended to both multi-class problems and regression problems, simply by updating the final two constraints. The same is also true of MFDA and its derivatives, which enhances its flexibility. The possibility of replacing the Naïve Bayes Fusion method for combining classifiers is another interesting avenue for research.

Finally, for the special case of SMFDA there is the possibility of using a stagewise optimisation procedure similar to the LARS [32] which would have the benefit of computing the full regularisation path, or alternatively greedy methods such as OMP or PFP could be applied to the algorithm. However, as shown theoretically and empirically, a far simpler and yet powerful MVL classification algorithm could be created by combining SMFDA with the Nyström method. This remains as future work.

6.1.4 Experimental applications

Genre classification from polyphonic audio

Many different approaches to genre classification have been taken both in terms of feature selection and in terms algorithm choice. The MIREX 2005 results indicate that boosting with an aggregated

feature set works well. However this really indicates that in a musical sense, the problem is still poorly understood. The short-term spectral features that are commonly used are really only examining different aspects of the texture of the sound, and not really the long-term temporal dynamics. Some attempts to look at temporal dynamics using autocorrelation/autoregression have been attempted, but currently these methods do not perform as well as methods based on short-term spectral features. Clearly some way of combining these two methods appears to be desirable. The experimental results using a replication of the AdaBoost currently have produced seemingly poor results. More work is required to determine the source of the problems causing these results. Experiments with LPBoost are ongoing, but it is expected that improvements will be shown over the existing AdaBoost technique, due to the sparsity of the solutions and the faster convergence of the algorithm.

Compressed sensing for radar

Experimental results have been presented that showed how the ADC sampling rate in a digital radar can be reduced—without reduction in waveform bandwidth—through the use of CS. The use of a Gabor or chirp dictionary and BP allowed reconstruction of the radar backscatter signal in such a way that the range profiles and resulting range-frequency surfaces were still acceptable for conventional use.

The reconstructed data had a worse SNR than the original data. This was attributed to the BP process attempting to reconstruct the noise from the entries in the dictionary. Since these entries are not noise like, the matched filter no longer produced a maximized SNR output. Reconstruction of the samples in low SNR situations is a recognized problem in CS [142, 143, 167, 10]. However, there are other ways to approximate the ℓ_0 solution, such as by greedy iterative methods (Matching Pursuit, Orthogonal Matching Pursuit [56]), and more recently with non-convex penalties and DC programming [62, 63]. Such methods are more robust to noise than BP and it is possible that the presented results can be improved though use of these methods. Investigation of these methods forms the basis of ongoing research by the authors.

One potential problem encountered using this methodology, is that for very fast moving objects there are significant deviations from one fast-time sample to the next. This manifests as a delay and phase shift. As a result, the reconstructions that are generated from a series of such samples are less accurate, because a single set of atoms cannot represent these modulations. One possible way to circumvent this, which could be implemented in hardware, would be to create atoms whose definition include sequences of the atom shifted and translated by some predefined amount. Now each signal is convolved with its corresponding entry (the later signals with the more shifted entries) before performing reconstruction. This would not increase the computation at the learning stage but would increase the size of the potential dictionary.

In conclusion, this work has demonstrated that CS can be applied to conventional pulse-Doppler radars. The reconstructed signals are accurate, and so long as the reduction in received pulses is acceptable, the AIC could be used in radar.

Classification of tonality from EEG recordings

The results demonstrate that using standard modern signal processing and machine learning techniques with careful manipulation of the data can enable us to differentiate between certain patterns of brain activity. Coherence analysis and other types of cross-spectral analysis may be used to identify variations which have similar spectral properties (high power in the same spectral frequency bands) if the variability of two distinct time series is interrelated in the spectral domain. The results demonstrate that it is possibly to reliably distinguish between whether a listener was attending to tonal or atonal music. This can be considered to be a task of high-order cognitive processing, rather than a simple sensory input task. As the differentiation was based on properties of the EEG over relatively long timespans (*i.e.* the length of an epoch, or 8 seconds), this is clearly not due to simple evoked potentials, but instead represents a more fundamental change in the pattern of processing over time.

Further analysis using KCCA demonstrated that through the use of unsupervised methods it is possible to significantly improve the classification accuracy. The new classification method defined in Section 3.5.1 using the shared semantic space given by projections from KCCA weight vectors together with a nearest neighbour method was applied. This was able to distinguish between the tonal and atonal experimental conditions with a high degree of accuracy. It was also shown that an SVM trained on projected data performed extremely well. The success of both of these methods is due to the KCCA projections acting as a data cleaning step, in which a form of semantic dimensionality reduction is occurring. As the musical stimuli are sufficiently distinct between conditions, the additional information extracts the directions correlated with the differing experimental conditions. The key ingredient in the approach is the introduction of a clean source of data that encodes a complex description of the experience of the subject. It would ssem that this approach to information extraction has enormous promise in a wide range of signal processing and time series data analysis tasks.

Subtler discriminations in the task of the listener were also reliably discriminated, such as distinguishing a move from one key to a close or distant key. However the results were not as convincing as for the tonal-atonal distinction. There are several possible reasons for this. Firstly, there were fewer examples of these events by a factor of 3, which on its own increases the difficulty in learning. Secondly, the cognitive task is clearly much more subtle than the tonal vs atonal case, and as such the changes in patterns of activity are likely to be much more subtle, although this is of course speculative. Finally, the type of relationship between the patterns of activity in this case may be qualitatively rather than quantitatively different, meaning that the signal processing techniques employed were unable to detect them (as opposed to the learning algorithm). Further experiments with larger datasets (more repetitions or more participants) could provide the answers to these questions.

