Pattern recognition and machine learning for magnetic resonance images with kernel methods

by

Chia-Yueh Carlton CHU

Wellcome Trust Centre for Neuroimaging

Institute of Neurology

A thesis submitted for the degree of Doctor of Philosophy

University College London

May, 2009

Primary supervisor: Dr. John Ashburner

Secondary supervisor: Professor Karl Friston

Declaration

I, Chia-Yueh Carlton CHU, 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.

“Prediction is very difficult, especially about the future.”

(Danish Physicist,Nobel Prize for Physics in 1922)

Abstract

The aim of this thesis is to apply a particular category of machine learning and pattern recognition algorithms, namely the kernel methods, to both functional and anatomical magnetic resonance images (MRI). This work specifically focused on supervised learning methods. Both methodological and practical aspects are described in this thesis.

Kernel methods have the computational advantage for high dimensional data, therefore they are idea for imaging data. The procedures can be broadly divided into two components: the construction of the kernels and the actual kernel algorithms themselves. Pre-processed functional or anatomical images can be computed into a linear kernel or a non-linear kernel. We introduce both kernel regression and kernel classification algorithms in two main categories: probabilistic methods and non-probabilistic methods. For practical applications, kernel classification methods were applied to decode the cognitive or sensory states of the subject from the fMRI signal and were also applied to discriminate patients with neurological diseases from normal people using anatomical MRI. Kernel regression methods were used to predict the regressors in the design of fMRI experiments, and clinical ratings from the anatomical scans.

Acknowledgements

I am greatly thankful to both of my supervisors, John Ashburner and Karl Friston, for educating and mentoring me. I learnt immense amount of things from both of you, especially the British humour. John is one of the most important people in my life, he is also one of the most generous, pure, and virtuous people I have ever known. In fact, it is very difficult to describe how John motivated and inspired me in words. But I am sure, I will always remember that John taught me something others could not teach me.

I also thank my parents for their financial support, so I did not die in hunger and homeless. I thank my girl friend Han-Hui Crystal Tsai countless help, and take care of me when I was writing thesis in the states.

I have to thank Geoffrey, for all the spiritual talk we had, for revising my writings, for many other small things. You are one of my best mates, and I am sure we will keep good relationship always.

I have to all my collaborators Stefan Kloopel, Cynthia Stonnington, Demis Hassabis, Cynthia Fu, Janaina Mourau-Miranda.

I have to thank my lab mates, friends and fellows in the 2nd floor back room, especially the French guys. I really love listening to French.

Also, I have thank Richard Frackowiak for leading the Wednesday meeting and various activities.

This work was funded by the Overseas Research Scholarship and University College London, as well as my lovely parents.

Finally, I thank Dr. Lee at Society for Spiritual Calligraphy and Painting for delivering the oracle, which eventually guided me to London.

Table of Contents

Figures

Chapter 1.

Introduction

Contents

1.1 Motivation and Aims

The initial objective of my PhD was to develop robust machine learning systems, which are capable of classifying anatomical brain scans into different disease (or other) categories, using state of the art supervised learning techniques. The aim was to use kernel methods to represent patterns of similarity among brains, the basic idea being that similar brains are more likely to be in the same group. The majority of this work involved collaborations with neurologists and neuroscientists. The Pittsburgh brain activity interpretation competition (PBAIC) in both 2006 and 2007 was an ideal opportunity to compare my machine learning strategies with those of others, and broadened my research interests into the field of "brain decoding" for functional imaging.

The thesis is written for both methodological and general readers. For those who understand neuroimaging methodology, it should contain sufficient mathematical detail to replicate our results. For those who are less interested in technical detail, it also contains intuitive explanations of the algorithms and procedures.

1.1.1 Diagnoses of Neurodegenerative Diseases

The idea of Evidence-based medicine (EBM) was introduced in the early 1990s (1992; Sackett, 1997), before the prevalence of the Internet and Google. The main objective of EBM was to promote the practice of searching published work, and making effective diagnoses and decisions based on the latest evidence. Generally speaking, the framework proposed in my PhD can be interpreted as a form of EBM, by constructing models from existing data, which can make predictions about new cases. Peer reviewed publications only report highly simplified characterisations of data. Much of the relevant information has to be discarded in order to present a few salient results on the printed page. Also, as the number of medical publications grows, new

and more efficient strategies will be needed for encoding medical knowledge. In terms of making diagnoses and clinical decisions, a pattern recognition procedure, optimally trained using relevant data, may eventually prove to be more useful than the entire collection of publications written about those same data.

Neurological diseases and psychiatric disorders are associated with anatomical and functional changes in the brain. For example, Alzheimer's disease involves grey matter loss in the temporal lobe. There is currently great interest in finding markers that may guide the early diagnosis of neurodegenerative disorders, based on anatomical and functional MR images. Such research is sometimes impoverished by a lack of the necessary engineering and statistical expertise. As a result, the end product of such work is often a simple table of manually derived average measurements, along with their standard deviations, and perhaps a few p values relating to group differences. A much more useful solution would be to model the data using state of the art pattern recognition and machine learning techniques. The basic idea of my project was to develop classification systems that can be trained with existing images of known labels (disease states or clinical outcomes). The simplest case involves images from a group of patients and a group of controls, whereby the algorithm would learn the pattern in the images that differentiates between the groups. Then, when a new subject's image is presented to the trained algorithm, it should be able to determine how likely it is that the subject is a control or a patient.

Voxel-Based Morphometry (VBM) (Ashburner and Friston, 2000), is often used to make voxel-wise comparisons of regional volumes of grey matter, among populations of subjects. This could be considered as one way of identifying markers of neurodegenerative disorders. Other methods involve analyzing shape representations of anatomical structures, such as hippocampus. Although those techniques can characterize local differences between patients and controls, they were not designed to

classify new subjects and perform diagnosis. This project will try to parameterize and quantify all these differences observed among subjects, and put them into a machine learning framework. The Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000) is one of the most popular supervised learning algorithms, and is employed in various fields with promising results and reasonable computing time. It's potential had already been demonstrated by identifying subjects' genders from their structural MRI scans (Lao et al., 2004). Besides SVM, there are many other related kernel algorithms. The PhD research focused on implementing some of those algorithms, using simple measurements of brain similarity. There was a specific focus on differentiating Huntington's Disease (HD) and Alzheimer's disease (AD) from healthy controls. Patients who will manifest HD can easily be diagnosed from genetic information, so HD patients can be used for testing models for classifying neurodegenerative diseases. Unlike HD, there are no biomarkers that guarantee 100% accurate diagnosis of AD, other than post mortem examinations. Therefore, there is growing interest in early detection of AD. To accelerate scientific advances in improving the detection of AD from imaging modalities, the "Alzheimer's Disease Neuroimaging Initiative" (ADNI) Database was launched (Mueller et al., 2005). This database allows people to access AD images freely, and apply their algorithms to the dataset. Although the neuroimaging field is a long way behind the geneticists in terms of sharing primary data, a number of other publically available neuroimaging datasets are also beginning to emerge.

During the period of my PhD, several others have also shown interest in applying pattern recognition methods to brain images, for the purpose of making diagnoses. These have involved both structural and functional MRI (Davatzikos et al., 2008; Demirci et al., 2008; Fan et al., 2007a; Fan et al., 2008b; Fan et al., 2005; Fu et al., 2008; Vemuri et al., 2008). Most of the works showed promising results, which may

indicate that this research area has a certain importance to the field. With the increase of data sharing, and computational power continuing to grow exponentially (multi-core processors, cloud computing, etc), automatic diagnostic/screening tools will become applicable in clinical environments. When large training datasets become available, pattern recognition methods will probably become as robust as experienced clinicians. With aging populations, and hints that effective treatments may soon emerge, new developments in the computer aided diagnosis (CAD) of neurodegenerative disease are set to become increasingly important for clinical decision making.

1.1.2 Prediction Based Functional Images Analysis

Conventionally, functional imaging studies mainly focus on finding regions showing variation under controlled experimental stimuli. The most well-known technique is Statistical Parametric Mapping (SPM) (Friston et al., 1995). Under the assumptions of the general linear model (GLM), the time series at each voxel is modelled by a linear combination of experimental conditions and confounds (e.g. low frequency drifts). The statistical tests are later applied to the weighting of each experimental regressor, to infer where the contrasts of experimental conditions have significant effects on the pattern of brain activity. In other words, the objective is to detect regions of activation in the brain during tasks. Three dimensional statistical maps would be generated, showing activation patterns that relate to contrasts of experimental conditions. The location of activation patterns provides insight into brain function. This is also called an encoding model in the sense that the brain responses are encoded the experimental factors.