EEG data is notoriously noisy and unreliable, so it is extremely encouraging that it is possible to generate reliable discriminations using fully automatic procedures. It is usual to perform artefact rejection by hand during the preprocessing stage, as well as other manual techniques. In this work, automatic techniques were used at every stage of the process (preprocessing, feature extraction, data treatment, and classification). The methods presented demonstrate the ability to reliably discriminate

between brain signals associated with different sequences of music in both within-subject and out-of-subject paradigms.

Classification of genre from MEG recordings

In this study it was shown that classification of musical genre from brain signals alone is feasible, but unreliable. It was shown that through the use of sparse multiview methods, such as SMFDA, it was possible to improve the discrimination between different genres.

The procedures [164, 12] both incorporate information from the stimuli themselves to improve classification performance. These were extended through the use of Sparse Multiview Fisher Discriminant Analysis (SMFDA) [13]. The key difference is that SMFDA uses label information to find informative projections. It is also important that the method is sparse, as the MEG data is extremely high dimensional.

The key ingredient in the approach of this work is the introduction of a clean source of data that encodes a complex description of the experience of the subject. It seems that this approach has enormous promise in a wide range of signal processing and time series data analysis tasks.

6.2 Further Work

6.2.1 Synthesis of greedy/Nyström methods and MVL methods

A very natural extension to the work described in Chapter 3 would be a synthesis of the greedy methods (and/or Nyström methods) with the MVL methods described later in the chapter. Specifically, SMFDA lends itself to this method of optimisation. A feature that makes KFDA and its derivatives an interesting choice in many applications is its strong connection to probabilistic approaches. Often it is not only important to get a small generalisation error but also to be able to assign a confidence to the final classification. Unlike for the SVM, the outputs of KFDA can (under certain assumptions) be directly interpreted as probabilities. A drawback is that the theoretical framework to explain the good performance is somewhat lacking, this very much in contrast to *e.g.* SVMs. Whilst maximising the average margin instead of the smallest margin does not seem to be a big difference most up to date theoretical guarantees are not applicable. Two possible ways to derive generalization error bounds for KFDA based on stability and algorithmic luckiness were described in [168]. Note, however, that through the use of greedy methods such as the described in Section 3.2.1, we were able to produce generalisation error bounds relying on the compression scheme introduced by the Matching Pursuit (MP) algorithm. This provides the possibilty that this theoretical analysis could also be extended to the Multiview setting if we apply the same MP framework. Similarly, it was shown in Section 3.4.1 that by working in the space defined by the Nyström projection, we are still able to learn efficiently, and it should be straightforward to verify that this is still true when performing multiple Nyström projections in multiple views.

6.2.2 Nonlinear Dynamics of Chaotic and Stochastic Systems

There is an emerging field of nonlinear multivariate time series analysis of neuropsychological signals. Multivariate time series analysis is used in neurophysiology with the aim of studying simultaneously recorded signals from different spatial locations. Until recently, the methods have focussed on searching for linear dependencies (*e.g.* cross power spectral density, cross phase, coherence). Recently the theory of nonlinear dynamical systems ("chaos theory") has increasingly been employed to study the pattern formation of complex neuronal networks in the brain [169, 170]. One approach to nonlinear time series analysis consists of reconstructing for time series of EEG or MEG recordings the attractors of the underlying dynamical system. These attractors can be characterised in various different ways (*e.g.* Correlation dimension, Lyapunov exponents), which in turn can act as features for the application of Machine Learning methods.

Here, in the case of the analysis of the brain as a dynamical system, we are interested in nonlinear continuous autonomous conservative systems. At present, it is of no great benefit to analyse this any deeper, as we will be attempting to derive the properties of the dynamical system (the brain) from a temporal series of empirical measurements (EEG data). As such we will not be explicitly creating systems of differential equations or any other such mathematical models. This model free approach requires that extreme care is taken in the interpretation of results, as factors such as experimental noise can introduce dramatic effects.

As mentioned before, we will not be constructing explicit mathematical models of the brain's dynamics. Instead, we will be using empirical time series data from EEG recordings and attempting to reconstruct the dynamics of the system in reverse. There are several steps that need to be taken in order to achieve this. Firstly, the time series data must be embedded into "phase space". There are methods for achieving this, known as temporal and spatial embedding. Once the data has been embedded into phase space, the process of characterising the reconstructed attractors can then occur. While it is outside the present scope to define these techniques formally, an overview will be given below.

The methods that are of interest here are statistical measures, such as Correlation Dimension, Lyapunov Exponents, and Entropy. Each of these methods attempts to characterise the stastical nature of the attractor, such as the exponential rate of divergence of nearby paths on the attractor in the case of Lyapunov Exponents. The first two of these will be described below. The nonlinear entropy measure has been excluded for brevity, but may also prove to be useful.

The correlation integral is the likelihood that two randomly chosen points of the attractor will be closer than r, as a function of r, and is determined by from the distribution of all pairwise distances of points on the attractor. This can be numerically estimated by performing linear regression between $\log(C(r,n))$ and $\log(r)$. If the attractor dimension is finite, then as n increases D_c saturates.

The exponential instability of choatic systems is characterised by a spectrum of Lyapunov Exponents [171]. These are calculated by examining the time evolution of small perturbations of the a trajectory. This then allows the linearisation of the evolution operator. Here is a list, taken from [169], of other nonlinear time series methods, some of which are multivarate, that have been developed recently.

Clearly there are too many methods to go into detail here, and too many to be able to experiment with all of them. Some in particular, such as phase synchronisation in multivariate systems [172, 173, 174] appear to be well suited to the particular nature of the system we are dealing with (*i.e.* EEG measurements). Some analysis of this is given below.

- Nonlinear forecasting
- Local deterministic properties of dynamics
- Determination of optimal probability by Gaussian vs deterministic models
- Cross recurrence
- False nearest neighbours
- 'S' statistic for time irreversibility
- Nonlinear cross prediction
- Unstable periodic orbits
- Phase synchronisation
- Phase synchronisation in multivariate systems
- Cross prediction measure of generalised synchronisation
- 'S' measure of generalised synchronisation
- Synchronisation likelihood
- Mutual dimension (shared DOF of 2 dynamical systems)

It would be an interesting line of research to see if any of these methods are capable of producing stable sets of features that can then be employed for pattern recognition tasks. With the framework outlined in this thesis, it would be a simple case of "plug-and-play" to evaluate various different nonlinear multivariate methods for feature extraction from the brain signals.

6.3 One-class Fisher Discriminant Analysis

The problem of detecting outliers is a classical topic in robust statistics. Recent methods to address this problem include One-Class Support Vector Machines (OC-SVM) [175, 3] and One-Class Kernel Fisher Discriminant Analysis (OCC-FDA) [176], where a kernel induced feature space is used to model non-spherical distributions. A natural extension of the mathematical programming to KFDA of Mika and colleagues [34, 35, 36] would be to the one-class setting, which can be solved using off-the-shelf optimisers. The approach allows the enforcement of sparsity through an ℓ_1-norm constraint on the weight vector. Estimation of the boundary positions could be performed by calculating the quantiles of the posterior probability, which in turn are derived from the conditional class density of the single positive class. This method is simpler to compute and more intuitive than (non-convex) method proposed in [176]. Adjustments to the size of the enclosing hypersphere can then be made using different quantile values adjusted by a single parameter. In fact one could also naturally extend the MFDA described in Section 3.5.2 to this setting, which would result in a novel Multiview One-Class algorithm.

6.4 Summary and Conclusions

This thesis detailed theoretical and empirical work drawing from two main subject areas: Machine Learning (ML) and Digital Signal Processing (DSP). A unified general framework was given for the application of sparse machine learning methods to multivariate signal processing (Chapter 3). In particular, methods that enforce sparsity were employed for reasons of computational efficiency, regularisation, and compressibility. The methods presented can be seen as modular building blocks that can be applied to a variety of applications. Application specific prior knowledge can be used in various ways, resulting in a flexible and powerful set of tools. The motivation for the methods is to be able to learn and generalise from a set of multivariate signals.

In addition to testing on benchmark datasets, a series of empirical evaluations on real world datasets were carried out. These included: the classification of musical genre from polyphonic audio files; a study of how the sampling rate in a digital radar can be reduced through the use of Compressed Sensing (CS); analysis of human perception of different modulations of musical key from Electroencephalography (EEG) recordings; and classification of genre of musical pieces to which a listener is attending from Magnetoencephalography (MEG) brain recordings. These applications demonstrate the efficacy of the framework and highlight interesting directions of future research.

Appendix A

Mathematical Addenda

Sets	
$\mathbb{Z}$	Integers
$\mathbb{R}$	Real numbers
$\mathbb{R}^+$	Positive real numbers
$\mathbb{C}$	Complex numbers
$\|\Delta\|$	Cardinality of set Δ
Spaces	
$\mathcal{H}$	Hilbert space
$\mathcal{F}$	Feature space
$\mathbb{L}_1(\mathbb{R})$	Functions such that $\int \|f(t)\|\, dt < \infty$
$\mathbb{L}_2(\mathbb{R})$	Finite energy functions $\int \|f(t)\|^2\, dt < \infty$
$\ell_1(\mathbb{R})$	Vector space of absolutely convergent series
$\ell_2(\mathbb{R})$	Vector space of square summable sequences
$< f, g >$	Inner product
$\|f\|_1$	ℓ_1 or $\mathbb{L}_1$ norm
$\|f\|_2$	Euclidean or Hilbert space norm
$\|\mathbf{A}\|_F$	Frobenius norm of matrix $\mathbf{A}$
Scalars, vectors, and matrices	
$\mathbf{x} \in \mathbb{R}^n$	Examples
$y \in \{-1, 1\}$	Labels (for classification)
$y \in \mathbb{R}$	Labels (for regression)
$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)'$	Inputs as row vectors
$\mathbf{y}$	Outputs as a vector
$\mathcal{X}$	The space of all possible inputs
$\mathcal{Y}$	The space of all possible outputs
$S \sim \{\mathcal{X} \times \mathcal{Y}\}$	A set input output pairs drawn i.i.d. from a fixed but unknown distribution
$\mathbf{n} \in \mathbb{R}^n$	Vector of i.i.d. random variables with mean 0 and variance σ^2
$\mathbf{A}'$	Transpose of matrix $\mathbf{A}$
$\mathbf{A}^\dagger$	Moore-Penrose pseudo-inverse of matrix $\mathbf{A}$
$\mathbf{\Sigma} = \mathbf{X}'\mathbf{X}$	Covariance matrix
$\mathbf{G} = \mathbf{X}\mathbf{X}'$	Gram matrix
$\mathbf{w}$	Primal weight vector
$\boldsymbol{\alpha}$	Dual weight vector
$\mathbf{e}$	Unit vector
$\mathbf{1}$	Vector of all ones
$\mathbf{I}$	Identity matrix
$\mathbf{K}$	Kernel matrix has entries $\mathbf{K}[i, j] = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$
$\mathbf{K}[:, i]$	ith column of $\mathbf{K}$
$\mathbf{i} = \{i_1, \ldots, i_k\}$	Set of indices
$\mathbf{K}[\mathbf{i}, \mathbf{i}]$	Square matrix defined by index set $\mathbf{i}$
$\boldsymbol{\xi} \in \mathbb{R}^n$	Vector of slack variables
γ	Margin
ϵ	Epsilon (small value)
Functions	
$\phi(\mathbf{x})$	Feature map
κ	Kernel function
$\mathcal{L}$	Loss function
Probability	
$\Pr(x)$	Probability of event x
$\mathbb{E}[x]$	Expected value of x
$\mathcal{R}$	(True) Risk
$\hat{\mathcal{R}}$	Empirical Risk