In recent years, pattern recognition and machine learning methods have been used to predict, or decode, an experimental variable from high-dimensional functional imaging data. Not all methods are truly multivariate, as some still assume

independence among voxels (Shinkareva et al., 2008). In general, these studies have well-controlled experimental stimuli, and the number of conditions are limited. Measures of predictive accuracy are determined by cross-validation, which involves partitioning the data into training and testing sets. The discriminating machine, or classifier, is trained using the functional images, and labels indicating the corresponding experimental conditions. In the testing phase, the classifier returns the predicted experimental conditions using test images as input. Because the true experimental conditions are known, the predictive accuracy can be calculated. This is also called a decoding model in the sense that it models the decoding of neuronal activity that causes a percept or behaviour. In most studies, the design involved block stimuli with categorical conditions, such as observing different categories of image stimuli or performing different tasks (Carlson et al., 2003; Cox and Savoy, 2003b; Haxby et al., 2001; Haynes and Rees, 2005, 2006; LaConte et al., 2005; Mourao-Miranda et al., 2005).

Scientific theories are essentially models. Within a Bayesian framework, the objective is to determine the model (from a number of candidates) that best describes the observed distribution of data in the most parsimonious way. Such a model essentially says something about what could be expected from unobserved data. If only parts of such data are presented to the model, then it should be possible to use the model to make an informed estimate about the missing information. In other words, it is able to predict what is unknown, from known facts. Models are generally considered "better" if they can be used to make more accurate predictions about unknown data. The real benefit of using Bayesian approaches is that they allow us to determine the structure of the most accurate model (from among the candidates), through the process of Bayesian model comparison. Many scientists take the *realist* view, which considers the structure of the model to be of most scientific interest,

ignoring the actual probability density that is encoded. Unfortunately, most biological systems are extremely complex. If more and more data are collected, or the quality of the data is improved, then increasingly complex models can be supported (Morch et al., 1997). The actual complexity of the model that is deemed to be "best" is largely a function of data quality. A report describing the model structure with the highest evidence may tell us more about the quantity and quality of data, than it does about the biological system itself. Most neuro-imagers prefer to treat estimates of model parameters as the important findings (e.g. SPM maps). Such studies generally involve simplified models, as these allow findings to be more easily visualised and explained. It is acknowledged that the models may depend on unlikely assumptions, but the benefits of adopting them should be evident from the literature. For example, mass-univariate statistical testing in SPM has proven to be a very powerful tool for visualising differences, despite the fact that it usually ignores the possibility of connections among different brain regions.

The *instrumentalist* view of science is that it is the predictions themselves, or the ability to make such predictions, that are of interest (Forster, 2002). Scientific research is sometimes funded according to the contribution it can make (or potentially make) to society. Some of the benefits of neuroimaging may come from its potential to make predictions, rather than from the actual models or parameter estimates. It is difficult to anticipate all the benefits from such predictions, just as it is difficult to anticipate the ultimate utility of any area of research.

Predictive models may also allow different forms of questions to be posed. For example, it becomes possible to estimate whether task C activates a network that is more similar to that activated by task A, or that activated by task B. By accurately characterising the pattern of difference between A and B, it becomes possible to formulate questions in terms of this difference. More accurate characterisations of

differences may also lead to tests with greater sensitivity. This has been demonstrated in studies that applied pattern recognition approaches to particular brain regions (Eger et al., 2008; Haynes and Rees, 2005). Such work has allowed differences to be detected that could not be found by mass-univariate approaches (Kriegeskorte et al., 2006).

1.2 Overview of Chapters

Because I was involved in multiple projects using similar methods, this thesis is mainly divided into technical and application sections. In the technical sections, equations and algorithms are introduced with sufficient details for them to be implemented. The application sections will state which algorithms were used, and the reader should then refer to the appropriate technical section. Specifically, this thesis is largely about kernel pattern recognition approaches, which can be roughly divided into kernel generation and the kernel algorithms themselves. Methods of kernel generation are described in chapter 3, and the algorithms are described in the first half of chapters 4 and 5. The remaining chapters are organised as follows.

Background of Machine Learning Theories and Methods

For readers with a less methodological background, this chapter explains basic concepts of pattern recognition and machine learning, with some practical examples using AD data. The chapter begins with probability theory in the Bayesian framework, which is used throughout the thesis. The notation and equations commonly used in Bayesian methods are described. Only two probability distributions are mentioned, because all the probabilistic algorithms used for this thesis are based on either Gaussian or Bernoulli distribution. A section introduces decision theory, which is essential for classification. Generative and Discriminative models are compared. For

classification problems, a generative model would describe the entire probability distribution of each of the classes of data. The alternative is to use a discriminative model, which only needs to model the probability density of the differences between the classes. Generative models are not usually the most accurate approach for predicting, as they require more hidden variables, so marginalisation over higher dimensional probability densities is needed. Empirical evidence shows that discriminative pattern recognition models usually outperform generative models in terms of their predictive accuracy. That is also the reason why the applications did not use generative models for classification. Simple regression and classification algorithms are illustrated in this chapter, to assist readers to understand the more advanced models described in later chapters. Cross-validation is often used to evaluate the performance of different models. Some models can also be compared using criteria based on the Bayesian evidence framework, which measures the goodness of models in terms of their trade-off between fitting the data and their complexity. In this framework, integrating out the parameters can lead to the conditional probability of data given the model $p(D|M)$ or the "evidence for the model".

Kernel Methods and Kernel Construction from Neuroimaging Data

The first part of this chapter describes mathematical definitions and properties of kernels. Because most algorithms applied in this thesis are kernel methods, it is essential to understand the constraints and limits of kernel methods. Unlike common pattern recognition models, kernel methods take "kernels" as the input rather than features of the data. Intuitively, kernels encode measures of pair wise similarity between all the data points. Information, describing patterns in the training set, is encoded in the kernel. The kernel trick also allows efficient construction of various kernels, which are the equivalent of input features projected into higher dimensions.

This can enable non-linear patterns in the original space to appear linear or separable in the new feature space.

To construct a kernel from imaging data (either functional MRI or structural MRI), we have to establish a measure of similarity. Ugly Duckling Theorem (Duda et al., 2000; Watanabe, 1970) tells us that measures of similarity between things can not exist without prior assumptions. From our knowledge of the physiological basis, we can extract meaningful information that is more related to the conditions we intend to characterise. For example, we know that neurodegenerative diseases would cause grey matter changes more than white matter changes, or we know that low frequency drift in the fMRI time series is more likely to be noise than informative signal (Henson, 2004). To extract the useful “features”, both structural and functional MRI has to be pre-processed. The pre-processing procedures are introduced in this chapter, along with a description of information that may be encoded in the outputs, which are later used to generate the kernels. It is also possible to apply operations that can efficiently remove the confounding factors, such low frequency drifts, ages or genders, directly from the kernel. Temporal compressing techniques for fMRI data are also introduced. The last part of the chapter mentions some basic kernel algorithms, for example kernel principal component analysis, kernel K-nearest neighbour classification, a simple novelty detection method and some clustering methods. These simple algorithms sometimes allow useful visualization of the structure of the patterns.

Kernel Regression Methods and their Application in Functional and Structural MRI

Following the basic regression method introduced in chapters 2 and 3, more advanced kernel regression methods are described in this chapter. These algorithms are Support Vector Regression (SVR), which is a non probabilistic model, together with two probabilistic models, which are Relevance Vector Regression (RVR) and

Gaussian Process Regression (GPR). The first half of the chapter is about the technical details of these three algorithms, whereas the second half describes applications of those methods. Two of the projects are work for the “Pittsburgh Brain Activity Interpretation Competition” (PBAIC) of 2006 and 2007. The competitions were open globally, enabling teams from around the world to test their algorithms on the same dataset. The competition allowed a comparison among a diverse range of approaches for making predictions from brain imaging data. As in any model comparison problem, it allowed the most accurate approach to be selected from a range of candidates. We achieved 5th place in 2006 and 1st place in 2007. Details of how we tackled the tasks are described in the chapter. Another application concerns predicting clinical scores from structural MRI. Unlike conventional correlation analysis, this analysis was based on “predictive power” and we demonstrated that by using RVR, it is possible to achieve good predictive accuracies. The framework also involves a comparison among different clinical scores, as some clinical scores could be more accurately predicted, from the structural images, than others.

Kernel Classification Methods and their Application in Functional and Structural MRI

In the chapter 5, support vector classification (SVC), which is one of the most popular classification algorithms for practical applications, is explained in detail. Two Bayesian classification algorithms, namely Relevance Vector Classification (RVC) and Gaussian Processes Classification (GPC) are described. These two algorithms have similar forms to regularised logistic regression, and the corresponding hyper-parameters can be optimised via marginal likelihood maximisation. A novel multi-class classifier, which utilises the temporal information of fMRI data, is also present. Another classification method introduced is called the one-class classifier, which is based on smallest hypersphere enclosing all the training data. Like chapter 4,

the first half of the chapter is about the technical details of these three algorithms, whereas the second half describes applications of those methods. Three applications are about classification between patients and controls using anatomical MRI data for Alzheimer's disease (AD), Huntington's disease (HD), and major depressive disorder (MDD). Some methods of feature selection are also mentioned. Two applications involved fMRI decoding, one was applied with the novel multi-class classifier, and another one was applied with standard SVC in a searchlight fashion. The novel multi-class classifier demonstrated high predicting accuracy in single subject. The searchlight SVC revealed regions in the hippocampus which are relevant to navigation tasks.