Table A.1: *Table of commonly used mathematical symbols*

Appendix B

Acronyms

AdaBoost Adaptive Boosting 27
ADC Analogue to Digital Conversion 15
AIC Analogue to Information Conversion 15
AP Average Precision
AR Autoregression 103
ARMA Autoregressive Moving Average 49
BER Balanced Error Rate
BP Basis Pursuit 26
CCA Canonical Correlation Analysis 38
CDF cumulative distribution function 124
CS Compressed Sensing 3
CV cross-validation 26
DC Difference of Convex 43
DELVE Data for Evaluating Learning in Valid Experiments 60
DFT Discrete Fourier Transform 101
DPSS Discrete Prolate Spheroidal Sequences 123
DSP Digital Signal Processing 3
DTW Dynamic Time Warping 11
ECOC Error-Correcting Output Codes 104
EEG Electroencephalography 3
EOG Electrooculogram 121
ERP Event-Related Potential 118
FDA Fisher Discriminant Analysis 27
FDR Fisher Discriminant Ratio 55
FFT Fast Fourier Transform 101
FIL Functional Imaging Laboratory 130
FM Frequency Modulated 110
FMCW Frequency Modulated Continuous Wave 110
fMRI functional Magnetic Resonance Imaging 15
GLM General Linear Model 21
GVSM Generalised Vector Space Model 38

Bibliography

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[3] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, U.K., 2004.

[4] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):1455–1480, 1998.

[5] Ayhan Demiriz, Kristin P. Bennett, and John Shawe-Taylor. Linear programming boosting via column generation. *Journal of Machine Learning Research*, 46(13):225–254, 2002.

[6] Vincent Guigue, Alain Rakotomamonjy, and Stephane Canu. Kernel basis pursuit. In *Proceedings of the 16th European Conference on Machine Learning*, 2005.

[7] Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, volume 13, pages 682–688. MIT Press, 2001.

[8] Alfred M. Bruckstein, David L. Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images, 2007.

[9] E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ell-1 minimization. *Journal of Fourier Analysis and Applications*, 14(5–6):877–905, December 2008.

[10] Sami Kirolos, Jason Laska, Michael Wakin, Marco Duarte, Dror Baron, Tamer Ragheb, Yehia Massoud, and Richard Baraniuk. Analog-to-information conversion via random demodulation. In *Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on*, pages 71–74, Oct. 2006.

[11] T. Diethe and J. Shawe-Taylor. Linear programming boosting for classification of musical genre. Technical report, Presented at the NIPS 2007 workshop Music, Brain & Cognition, 2007.

[12] T. Diethe, S. Durrant, J. Shawe-Taylor, and H. Neubauer. Semantic dimensionality reduction for the classification of EEG according to musical tonality. Technical report, Presented at the NIPS 2008 workshop Learning from Multiple Sources, 2008.

[13] T. Diethe, D.R. Hardoon, and J. Shawe-Taylor. Multiview fisher discriminant analysis. Technical report, Presented at the NIPS 2008 workshop Learning from Multiple Sources, 2008.

[14] T. Diethe, S. Durrant, J. Shawe-Taylor, and H. Neubauer. Detection of changes in patterns of brain activity according to musical tonality. In *Proceedings of IASTED 2009 Artificial Intelligence and Applications*, 2009.

[15] T. Diethe and Z. Hussain. Matching pursuit kernel fisher discriminant analysis. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics 2009 (5)*, pages 121–128, 2009.

[16] T. Diethe, G. Teodoru, N.Furl, and J. Shawe-Taylor. Sparse multiview methods for classification of musical genre from magnetoencephalography recordings. In J. Louhivuori, T. Eerola, S. Saarikallio, T. Himberg, and P-S. Eerola, editors, *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009) Jyväskylä, Finland*, 2009.

[17] T. Diethe and Z. Hussain. Kernel polytope faces pursuit. In *ECML/PKDD (1)*, pages 290–301, 2009.

[18] G.E. Smith, T. Diethe, Z. Hussain, J. Shawe-Taylor, and D.R. Hardoon. Compressed sampling for pulse doppler radar. In *Proceedings of the IEEE International Radar Conference RADAR2010, Washington, USA*, 2010.

[19] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[20] Jerome H. Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data Min. Knowl. Discov.*, 1(1):55–77, 1997.