Chapter 2

Background of machine learning theories and methods

Contents

This Chapter will describe the basic probability tools and simple algorithms, which will lead to the advanced algorithms in the later chapters.

2.1 Basic Probability Theory

Probability theory provides a quantitative framework to measure and manipulate uncertainty. In the context of pattern recognition and machine learning, probability rules also enable us to use mathematical language to abstract the practical problems into models and equations.

The commonly used examples to introduce probabilities are either flipping coins or drawing coloured balls from a box. These examples have discrete events over repeatable trials. For instance, we can toss a coin 100 times, and measure the number of times the coin faces up with heads or tails. We introduce the "random variable" $X \in \{'head','tail'\}$, which means it can take the condition of either "head" or "tail". Then we define the probability of having a head as *p(X=head)=Number of heads/number of tosses*. However, as the topic of this thesis focuses on applications of pattern recognition on neuroimaging, we will use practical examples from neuroimaging.

In the context of imaging data, most of the measurements and observations are continuous variables rather than discrete ones. To demonstrate the rules of probability with meaningful examples, we use the left and right hippocampal volumes from 91 control subjects and 99 patients with clinically confirmed Alzheimer's disease as two random variables, *L* and *R*.

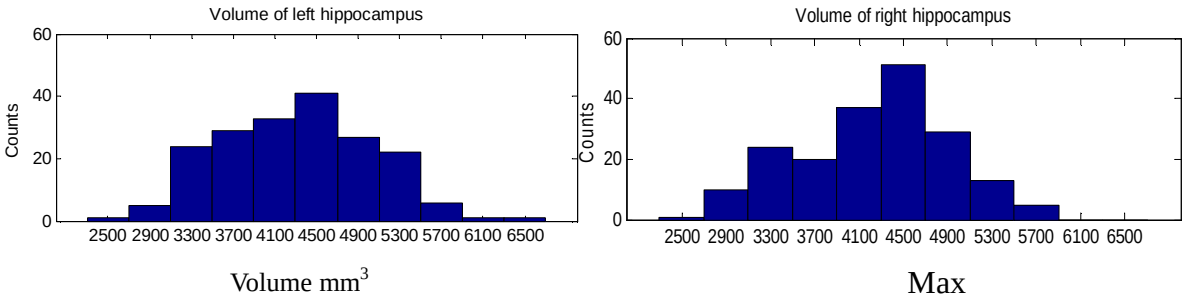


Figure 2.1 Histogram of hippocampal volume

Histogram of the volume of both left and right hippocampi in cubic millimetres.

We present the distribution of the volume for both left and right hippocampus in figure 2.1, using histograms with intervals of 400 cubic millimetres. A histogram is a method of representing the distribution of a sampled population using bins. The horizontal axis is usually specified as non-overlapping intervals of the random variable. The height of each particular bin indicates the frequency or number of samples that lie in the interval. Histograms provide simple ways to discretise continuous variables into frequencies over different intervals. However, because the number of counts depends on the total sample size, to generalize the representation, the heights of each bin are be normalized into "portions" or "percentage" of the population. This is achieved by applying the rule that the probability sums to one over the viable: $\sum_{x \in X} p(x) = 1$ for a discrete variable, and $\int_{-\infty}^{\infty} p(x)d(x) = 1$ for a probability density over a continuous variable. In our histogram example, the heights of each bin are simply divided by the total number of samples to represent the probability. For example, $p(L = 4300 \le l < 4700) = 41/190 = 0.2158$, means if a random subject is selected form the sample set, the probability of observing a left hippocampal volume of between 4300mm^3 and 4700mm^3 is around 22%, or 0.22.

2.1.1 Probability densities

By discretising a continuous variable over a series of intervals (bins), and using normalized histograms to represent probability distributions, this leads to the

mathematical abstraction of “probability density function” (pdf). Assuming we have infinite samples and infinite bins, of which have infinitesimal range over a continuous variable, such as the volume of hippocampus, i.e. $P(X = (x-\delta/2) \le x < (x+\delta/2)) = p(x)\delta x$ for $\delta x \to 0$, then $p(x)$ is called the probability density function (pdf) over *x*. The equation to calculate the probability that *x* will lie within an interval is given by

$$P(a < x < b) = \int_a^b p(x)dx \tag{2.1}$$

By definition, the probability density is non-negative $p(x) \ge 0, \forall x$, but it can have values larger than one, as the definition only bounds the total integral $\int_{-\infty}^{\infty} p(x)d(x) = 1$, to be one.

The “cumulative distribution function” (cdf) of a particular probability density function is defined by the probability that *x* falls in the interval from minus infinity to a particular value.

$$P(a) = \int_{-\infty}^{a} p(x)dx \tag{2.2}$$

The derivative of a cdf equals the pdf, $dP(x)/dx = p(x)$. Notice the cumulative distribution function or cdf, is symbolized by the capital P. Some texts use capital P to denote probability mass function (pmf) for discrete events. Readers should be aware that p(x) may indicate a pdf, cdf or pmf, depending on the context.

2.1.2 Joint probability and conditional probability

Returning to the example of hippocampal volumes represented by histograms, when we consider more than one variable, we can also calculate the joint histogram or joint probability. For example, $p(4300 \le l < 4700, 4000 \le r < 4300) = 9/190 = 0.0474$, means there are 9 subjects, or around 5% of the samples, that satisfy both conditions that the left hippocampal volume is between 4300 mm^3 and 4700 mm^3 and the right hippocampal volume is between 4000 mm^3 and 4300 mm^3.

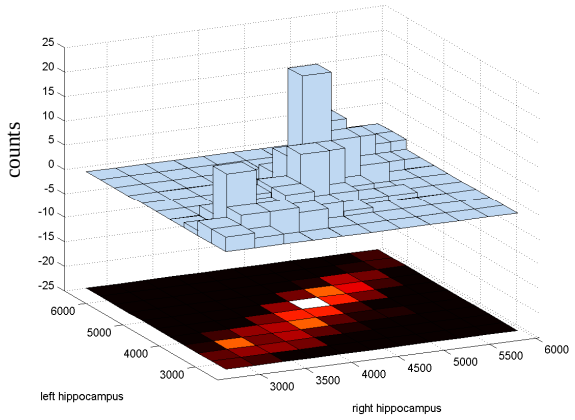


Figure 2.2 Joint histogram of hippocampal volume

Joint histogram of the volume of both left and right hippocampi in cubic millimetres.

Conditional probability is defined by the fraction of particular instances, given the condition of some other instances. For example, $p(4300 \le l < 4700 \,|\, 4000 \le r < 4300) = 9/26 = 0.34624$, means there are 26 subjects that satisfy the condition that the right hippocampal volume is between 4000 mm^3 and 4300 mm^3, and out of those 26 subjects, there are 9 subjects who also satisfy the condition that the left hippocampal volume is between 4300 mm^3 and 4700 mm^3. The relationship between joint probability and conditional probability is given by

$$p(x \,|\, y) = \frac{p(x,y)}{p(y)} \quad \text{or} \quad p(x,y) = p(x \,|\, y)\,p(y) \tag{2.3}$$

We can also marginalise the joint probability with respect to one of the variables to obtain the marginal probability.

$$p(y) = \int_{-\infty}^{\infty} p(x,y)\,dx \quad \text{or} \quad p(x) = \int_{-\infty}^{\infty} p(x,y)\,dy \tag{2.4}$$

In plain English, we say “the probability of x and y is the product of the probability of y and the probability of x given y”. The principles of joint probability, conditional probability, and marginalisation can all generalise to more than two variables. In addition, if $p(x,y) = p(x)\,p(y)$ or $p(x \,|\, y) = p(x)$, we say that both variables are independent.

2.1.3 Bayesian probability

By rearranging equations 2.3 and 2.4, we can reverse the conditional dependence between two variables

$$p(y \mid x) = \frac{p(x \mid y)\, p(y)}{p(x)} \quad \text{or} \quad p(y \mid x) = \frac{p(x \mid y)\, p(y)}{\int p(x \mid y)\, p(y) dy} \tag{2.5}$$

The relationship above, between two conditional probabilities, is called Bayes' theorem. Bayes' rule plays a major role in decision theory, as well as being the foundation for the advanced machine learning methods that will be introduced in later chapters. The general Bayesian view further provides the framework to formulate the calculation of belief using Bayes' rule. As mentioned by Cox (Cox, 1946), there are two ways to conceptualize probability. One is the idea of frequency in a group of ensemble, such as the frequency of drawing coloured balls from a large number of boxes (or repeated trials) with the same contents. The other idea is the reasonable expectation. For example, probability would represent the strength of belief that a white ball will be drawn from a box containing two black balls and one white ball, in a single trial. Using reasonable expectation as probability would make more sense when terms like "probability of raining tomorrow" and "probability of getting elected" are in use, as not all events can be repeated multiple times. In the frequentist (also known as "sampling theory") approach (MacKay, 2002), one calculates the estimators from the samples of interest, and then uses some criterion to select between those estimators. In contrast, we only need to make assumptions on the form of the models and distributions for Bayesian inference, and can rely on the rules of probability and Bayes' theorem to return the quantitative degree of belief.