[21] Colin McDiarmid. On method of bounded differences. *Surveys in Combinatorics*, 141:148–188, 1989.

[22] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[23] H. Chernoff. A measure of asymptotic efficiency of tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[24] V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, Moscow, 1974. (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie-Verlag, Berlin, 1979).

[25] V. Vapnik and A. Chervonenkis. Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability and its Applications*, 26(3):532–553, 1981.

[26] V. Vapnik. Inductive principles of statistics and learning theory. In P. Smolensky, M. C. Mozer, and D. E. Rumelhart, editors, *Mathematical Perspectives on Neural Networks*. Lawrence Erlbaum, Mahwah, NJ, 1995.

[27] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems*. W. H. Winston, Washington, D.C., 1977.

[28] R. L. Plackett. Some theorems in least squares. *Biometrika*, 37(1-2):149–157, 1950.

[29] David L. Donoho. For most large underdetermined systems of linear equations the minimal $\ell 1$-norm solution is also the sparsest solution. *Comm. Pure Appl. Math*, 59:797–829, 2004.

[30] R. Tibshirani. Regression selection and shrinkage via the lasso. Technical report, Department of Statistics, University of Toronto, June 1994. ftp://utstat.toronto.edu/pub/tibs/lasso.ps.

[31] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[32] Bradley Efron, Trevor Hastie, Lain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2002.

[33] Michael R. Osborne, Brett Presnell, and Berwin A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9:319–337, 1999.

[34] Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, and Klaus-Robert Müller. Fisher Discriminant Analysis with kernels. In E. Wilson Y. H. Hu, J. Larsen and S. Douglas, editors, *Proc. NNSP'99*, pages 41–48. IEEE, 1999.

[35] S. Mika, G. Rätsch, and K.-R. Müller. A mathematical programming approach to the kernel Fisher algorithm. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 591–597, 2001.

[36] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K-R. Müller. Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):623–628, 2003.

[37] Alex J. Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of 17th International Conference on Machine Learning*, pages 911–918. Morgan Kaufmann, San Francisco, CA, 2000.

[38] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[39] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

[40] Yoav Freund. Boosting a weak learning algorithm by majority. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1990.

[41] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, page 148156, 1996.

[42] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

[43] T. Zhang. Convex risk minimization. *Annals of Statistics*, 2004.

[44] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, 1963.

[45] G. Rätsch, T. Onoda, and K.-R. Mller. Soft margins for adaboost. *Machine Learning*, 42(3):287–320, 2001.

[46] Saharon Rosset Watson, Ji Zhu, and Trevor Hastie. Margin maximizing loss functions. In *In Advances in Neural Information Processing Systems (NIPS) 15*, page 16. MIT Press, 2003.

[47] Gunnar Rätsch and Manfred K. Warmuth. Efficient margin maximizing with boosting. *J. Mach. Learn. Res.*, 6:2131–2152, 2005.

[48] Manfred K. Warmuth, Karen A. Glocer, and Gunnar Rätsch. Boosting algorithms for maximizing the soft margin. In *NIPS*, 2007.

[49] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Technical Report 44, Max-Planck-Institut für biologische Kybernetik, 1996.

[50] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:312–377, 1936.

[51] A. Vinokourov, J. Shawe-Taylor, and N. Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances of Neural Information Processing Systems 15*, 2002.

[52] D.R. Hardoon, S.Szedmak, and J.Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[53] David R. Hardoon and John Shawe-Taylor. Sparse canonical correlation analysis. Technical report, University College London, UK, 2007.

[54] L. Elden and L. Wittmeyer-Koch. *Numerical Analysis: An Introduction*. Academic Press, Cambrigde, 1990.

[55] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69:331–371, 1910.

[56] Stéphane Mallat and Zhifeng Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[57] R.G. Baraniuk and D.L. Jones. Shear madness: New orthonormal bases and frames using chirp functions. *IEEE Trans. Signal Processing: Special Issue on Wavelets in Signal Processing*, 41:3543–3548, 1993.

[58] Ingrid Daubechies. Time-frequency localization operators: A geometric phase space approach. *IEEE Transactions on Information Theory*, 34(4):605–612, 1988.

[59] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.

[60] D.L. Donoho. Neighborly polytopes and sparse solution of underdetermined linear equations. Technical report, Department of Statistics, Stanford Univ., Stanford, CA, 2005.

[61] M. D. Plumbley. Recovery of sparse representations by polytope faces pursuit. In *ICA*, pages 206–213, 2006.

[62] R. Horst and N. V. Thoai. Dc programming: overview. *J. Optim. Theory Appl.*, 103(1):1–43, 1999.

[63] Gilles Gasso, Alain Rakotomamonjy, and Stéphane Canu. Recovering sparse signals with a certain family of non-convex penalties and dc programming. *IEEE Transactions on Signal Processing*, in press.

[64] Yagyensh Chandra Pati, Ramin Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, pages 40–45, 1993.

[65] Pascal Vincent and Yoshua Bengio. Kernel matching pursuit. *Machine Learning*, 48(1-3):165–187, 2002.

[66] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*. The Johns Hopkins University Press, 3rd edition, October 1996.

[67] Mark D. Plumbley. Polar polytopes and recovery of sparse representations, 2005.

[68] S. Chen. *Basis Pursuit*. PhD thesis, Department of Statistics, Stanford University, November 1995.

[69] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[70] E. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, Dec. 2006.

[71] David L. Donoho and Michael Elad. On the stability of the basis pursuit in the presence of noise. *Signal Process.*, 86(3):511–532, 2006.