One practical example of the utility of Bayes' rule is the chance of having HIV when a test shows a positive result (Gigerenzer, 2002; Hunt, 2003). Today's blood test for HIV offers the sensitivity of 99.9% and specificity of 99.99%. That means if

someone is HIV positive, the test will have a probability of 99.9% of giving a positive result, and for a person who does not have HIV, the probability of the test giving a negative result will be 99.99%. In a probabilistic representation, $p(test=+|HIV=+)=0.999$ and $p(test=-|HIV=-)=0.9999$, and the quest is to find $p(HIV=+|test=+)$. Despite hearing propaganda about HIV all the time, the virus infects a very low percentage of the general population in developed countries. For example, only 0.01% of the US population not belonging to a high-risk group has HIV. Therefore we can say $p(HIV=+)=0.0001$. Sometimes, people call this $p(HIV)$ the prior, which means the prior knowledge, or belief, before observing any data. The probability $p(test|HIV)$ is called the likelihood, and expresses how probable the observed data is for different conditions. In fact, what we are interested is the posterior term $p(HIV|test)$, which gives us the probability of having HIV given the test result. Often people state Bayes' theorem in words

$$posterior=\frac{prior\times likelihood}{evidence} \quad \text{or} \quad posterior\propto prior\times likelihood \tag{2.6}$$

The "evidence" is the probability of observing this particular data given all possible conditions. In our example, both the test result and HIV status each have only two states, namely positive or negative. Therefore, the evidence is formulated as $p(test)=\sum_{HIV\in\{+,-\}} p(test|HIV)\,p(HIV)$ and we can represent the solution as

$$p(HIV=+|test=+)=\frac{p(test=+|HIV=+)\,p(HIV=+)}{p(test=+|HIV=+)\,p(HIV=+)+p(test=+|HIV=-)\,p(HIV=-)} \tag{2.7}$$

$$p(HIV=+|test=+)=\frac{0.999\times 0.0001}{0.999\times 0.0001+0.0001\times 0.9999}=0.4998 \tag{2.8}$$

This calculation shows that for someone, not from a high risk group, who has a positive blood test result, the actual probability of having HIV is only 0.5. If the

person takes another blood test, then we can rely on Bayes' theorem to calculate the new probability from the additional observed evidence. By assuming both blood test results are independent, if the second test still shows positive, then this person will now have a 99.99% probability of having HIV.

In practice, the Bayesian formulation provides an elegant way to aggregate all known information. One example is in the context of tissue segmentation (Ashburner and Friston, 2005). The unified segmentation approach combines many components, from the intensity distribution of tissues, the inhomogeneity field of the scanner, to image normalization. By defining a prior distribution for the models, such as the spatial prior of tissue classes and regularization for the image registration, the optimization can be solved on the integrated equation to obtain the posterior probability of each tissue class at each voxel.

2.1.4 Mean and covariance

Simplification is essential to characterize particular samples from a population. The most intuitive way to generalize a particular group is by averaging the observations. For example, it is said that Germans are tall and Japanese are short, and this conception is mainly based on the average heights in both populations. The average, or the mean, of the variable is often denoted as $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, where n is the number of samples. To describe how variable each sample is in the observations, another measurement called variance can be calculated by $\mathrm{var}(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$. This is often called the biased estimate of the variance. An alternative is the unbiased variance estimate, which is defined by $\mathrm{var}(x)_{unbiased} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$. This gives a slightly higher measure, especially when the sample size is small. The sample mean $\bar{x}$ is estimated from the samples, and is further used to calculate the variance. Hence the estimate of the variance should have one less degree of freedom. That is where the

n-1 term comes from. A different proof can be found in Appendix A.

Here, we assume equal probabilities for the variable, but a more generalized formulation can be given for some function $f(x)$ with probability $p(x)$. We call this average, weighted by its probability, the expectation of $f(x)$. It is often denoted as following:

$$\mathrm{E}(f) = \int f(x)\,p(x)dx \qquad \text{or} \quad \mathrm{E}(f) = \sum_{i=1}^{n} f(x_i)\,p(x_i) \tag{2.9}$$

The variance of $f(x)$ is defined as

$$\mathrm{var}(f) = \mathrm{E}[(f(x) - \mathrm{E}[f(\mathrm{x})])^2] \tag{2.10}$$

By expanding the square, we can write the variance in another form

$$\mathrm{var}(f) = \mathrm{E}[f(x)^2] - \mathrm{E}[f(\mathrm{x})]^2 \tag{2.11}$$

There is an advantage of using this formulation when the variance is calculated online or when memory is an issue to store all the observations. Notice that the equation in (2.10) requires the expectation to be computed before the variance can be calculated, whereas equation (2.11) can be used to update the variance and expectation iteratively when a new observation is measured.

In cases when there are more than one variable, another measurement called covariance is computed from pairs of variables:

$$\mathrm{cov}(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \quad \text{or} \quad \mathrm{cov}(x,y) = \mathrm{E}[xy] - \mathrm{E}[x]\mathrm{E}[y] \tag{2.12}$$

Notice that variance is non-negative, but that covariance can be negative. Commonly, with multivariate data, the covariance between each pairs of variable is represented as a covariance matrix. If we define the matrix $\mathbf{X} = [\mathbf{x_1} \quad \mathbf{x_2} \cdots \mathbf{x}_n]^{\mathrm{T}}, \mathbf{x} \in \Re^d$, where each $\mathbf{x}$ is a column vector of one observation or sample with d number of variables (sometimes called the dimension). If we use the hippocampal volume mentioned in previous section as an example, then d=2, and n will be 190. To calculate the covariance matrix, we first remove the mean over the observations from each variable.

$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{x}_{1,1} - \mu_1 & \cdots & \mathbf{x}_{1,d} - \mu_d \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{n,1} - \mu_1 & \cdots & \mathbf{x}_{n,d} - \mu_d \end{bmatrix}$, where $\mu_d = \frac{1}{n}\sum_{i=1}^{n} x_{i,d}$, then the covariance matrix is computed by

$$\boldsymbol{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^{\mathrm{T}} = \frac{1}{n}\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}} \tag{2.13}$$

and the size is *d* by *d*. The matrix is a symmetric positive semi-definite matrix. Notice that the diagonal of the covariance matrix is the variance of each variable. When two variables have positive covariance, it implies these two variables tend to vary in the same direction, i.e. if one variable is very large in one particular observation, the other variable is also likely to be large in the same observation. If the variables have negative covariance, they would be likely to vary in the opposite direction. The covariance matrix plays an important role in linear regression, and also principle component analysis, which will be explained in a later chapter.

In order to provide a standardized measurement describing the co-variation between two variables, Pearson's correlation coefficient is defined as the normalized covariance ranging from -1 to 1.

$$\operatorname{corr}(x, y) = \frac{\operatorname{cov}(x, y)}{\sqrt{\operatorname{var}(x)\operatorname{var}(y)}} \tag{2.14}$$

The correlation matrix can be computed from the covariance matrix by:

$$\mathbf{R}_{i,j} = \frac{\Sigma_{i,j}}{\sqrt{\Sigma_{i,i}\Sigma_{j,j}}} \tag{2.15}$$

2.2 Probability Distributions

We have defined the concept of probability density function in section 2.1.1. In the statistics literature; there are many forms of parametric distributions, of which the

probability distribution varies by adjusting the parameters. Each distribution has its applications and theories associated. In the context of this thesis, we are mainly concerned with the two common distributions used in machine learning: the Normal distribution (also known as the Gaussian distribution), and the Bernoulli distribution.

2.2.1 Gaussian distribution

The Gaussian distribution is probably best known for its bell shape. In the simplest case of a single variable, the probability density function of a Gaussian distribution is defined by

$$N(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \tag{2.16}$$

There are two parameters controlling the shape of this distribution: μ is the mean, and σ is the standard deviation, which is the square root of the variance. In other words, the expectation of *x* equals the mean, $\mathrm{E}(x) = \mu$, and $\mathrm{var}(x) = \sigma^2$. The inverse of the variance, $\frac{1}{\sigma^2}$, is called the precision, which will be mentioned often in later chapters. Recalling the probability rule that the total integral of any distribution is one, we can utilize this property to derive the following equation (it can also be derived from a general method).

$$\int_{-\infty}^{\infty} \exp\left\{-\frac{x^2}{2\sigma^2}\right\} dx = \sigma\sqrt{2\pi} \tag{2.17}$$

Sometimes in the context of neuroimaging, the spread of a Gaussian distribution is specified by the full width at half maximum (FWHM), rather than the variance. The FWHM is defined by the width of the distribution between the points having half the value at the peak. (See Fig 2.3)

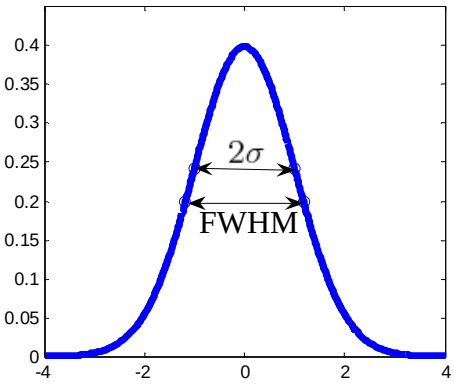


Figure 2.3 Univariate Gaussian

Illustration of the univariate Gaussian distribution with 0 mean, and 1 standard deviation. The FWHM is also shown, and is slightly larger than two standard deviations.

In a Gaussian distribution, both variance and FWHM can be directly calculated from each other. Notice the peak of the Gaussian distribution appears at its mean, hence FWHM is invariant to the mean of the distribution, and we can simplify the equation to

$$\begin{aligned}
&\exp\left\{-\frac{(x_{half})^2}{2\sigma^2}\right\}=\frac{\exp(0)}{2}=\frac{1}{2}\\
&-\frac{(x_{half})^2}{2\sigma^2}=-\ln 2\\
&(x_{half})^2=2\sigma^2\ln 2\\
&x_{half}=\pm\sigma\sqrt{2\ln 2}\\
&\text{FWHM}=x_{half+}-x_{half-}=2\sigma\sqrt{2\ln 2}\approx 2.3548\sigma
\end{aligned} \tag{2.18}$$

The multivariate Gaussian distribution is defined by

$$N(\mathbf{x}\,|\,\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{1}{|\boldsymbol{\Sigma}|^{1/2}\,(2\pi)^{D/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \tag{2.19}$$

where D refers to the dimension, or length, of the vector **x,** the $\boldsymbol{\mu}$ is the vector of means, and $\boldsymbol{\Sigma}$ is the D by D covariance matrix. The multivariate version of equation (2.17) is given by

$$\int_{-\infty}^{\infty}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}d\mathbf{x}=|\boldsymbol{\Sigma}|^{1/2}\,(2\pi)^{D/2} \tag{2.20}$$

This equation is particularly useful when marginalizing over a square exponential

function, such as a Gaussian distribution, is required.

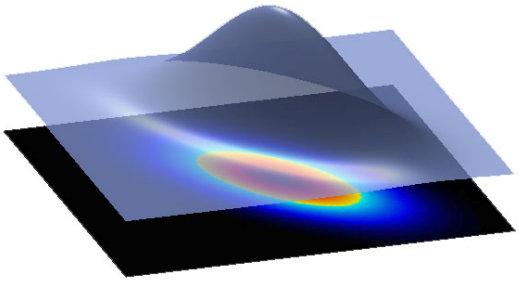


Figure 2.4 Two dimensional Gaussian

Illustration of a two dimensional multivariate Gaussian distribution. The coloured contour shows an elliptical shape, the major and minor axes of the ellipse are the corresponding eigenvectors of the covariance matrix Σ. The major axis is the eigenvector with the highest eigenvalue.

One reason that Gaussian distributions are commonly used in the field of statistical modelling is its simplicity, as it has only two parameters. The reason that Gaussian distributions are so prevalent and widely observed in nature may be the consequence of the central limit theorem, which states that the sum of a set of random variables, which are not necessarily Gaussian distributed, would approach a Gaussian distribution when the number of terms in the sum increases. For example, roll a dice a hundred times and sum up the numbers, then repeat this multiple times, the distribution of the sum of the numbers will approach a Gaussian with mean at 350. However, not everything in the real world follows the Gaussian distribution. Some events have distributions with heavier tails than the Gaussian, such as the chance of economic crises (Buchanan, 2007; Gopikrishnan et al., 1998). For those cases, power law distribution may be more suitable. In this thesis, for the simplicity of most algorithms, only Gaussian distributions are considered for modelling populations and noise.

2.2.2 Parametric models and maximum likelihood (ML) estimates

In the probabilistic framework of machine learning, one main task is to model

the distribution of the population given some samples, such as the hippocampal volumes shown in figure 2.1. Having a nearly infinite number of samples is unlikely, so fully characterising a population density function using histograms, with minimum precision of the observations (i.e. very narrow bins) becomes more difficult. For cases when the sample size is too small to fully cover all possible measurement points, parametric models may provide a more robust way to characterize the distribution of the population - providing the true distribution of the population is not too far from the model assumptions. Usually, parametric models only require a few parameters to fully describe the distribution. For instance, a Gaussian distribution only needs the mean and the covariance. An approach called “maximum likelihood estimation” can be used to estimate model parameters from collected observations. In this formulation, the model parameters that yield the highest likelihood of the observed data would be determined as the solutions. Mathematically, we can define a set of observed data $D = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ drawn from the same distribution independently. In other words, the observed samples are “independent and identically distributed” (i.i.d.). The maximum likelihood estimates of the parameters is defined by

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} p(D \mid \boldsymbol{\theta}) \tag{2.21}$$

When estimating parameters for a Gaussian distribution, we can firstly define the likelihood function of the observed data as

$$p(D \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} N(\mathbf{x}_n \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2.22}$$

Because each observation is assumed to be independent, the likelihood of the dataset is the product of the likelihoods of each observation. The objective is to determine the parameters $\boldsymbol{\mu}_{ML}$ and $\boldsymbol{\Sigma}_{ML}$ that give the highest value of the likelihood function. Since the logarithm is a monotonically increasing function, the parameters that maximize the likelihood function are equivalent to those that maximize the log likelihood

function. The log likelihood for a Gaussian is given by

$$\begin{aligned}&\ln \prod_{n=1}^{N} N(\mathbf{x}_n \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \sum_{n=1}^{N} \ln N(\mathbf{x}_n \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= -\frac{1}{2}\{ND\ln(2\pi) + N\ln|\boldsymbol{\Sigma}| + \sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})\}\end{aligned} \tag{2.23}$$

By setting the derivative of the log likelihood with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to zero, we obtain the solutions for the maximum likelihood of the parameters (Bishop, 2006a; Magnus and Neudecker, 1999)

$$\boldsymbol{\mu}_{ML} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \tag{2.24}$$

$$\boldsymbol{\Sigma}_{ML} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu}_{ML})(\mathbf{x}_n - \boldsymbol{\mu}_{ML})^{\mathrm{T}} \tag{2.25}$$

These solutions are exactly the same as the definition of mean and covariance mentioned in section 2.1.4.

2.2.3 Mixture of Gaussians (MoG)

The simplicity of the Gaussian distribution explains its popularity for modelling probability distributions. However, not all distributions have the same “bell shape”. For example, some distributions may be skew or non-symmetrical. Some distributions may have heavy tails, and some may have multiple peaks. It is possible to model those distributions using other mathematical representations of probability distributions, and one commonly used approach is to model them by linear combinations of *K* Gaussian distributions.

$$p_{MoG}(\mathbf{x}) = \sum_{k=1}^{K}\pi_k N(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2.26}$$

Here, π_k is the mixing portion of each of the *K* Gaussians, which must have a value between zero and one inclusively, $\pi_k \in (0;1)$. Also, these mixing proportions must sum to one, $\sum_{k=1}^{K}\pi_k = 1$.

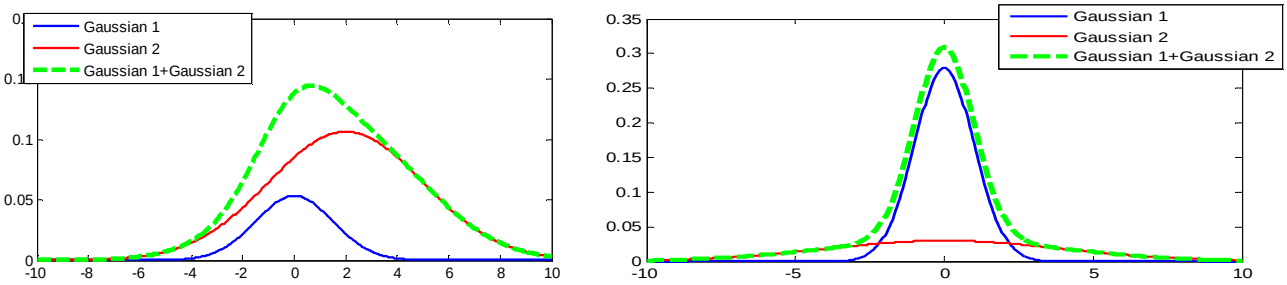


Figure 2.5 Mixture of Gaussians

Illustration of mixtures of two Gaussians. The left figure shows a combination of two Gaussians, with different means and variances, that model a skew distribution. The right figure shows a heavy tailed distribution modelled by two Gaussians with the same mean but different variances.