[72] Aapo Hyvärinen. Independent component analysis in the presence of gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22:49–67, 1998.

[73] M. Davy, C. Doncarli, and J.-Y. Tourneret. Supervised classification using mcmc methods. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 1:33–36, 2000.

[74] George Edward Pelham Box and Gwilym Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[75] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and K. Balázs. Aggregate features and ADABOOST for music classification. *Machine Learning*, 65 (2-3):473–484, 2006.

[76] Stephen Becker, Jerome Bobin, and Emmanuel Candes. NESTA: A fast and accurate first-order method for sparse recovery. Apr 2009.

[77] Peng Zhao and Bin Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006.

[78] Francis R. Bach. Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008.

[79] Cédric Archambeau and Francis Bach. Sparse probabilistic projections. In *NIPS*, pages 73–80, 2008.

[80] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.

[81] Theodoros Evgeniou, Charles A. Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[82] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *NIPS*, pages 41–48, 2006.

[83] Jaisiel Madrid-Sánchez, Miguel Lázaro-Gredilla, and Aníbal R. Figueiras-Vidal. A single layer perceptron approach to selective multi-task learning. In *IWINAC '07: Proceedings of the 2nd international work-conference on The Interplay Between Natural and Artificial Computation, Part I*, pages 272–281, Berlin, Heidelberg, 2007. Springer-Verlag.

[84] Manos Papagelis, Dimitris Plexousakis, and Themistoklis Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In P. Herrmann et al., editor, *iTrust*, volume 3477 of *LNCS*, pages 224–239. Springer-Verlag Berlin Heidelberg, 2005.

[85] John Shawe-Taylor and Nello Cristianini. Estimating the moments of a random vector. In *Proceedings of GRETSI 2003 Conference*, volume 1, page 4752, 2003.

[86] Gert R.G. Lanckriet, Laurent El Ghaoui, Chiranjib Bhattacharyya, and Michael I. Jordan. A robust minimax approach to classification. *J. Mach. Learn. Res.*, 3:555–582, 2003.

[87] Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. Technical report, University of California Santa Cruz, 1986.

[88] Zakria Hussain. *Sparsity in Machine Learning: Theory and Practice*. PhD thesis, University College London, 2008.

[89] Lelsie G. Valiant. A theory of the learnable. *Communications of the Association of Computing Machinery*, 27(11):1134–1142, November 1984.

[90] Peter L. Bartlett and Ambuj Tewari. Sparseness vs estimating conditional probabilities: Some asymptotic results. *Journal of Machine Learning Research*, 8:775–790, 2007.

[91] Y. Saad. Projection and deflation methods for partial pole assignment in linear state feedback. *IEEE Trans. Automat. Contr.*, 33:290–297, 1988.

[92] Isabelle Guyon, Asa Ben Hur, Steve Gunn, and Gideon Dror. Result analysis of the NIPS 2003 feature selection challenge. In *Advances in Neural Information Processing Systems 17*, pages 545–552. MIT Press, 2004.

[93] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal Processing, special issue "Sparse approximations in signal and image processing"*, 86:572–588, 2006.

[94] D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, New York, 1984.

[95] Tong Zhang and L. Bartlett. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.

[96] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[97] J. Shawe-Taylor, M. Anthony, and N. L. Biggs. Bounding sample size with the Vapnik-Chervonenkis dimension. *Discrete Applied Mathematics*, 42(1):65–73, 1993.

[98] M. Ouimet and Y. Bengio. Greedy spectral embedding. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 253–260, 2005.

[99] C. Baker. *The numerical treatment of integral equations*. Oxford Clarendon Press, 1977.

[100] S. Kumar, M. Mohri, and A. Talwalkar. Sampling techniques for the nyström method. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, AISTATS, volume 5 of JMLR: W &CP 5*, 2009.

[101] A. Blum. Random projections, margins, kernels, and feature-selection. *LNCS*, 3940:52–68, 2006.

[102] M.-F. Balcan, A. Blum, and S. Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, 65:79–94, 2006.

[103] Kai Zhang, Ivor W. Tsang, and James T. Kwok. Improved nyström low-rank approximation and error analysis. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1232–1239, New York, NY, USA, 2008. ACM.

[104] Petros Drineas and Michael W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[105] Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 6, New York, NY, USA, 2004. ACM.

[106] D.R. Hardoon and J. Shawe-Taylor. KCCA for different level precision in content-based image retrieval. In *Proceedings of Third International Workshop on Content-Based Multimedia Indexing*, 2003.

[107] J.D.R. Farquhar, D.R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak. Two view learning: SVM-2k, theory and practice. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 355–362. MIT Press, Cambridge, MA, 2006.

[108] Seung-Jean Kim, Alessandro Magnani, and Stephen Boyd. Optimal kernel selection in kernel fisher discriminant analysis. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 465–472, New York, NY, USA, 2006. ACM.

[109] C. M. Christoudias, R. Urtasun, and T. Darrell. Multi-view learning in the presence of view disagreement. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008.

[110] Tonatiuh Peña Centeno and Neil D. Lawrence. Optimising kernel parameters and regularisation coefficients for non-linear discriminant analysis. *Journal of Machine Learning Research*, 7:455–491, 2006.

[111] Mark Girolami and Simon Rogers. Hierarchic bayesian models for kernel learning. In *ICML*, pages 241–248, 2005.

[112] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[113] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998.