A mixture of Gaussians can also be used as a clustering technique. By adopting the maximum likelihood approach, we can find the mixing portions, means, and covariances that maximize the log likelihood given by

$$\ln(p(D \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k N(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right\} \tag{2.27}$$

This optimization problem could be solved by a gradient decent approach, but usually it is solved by an "Expectation Maximization" or EM procedure (Dempster et al., 1977). EM algorithms never decrease the likelihood, but EM converges to only a local maximum rather than the global maximum. Therefore, initializing the parameters to reasonable estimates of the optimal values is important. The EM algorithm divides the iterative procedure into two stages (Bishop, 2006a; Ghahramani and Sahani, 2005). The first stage is called the "E step", which fills in values of latent variables according to posterior given data and the current estimate of the parameters. In the context of a mixture of Gaussians, the latent variables are the responsibilities, r_{nk}, of each data point to all K Gaussians.

$$r_{nk} = \frac{\pi_k N(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{i=1}^{K} \pi_i N(\mathbf{x}_n \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} \tag{2.28}$$

The responsibilities can be seen as the belonging probability of a particular sample to a particular Gaussian. In the case of hard classification or clustering, the cluster (Gaussian) to which the sample belongs, is chosen by the cluster k with the highest responsibility, $\hat{k} = \arg\max_k (r_{nk})$. After updating the latent variables in the E step, the next stage is the "M step", which re-estimating the parameters using current estimates for the responsibilities.

$$\begin{aligned} \boldsymbol{\mu}_k &= \frac{\sum_{n=1}^{N} r_{nk} \mathbf{x}_n}{\sum_{n=1}^{N} r_{nk}} \\ \boldsymbol{\Sigma}_k &= \frac{\sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}}{\sum_{n=1}^{N} r_{nk}} \\ \pi_k &= \frac{\sum_{n=1}^{N} r_{nk}}{N} \end{aligned} \tag{2.29}$$

By iterating between the E step and M step, the parameters should converge to a local maximum of the likelihood function.

Because the parameters in a MoG are determined by maximizing likelihood, introducing more Gaussians into the model will always increase the likelihood, which may result in over-fitting of the data. The problem of over-fitting will be described in more detail in later chapters. The choice of the optimum number of Gaussians should be based on model selection criteria (Lee et al., 2006; McKenzie and Alder, 1994).

We applied the MoG and EM algorithm to the dataset of hippocampal volume (Figure 2.6), where the only prior knowledge we provided was the number of Gaussians, K=2. The algorithm had no further information about the controls and patients, yet the groups seemed to be separated quite successfully. If we set the threshold at 0.5 for the responsibilities as the classification boundary, the algorithm

had 96.7% of specificity and 70.7% of sensitivity. However, when the dimensionality is high, the size of the covariance matrix grows quadratically with the number of dimensions. A related issue is the “curse of dimensionality”, which occurs when the number of samples is less than the number of dimensions, and the sample covariance matrix will be non-invertible. Although this problem can be resolved by adding a small constant in the diagonal terms of the covariance matrix, computationally the EM-MoG approach is still very expensive for high dimensional data.

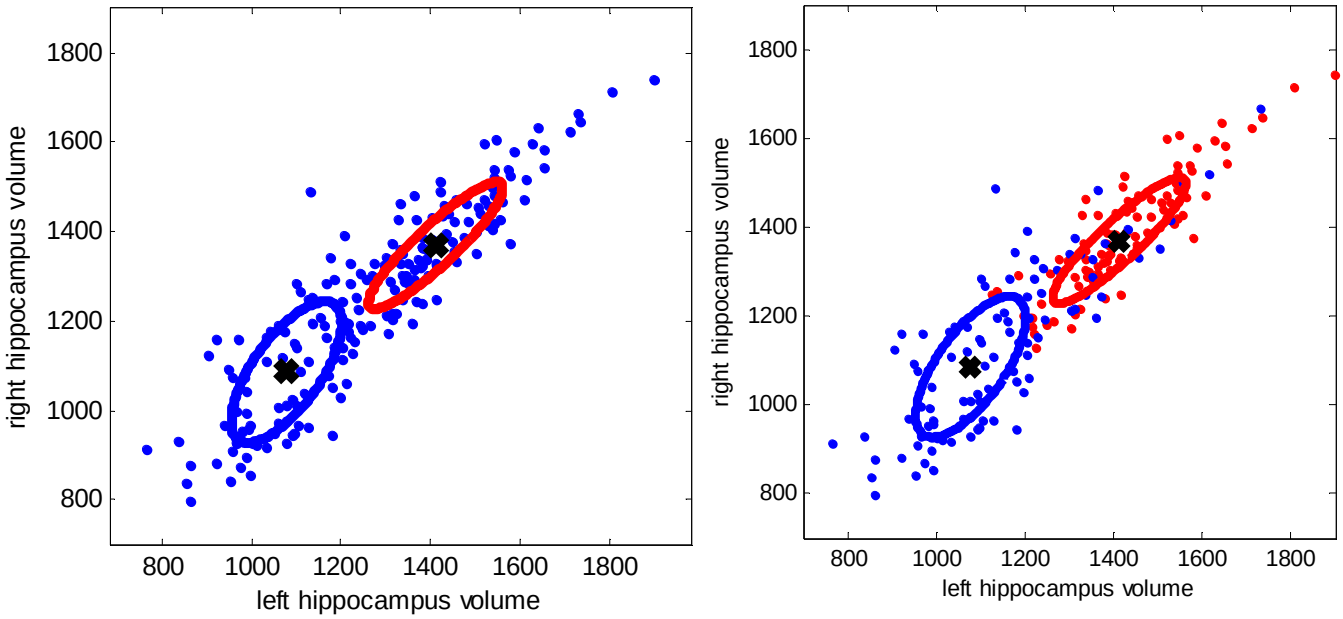


Figure 2.6 EM with a mixture of Gaussians

The left figure shows the clustering result with two Gaussians of the left and right hippocampal volume dataset. The elliptical contours are the one standard deviation boundary for both Gaussian distributions. The crosses are the mean of both distributions. The right figure shows the same dataset and the same clustering results by revealing the identity of the patients and controls. The red colour indicates the controls and the blue indicates the patients. The MoG clustering seems to identify both populations well without any prior information about the patients and controls. An examination of the separation of patients from controls using the responsibilities shows that 96.7% of controls have responsibilities over 0.5 for the red Gaussian, and 70.7% of patients have responsibilities of over 0.5 for the blue Gaussian.

2.2.4 Bernoulli distribution

The Bernoulli distribution is a distribution for binary measurements, the most commonly used example of which is coin flipping. We can define a variable $y \in \{0,1\}$, which indicates head or tail in the coin flipping experiment. The probability of observing y=1 is defined by the parameter μ, so that $p(y=1|\mu)=\mu$, and $p(y=0|\mu)=1-\mu$. Therefore, the probability distribution has the form

$$\text{Bern}(y|\mu)=\mu^{y}(1-\mu)^{1-y} \tag{2.30}$$

The above formulation is derived from the fact that y is a binary variable, so it acts as a switch. When we observe a dataset of binary outcomes, $D=\{y_1,y_2,...,y_N\}$, the likelihood function of observing all those outcomes can be defined as

$$p(D|\mu)=\prod_{n=1}^{N}\mu^{y_n}(1-\mu)^{1-y_n} \tag{2.31}$$

The parameter that maximizes the likelihood of the above equation can be derived by setting the derivative to zero, so that $\mu_{ML}=\frac{1}{N}\sum_{n=1}^{N}y_n$. Here, we can further extend the formulation of maximum likelihood estimation into the general framework on which logistic regression, relevance vector classification, and Gaussian process classification are based. We may have a dataset $D=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),...,(\mathbf{x}_N,y_N)\}\subseteq(\Re^d\times\{1,0\})$, for example, where $\mathbf{x}$ is a two dimensional vector containing both left and right hippocampal volumes, and y is a binary variable indicating whether the subject is a patient or not. In this general formulation, the likelihood function can be written as

$$p(y|\boldsymbol{\theta},\mathbf{x})=\prod_{n=1}^{N}f(\boldsymbol{\theta},\mathbf{x}_n)^{y_n}(1-f(\boldsymbol{\theta},\mathbf{x}_n))^{1-y_n} \tag{2.32}$$

The function $f(\boldsymbol{\theta},\mathbf{x})\in(0;1)$ has the range between 0 and 1, and in practice, it may be a logistic function or a probit function, parameterised by the vector of parameters, $\boldsymbol{\theta}$. Often, we are interested in $\boldsymbol{\theta}_{ML}$, which are the values of $\boldsymbol{\theta}$ that

maximize the likelihood.

2.3 Decision Theory

An essential aspect of machine learning and pattern recognition is not just to learn the pattern and distribution of the observed data, but to make predictions about new data. In the context of clinical diagnosis, it is important to be able to make a decision about the group membership of a subject who underwent some tests, hence classify the subject into either the diseased or non-diseased group.

2.3.1 Bayesian Decision Theory

In the probabilistic framework, we can use the Bayesian probability in equation 2.5 to make the decision based on posterior probability. We can continue to use the Alzheimer's dataset as an example, and define two classes C_1 for patients and C_2 for controls; **x** would still be the volume of hippocampus. What we are interested in are the probabilities of both classes, given the measurements of hippocampus, $p(C_k \mid \mathbf{x})$. Intuitively, we would like to classify a person into the class with the highest posterior probability, $p(C_k \mid \mathbf{x})$. We can show indeed that this intuition is correct mathematically if we want to minimize the misclassification rate. We can define the probability of making a mistake by the following

$$p(mistake \mid \mathbf{x}) = \begin{cases} p(C_1 \mid \mathbf{x}) & \text{if we decide } C_2 \\ p(C_2 \mid \mathbf{x}) & \text{if we decide } C_1 \end{cases} \tag{2.33}$$

The average probability of mistake is given by

$$p(mistake) = \int_{-\infty}^{\infty} p(mistake \mid \mathbf{x})\, p(\mathbf{x}) d\mathbf{x} \tag{2.34}$$

To minimize the *p*(*mistake*), we come up with the Bayesian decision rule for minimizing the probability of a mistake: Decide C_1 if $p(C_1 \mid \mathbf{x}) > p(C_2 \mid \mathbf{x})$;

otherwise decide C_2. In binary classification, since $p(C_1 \mid \mathbf{x}) + p(C_2 \mid \mathbf{x}) = 1$, the decision criteria would be to decide the class that satisfies $p(C_k \mid \mathbf{x}) > 0.5$. At the border, where $p(C_1 \mid \mathbf{x}) = p(C_2 \mid \mathbf{x}) = 0.5$, it is called the decision boundary.

2.3.2 Loss function and Utility function

For many practical situations, the objective is not simply to merely reduce the misclassification rate. In many real world situations, often the penalties of misclassifying class 1 as class 2 are not the same as misclassifying class 2 as class 1. For example, the cost of erroneously misclassifying a patient as healthy, hence delaying the treatment, will certainly result in a higher loss (for the subject) than misclassifying a healthy subject as a patient. Therefore, the optimal decision should be the one that minimizes the expected loss when a misclassification occurs. Sometimes, a utility function, which is the inverse of the loss function, is considered, and the objective would be to make decisions that maximise the expected utility, rather than minimise the expected loss. For instance, the loss from classifying an Alzheimer's patient as normal may be 5, and the loss of classifying a normal into a patient may be 2. When we observe the posterior probability $p(normal \mid \mathbf{x}) = 0.6$, without the penalty of loss, the optimal decision should be put the subject into the normal group. However, when the loss is multiplied by the probability of mistake, based on equation (2.33), the expected loss of classifying the subject as normal is $p(patient \mid \mathbf{x}) \times 5 = (1 - 0.6) \times 5 = 2$, and the expected loss from classifying the subject as a patient is $p(normal \mid \mathbf{x}) \times 2 = 1.2$. In order to minimize the expected loss, the optimal solution should be to treat the subject as a patient.

2.3.3 Discriminative models vs. Generative models

There are commonly two approaches to solve decision problems, the generative methods and the discriminative methods (Bishop, 2007; Ulusoy and Bishop, 2005a; Ulusoy and Bishop, 2005b). The generative method requires the learning of class

conditional probabilities $p(\mathbf{x}|C_k)$. The name "generative" comes from the fact that when re-sampling from the joint distribution, it is possible to generate synthetic examples of the input feature **x**. To solve the inference problem, we can apply Bayes's theorem to calculate the posterior probability

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})} \tag{2.34}$$

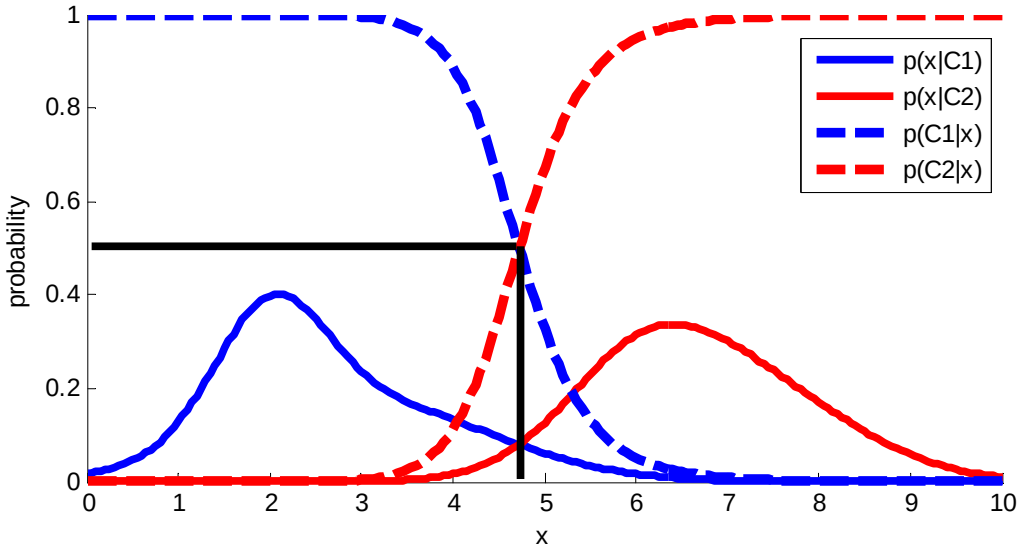


Figure 2.7 Class-conditional densities and corresponding posterior probabilities

Illustration for one dimensional class-conditional density of two classes and their corresponding posterior probabilities, The prior probabilities for both classes are assumed to be the same. Notice both class-conditional density and the posterior probabilities intercept at the value x, at which both posterior probabilities are 0.5.

An alternative approach is to find the conditional distribution $p(C_k|\mathbf{x})$ directly. In practice, discriminative models generally perform better than generative models. The complexity of generative methods is usually much higher than that of discriminative methods. Taking an naïve example of distinguishing spoken words between German and English, if we take the generative approach, we will have learn both German and English well. However, if our only purpose is to discriminate between German and

English, it will be more efficient to learn the differences between German and English. We can especially concentrate on the difficult words and pay less attention to the trivial words. However, the problem will arise when someone speaks Dutch. If we take the generative approach, we will realize it is neither German nor English, but with the discriminative approach, we may misclassify Dutch as German. This may be the reason why a lot of westerners tend to mis-identify Korean as Japanese. The advantages of the generative approach will arise when we want to further distinguish between German, English, Dutch, and French. If we take discriminative approaches, it will require us to learn the new discriminative functions each time we want to identify a new language, but with the generative approach, we will only need to learn the new language (the class conditional distribution of the new language), then we can distinguish all the languages we have learnt.

In the context of neuroimaging, because the input features generally have high dimensionality, it is nearly impossible to learn the class conditional distribution from limited samples. However, when people start pooling datasets together, the generative modelling approach should become more and more accurate.

There are some commonly used generative methods. For example, the Naïve Bayes classifier, which assumes independence between input features, is equivalent to sum of the mass univariate log likelihoods (Hirata et al., 2005). The linear discriminant analysis (LDA) (Sato et al., 2008b) and quadratic discriminant analysis (QDA) assume Gaussian distributions for the class conditional densities. Both LDA and QDA consider covariance structures between features, and LDA makes the further assumption that the within group covariance is the same for all classes.

The common discriminative models, which are also the main focus of this thesis are logistic regression, the support vector classifier (SVC), the relevance vector classifier (RVC), and the Gaussian processes classifier (GPC).

2.4 Basic Machine Learning Algorithms

Before introducing more advance methods, this section will show some basic and prevalent algorithms. There are two main categories in supervised learning. As mentioned in the previous chapter, a supervised learning method requires training from obtained training data. A training set contains input/output pairs, $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$. If the target variable y comes form a set of discrete labels $y \subseteq \{C_1, C_2 ... C_k\}$, then it is a classification problem (e.g. y is the label for patients or normal). If y is a continuous number $y \subseteq \Re$, then it is a regression problem (e.g. y is the age).

2.4.1 Linear least squares regression

The history of least squares fitting goes a long way back, and it is one of the most popular methods in the world. When people refer to regression, by default, they usually mean least squares regression. The basic linear regression models the output as a weighted linear combination of the input features, with an offset term.

$$y = \sum_{d=1}^{D} w_d x_d + w_0 \tag{2.35}$$

Where w_0 is a constant to model the bias or offset. We can also write this in a matrix form, $y = \mathbf{x}^{\mathbf{T}}\mathbf{w} + w_0$, or we can further simplify by adding a constant element in the **x**, so that $\mathbf{x}_* = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$ and $\mathbf{w}_* = \begin{bmatrix} \mathbf{w} \\ w_0 \end{bmatrix}$, $y = \mathbf{x}_*^{\mathbf{T}}\mathbf{w}_*$. For simplicity of notation, we will assume the feature vector **x** contains the constant element by default. Recall the notation in section 2.1.4, where we define a data matrix, $\mathbf{X} = [\mathbf{x_1} \quad \mathbf{x_2} \cdots \mathbf{x}_n]^{\mathrm{T}}$, of input features. To estimate the weight vector **w,** we set up a least squares cost function, so that the optimum weight vector would minimize the sum of squares between the

observed target variables **t** and the predicted output $\mathbf{Xw}$.[1]

$$\arg\min_{\mathbf{w}} \sum_{n=1}^{N} (\mathbf{x}_n^{\mathbf{T}}\mathbf{w} - t_n)^2 = (\mathbf{Xw - t})^{\mathbf{T}}(\mathbf{Xw - t}) \tag{2.36}$$