[114] S. Mika, A. J. Smola, and B. Schölkopf. An improved training algorithm for kernel Fisher discriminants. In *In AISTATS*, pages 98–104, 2001.

[115] Gayle Leen and Colin Fyfe. Learning shared and separate features of two related data sets using GPLVMs. Technical report, Presented at the NIPS 2008 workshop Learning from Multiple Sources, 2008.

[116] V. Viitaniemi and J. Laaksonen. Techniques for image classification, object detection and object segmentation applied to VOC challenge 2007. Technical report, Department of Information and Computer Science, Helsinki University of Technology (TKK), 2008.

[117] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer vision*, pages 1150–1157, Kerkyra Greece, 1999.

[118] Janaina Mourão-Miranda, Emanuelle Reynaud, Francis McGlone, Gemma Calvert, and Michael Brammer. The impact of temporal compression and space selection on svm analysis of single-subject and multi-subject fmri data. *NeuroImage*, 33:4:1055–1065, 2006.

[119] Juan Jos Burred and Alexander Lerch. A hierarchical approach to automatic musical genre classification. In *Proceedings of the 6th International Conference on Digital Audio Effects*, September 2003.

[120] Tao Li and M. Ogihara. Music genre classification with taxonomy. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[121] Stefan Brecheisen, Hans-Peter Kriegel, Peter Kunath, and Alexey Pryakhin. Hierarchical genre classification for large music collections. In *IEEE 7th International Conference on Multimedia & Expo*, 2006.

[122] B. Whitman, G. Flake, and S. Lawrence. Artist detection in music with minnowmatch. In *Proceedings of the 2001 IEEE Workshop on Neural Networks for Signal Processing*, pages 559–568, 2001.

[123] D. Perrot and R. R. Gjerdigen. Scanning the dial: an exploration of factors in the identification of musical style. In *Proceedings of the 1999 Society for Music Perception and Cognition*, 1999.

[124] A. Meng. *Temporal Feature Integration for Music Organisation*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2006. Supervised by Jan Larsen and Lars Kai Hansen, IMM.

[125] P. Ahrendt and A. Meng. Music genre classification using the multivariate AR feature integration model, aug 2005. Extended Abstract.

[126] A. Flexer. Statistical evaluation of music information retrieval experiments. *Journal of New Music Research*, 35(2):113–120, 2006.

[127] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[128] J. Junqua and J. Haton. *Robustness in Automatic Speech Recognition*. Boston: Kluwer Academic, 1996.

[129] B. Kedem. Spectral analysis and discrimination by zero-crossings. In *Proceedings of the IEEE*, volume 74(11), pages 1477–1493, 1986.

[130] H. Monson. *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, 1996.

[131] Jurij Leskovec and John Shawe-Taylor. Linear programming boosting for uneven datasets. In *International Conference on Machine Learning*, 2003.

[132] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[133] K. Crammer and Y. Singer. On the algorithmic implementation of multi-class SVMs. *Journal of Machine Learning Research*, 2:265–292, 2001.

[134] Z. Wang and J. Shawe-Taylor. Large-margin structured prediction via linear programming. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 599–606, Clearwater Beach, Florida, USA, 2009.

[135] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Popular, classical and jazz music databases. In *Proceedings of the International Conference on Music Information Retrieval*, page 2878, 2002.

[136] A. Meng and J. Shawe-Taylor. An investigation of feature models for music genre classification using the support vector classifier. In *International Conference on Music Information Retrieval*, pages 604–609, sep 2005. Final version : 6 pages instead of original 8 due to poster presentation.

[137] P. Ahrendt, C. Goutte, and J. Larsen. Co-occurrence models in music genre classification. In V. Calhoun, T. Adali, J. Larsen, D. Miller, and S. Douglas, editors, *IEEE International workshop on Machine Learning for Signal Processing*, pages 247–252, Mystic, Connecticut, USA, sep 2005.

[138] Lei Zhang, Mengdao Xing, Cheng-Wei Qiu, Jun Li, and Zheng Bao. Achieving higher resolution ISAR imaging with limited pulses via compressed sampling. *Geoscience and Remote Sensing Letters, IEEE*, 6(3):567–571, July 2009.

[139] R. Baraniuk and P. Steeghs. Compressive radar imaging. In *Radar Conference, 2007 IEEE*, pages 128–133, April 2007.

[140] S. Bhattacharya, T. Blumensath, B. Mulgrew, and M. Davies. Synthetic aperture radar raw data encoding using compressed sensing. In *Radar Conference, 2008. RADAR '08. IEEE*, pages 1–5, May 2008.

[141] S. Bhattacharya, T. Blumensath, B. Mulgrew, and M.Davies. Fast encoding of synthetic aperture radar raw data using compressed sensing. In *Statistical Signal Processing, 2007. SSP '07. IEEE/SP 14th Workshop on*, pages 448–452, Aug. 2007.

[142] M.A. Herman and T. Strohmer. High-resolution radar via compressed sensing. *Signal Processing, IEEE Transactions on*, 57(6):2275–2284, June 2009.

[143] Guangming Shi, Jie Lin, Xuyang Chen, Fei Qi, Danhua Liu, and L. Zhang. UWB echo signal detection with ultra-low rate sampling based on compressed sensing. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 55(4):379–383, April 2008.

[144] Yeo-Sun Yoon and M.G. Amin. Imaging of behind the wall targets using wideband beamforming with compressive sensing. In *Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on*, pages 93–96, 31 2009-Sept. 3 2009.

[145] T.E. Derham, S. Doughty, K. Woodbridge, and C.J. Baker. Design and evaluation of a low-cost multistatic netted radar system. *Radar, Sonar & Navigation, IET*, 1(5):362–368, 2007.

[146] Hiroaki Sakoe. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26:43–49, 1978.

[147] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[148] J.V. Bach M. Barber C. Brigell M. Marmor M.F. Tormene A.P. Holder G.E. Vaegan A. Odom. Visual evoked potentials standard (2004). *Documenta Ophthalmologica*, 108(2):115–123, 2004. cited By (since 1996) 126.

[149] P. Heil. Representation of sound onsets in the auditory system. *Audiology and Neurootology*, 6:167–172, 2001.

[150] J. Mourao-Miranda, A.L.W. Bokde, C. Born, H. Hampel, and M. Stetter. Classifying brain states and determining the discriminating activation patterns: Support vector machine on functional mri data. *NeuroImage*, 4:980–995, 2005.

[151] S. Durrant, D.R. Hardoon, A. Brechmann, J. Shawe-Taylor, E. Miranda, and H. Scheich. GLM and SVM analyses of neural response to tonal and atonal stimuli: New techniques and a comparison. *Special Issue on Music, Brain and Cognition*, 2009.

[152] P. Janata, J.L. Birk, J.D. Van Horn, M. Leman, B. Tillmann, and J.J. Bharucha. The cortical topography of tonal structures underlying western music. *Science*, 298:2167–2170, 2002.

[153] R.J. Croft, J.S.Chandler, R.J.Barry, and N.R.Cooper. EOG correction: A comparison of four methods. *Psychophysiology*, 42:16–24, 2005.

[154] G. Gratton, M. G. H. Coles, and E. Donchin. A new method for off-line removal of ocular artifact. *Electroencephaolography and Clinical Neurophysiology*, 55:468–484, 1983.

[155] C.W. Pleydell-Pearce, S.E. Whitecross, and B.T. Dickson. Multivariate analysis of EEG: Predicting cognition on the basis of frequency decomposition, inter-electrode correlation, coherence, cross phase and cross power. In *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS03)*, 2003.

[156] M.A. Conway, C.W. Pleydell-Pearce, and S.E. Whitecross. The neuroanatomy of autobiographical memory: A slow cortical potential study of autobiographical memory retrieval. *Journal of Memory and Language*, 45:493524, 2001.

[157] Daniel Ruchkin. Eeg coherence. *International Journal of Psychophysiology*, 57(2):83 – 85, 2005. EEG Coherence.

[158] D.B. Percival and A.T. Walden. *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*. Cambridge: Cambridge University Press, 1993.

[159] J.C. Shaw. An introduction to the coherence function and its use in EEG signal analysis. *Journal of Medical Engineering Technology*, 5:279–288, 1981.

[160] J.C. Shaw. Correlation and coherence analysis of the EEG: a selective tutorial review. *International Journal of Psychophysiology*, 1:255–266, 1984.

[161] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA, 1999.

[162] F. Bach and M. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[163] D. Deutsch and J. Feroe. The internal representation of pitch sequences in tonal music. *Psychological Review*, 88:503–522, 1981.

[164] Simon Durrant, David R. Hardoon, Eduardo R. Miranda, John Shawe-Taylor, and André Brechmann. Neural correlates of tonality in music. Technical report, Presented at the NIPS 2007 workshop Music, Brain & Cognition, 2007.

[165] Rami Mangoubi, Mukund Desai, and Paul Sammak. Non-gaussian methods in biomedical imaging. In *AIPR '08: Proceedings of the 2008 37th IEEE Applied Imagery Pattern Recognition Workshop*, pages 1–6, Washington, DC, USA, 2008. IEEE Computer Society.

[166] T. Eerola and P. Toiviainen. Midi toolbox: Matlab tools for music research. Jyväskylän yliopisto, ISBN 951-39-1796-7. URL: http://www.jyu.fi/musica/miditoolbox/, 2004.

[167] E. Candès and M. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE,* 25(2):21–30, March 2008.

[168] S. Mika. *Kernel Fisher Discriminants*. PhD thesis, Technische Universität Berlin, 2002.

[169] C.J. Stam. Nonlinear dynamical analysis of EEG and MEG: Review of an emerging field. *Clinical Neurophysiology*, 116:2266–2301, 2005.

[170] Ernesto Pereda, Rodrigo Quian Quiroga, and Joydeep Bhattacharya. Nonlinear multivariate analysis of neurophsiological signals. *Progress in Neurobiology*, 77:1–37, 2005.

[171] J.-P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, 57:617, 1985.

[172] Joydeep Bhattacharya, Ernesto Pereda, and Hellmuth Petsche. Effective detection of coupling in short and noisy bivariate data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33(1):85–95, 2003.

[173] Joydeep Bhattacharya and Hellmuth Petsche. Phase synchrony analysis of eeg during music perception reveals changes in functional connectivity due to musical expertise. *Signal Process.*, 85(11):2161–2177, 2005.

[174] M.G. Rosenblum, A.S. Pikovsky, and Kurths J. Phase synchronization of chaotic oscillators. *Phys Rev Lett*, 76:1804–7, 1996.

[175] Larry M. Manevitz and Malik Yousef. One-class svms for document classification. *J. Mach. Learn. Res.*, 2:139–154, 2002.

[176] Volker Roth. Kernel fisher discriminants for outlier detection. *Neural Comput.*, 18(4):942–960, 2006.