To find the optimum parameters, **w**, we set the derivative with respect to **w** to 0, which yields the following equation

$$0 = \sum_{n=1}^{N} (\mathbf{x}_n^{\mathbf{T}}\mathbf{w} - t_n)\mathbf{x}_n^{\mathbf{T}} \tag{2.37}$$

This can be written in the matrix notation as

$$\begin{aligned} 0 &= \mathbf{X}^{\mathbf{T}}(\mathbf{Xw} - \mathbf{t}) \\ \mathbf{X}^{\mathbf{T}}\mathbf{Xw} &= \mathbf{X}^{\mathbf{T}}\mathbf{t} \\ \mathbf{w} &= (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{t} \end{aligned} \tag{2.38}$$

This is often referred to as estimating parameters by ordinary least squares (OLS), and the data matrix **X** is sometimes known as a design matrix (Friston et al., 2007c; Friston et al., 1995). The OLS solution can also be framed as a maximum likelihood estimate with Gaussian noise.

$$t = \mathbf{w}^{\mathbf{T}}\mathbf{x} + \varepsilon \tag{2.39}$$

The error, ε , is a zero mean Gaussian random variable with variance σ^2 . Therefore, we can express the likelihood function as

$$p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \sigma^2) = \prod_{n=1}^{N} N(t_n \mid \mathbf{w}^{\mathbf{T}}\mathbf{x}_n, \sigma^2) \tag{2.40}$$

Then we obtain the log likelihood function

$$\ln p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \sigma^2) = \frac{1}{2}\left\{ N \ln \sigma^{-2} - N \ln(2\pi) - \sigma^{-2} \sum_{n=1}^{N} (\mathbf{x}_n^{\mathbf{T}}\mathbf{w} - t_n)^2 \right\} \tag{2.41}$$

If we set the derivative of the log likelihood function to 0, and solve, then we obtain expressions (2.37) and (2.38).

---

[1] Although the output/target variable was loosely defined in the previous sections, to avoid further confusion in the equations, we will use **t** to refer to the observed target vector in the training set, and **y to** refer to the predicted output vector from the model.

2.4.2 Regularized least squares regression

When the sample size is limited, in order to solve ill-posed problems (i.e. $\mathbf{X^T X}$ is non-invertible (Tarantola, 2004)) or to prevent over-fitting, some form of regularization is often introduced into the model. The most common regularizer involves also minimising the sum of squares of the parameters. This is also known as "ridge regression".

$$\arg\min_{\mathbf{w}} \sum_{n=1}^{N} (\mathbf{x}_n^{\mathbf{T}}\mathbf{w} - t_n)^2 + \lambda \| \mathbf{w} \|^2 = (\mathbf{Xw - t})^{\mathbf{T}}(\mathbf{Xw - t}) + \lambda \mathbf{w}^{\mathbf{T}}\mathbf{w} \tag{2.42}$$

The regularization parameter λ, also called the decay or shrinkage term, controls the amount of regularization. When λ is large, the weight vector **w** will shrink toward zero, and when λ approaches zero, the estimated **w** will have a nearly identical solution to that obtained by OLS. To find the optimal solution, we set the derivative of equation with respect to **w** (2.42) to zero.

$$0 = \sum_{n=1}^{N} (\mathbf{x}_n^{\mathbf{T}}\mathbf{w} - t_n)\mathbf{x}_n^{\mathbf{T}} + \lambda \mathbf{w} \tag{2.43}$$

Which can be written in matrix notation as

$$\begin{aligned} 0 &= \mathbf{X}^{\mathbf{T}}(\mathbf{Xw} - \mathbf{t}) + \lambda \mathbf{w} \\ (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda \mathbf{I})\mathbf{w} &= \mathbf{X}^{\mathbf{T}}\mathbf{t} \\ \mathbf{w} &= (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{t} \end{aligned} \tag{2.44}$$

In the Bayesian view of ridge regression (Hsiang, 1975), the regularization can be viewed as priors on the weight vector. The prior is often modelled by zero mean Gaussian with the hyper-parameter, α, which denotes the precision (the inverse of the variance) of the prior distribution.

$$p(\mathbf{w} \mid \alpha) = N(\mathbf{w} \mid 0, \alpha^{-1}\mathbf{I}) \tag{2.45}$$

The resulting posterior distribution is proportional to the product of the prior (2.45) and the likelihood (2.40)

$$p(\mathbf{w} \mid \mathbf{t}, \mathbf{X}, \alpha, \sigma^2) \propto p(\mathbf{t} \mid \mathbf{w}, \mathbf{X}, \sigma^2) p(\mathbf{w} \mid \alpha) \tag{2.46}$$

The log of the posterior distribution

$$\ln p(\mathbf{w} \mid \mathbf{X}, \mathbf{t}, \alpha, \sigma^2) = -\frac{1}{2}\left\{ \sigma^{-2} \sum_{n=1}^{N} (\mathbf{x}_n^{\mathbf{T}}\mathbf{w} - t_n)^2 + \alpha \mathbf{w}^{\mathbf{T}}\mathbf{w} \right\} + \text{constant} \tag{2.47}$$

Now we can see the similarity between the objective function in ridge regression (2.42) and the log of the posterior (2.47) .The maximum posterior weight is

$$\mathbf{w}_{\mathbf{MAP}} = \sigma^{-2}(\sigma^{-2}\mathbf{X}^{\mathbf{T}}\mathbf{X} + \alpha\mathbf{I})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{t} \tag{2.48}$$

If we take the ridge regression view, the regularization parameter is equivalent to the product of the variance of the noise and the precision of the prior distribution $\lambda = \sigma^2\alpha$ (Bishop, 2006a; Hsiang, 1975).

In practice, the optimal regularization parameter could be learnt empirically through cross validation, which will be explained in later sections. However, we can also take the Bayesian approach, which is to marginalize with respect to the weight vector and find the hyper-parameters that can maximize the evidence function $p(\mathbf{t} \mid \sigma^2, \alpha)$. This will lead to the Bayesian learning of relevance vector machines in chapter four.

2.4.3 Logistic least squares regression

Least squares regression tries to minimize the mean squared difference between the predicted and observed variables. In the framework of a generalized linear model, it is possible to apply a non-linear function $f(\mathbf{x}^{\mathbf{T}}\mathbf{w})$ and convert the linear combination of the input features into non-linear outputs. Many different link functions could be used, but in this thesis only the logistic function will be considered.

A logistic function is one type of squashing function, which constrains the output between the range of zero and one. The Probit function is also another squashing function. The definition of the logistic function is

$$f(x) = \frac{1}{1+\exp(-x)} = \frac{\exp(x)}{\exp(x)+1} \tag{2.49}$$

This leads to an interesting property of the logistic model that $f(x)+f(-x)=1$

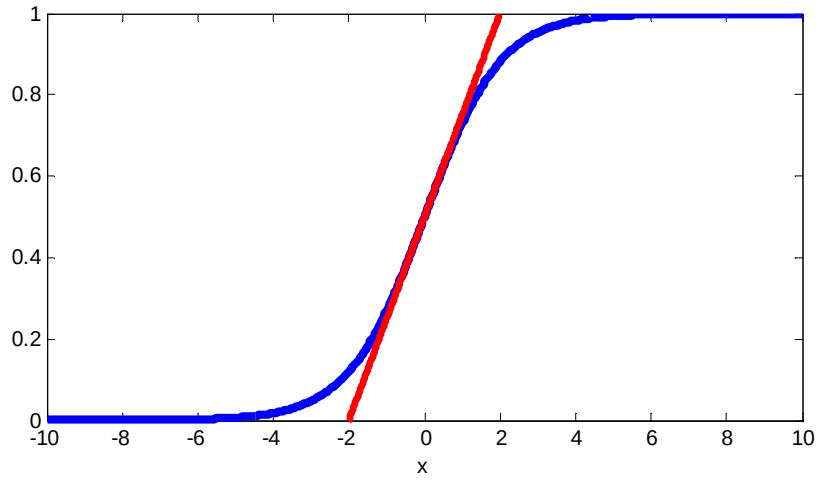


Figure 2.8 Logistic function

The logistic function constrains the output range between 0 and 1. Notice the regions around the middle of the function, where x=0, is approximately linear, with a gradient of 0.25.

The derivative of the logistic function also has a unique property

$$\frac{df(x)}{dx} = \frac{1}{1+\exp(-x)} \cdot \frac{\exp(-x)}{1+\exp(-x)} = f(x)(1-f(x)) \tag{2.50}$$

This equation implies $df(x)/dx = df(-x)/dx$. To obtain the weight vector of the logistic regression, we first define a least squares error function.

$$E(\mathbf{w}) = \sum_{n=1}^{N} (f(\mathbf{x}_n^{\mathrm{T}}\mathbf{w}) - t_n)^2 \tag{2.51}$$

Since *f(x)* is no longer a linear function, it is not possible to obtain a closed-form solution similar to equation (2.44). Therefore, estimating **w** requires iterative methods, such as the Newton-Raphson optimization:

$$\mathbf{w}_{new} = \mathbf{w}_{old} - (\nabla\nabla E(\mathbf{w}))^{-1}\nabla E(\mathbf{w}) \tag{2.52}$$

Please refer to the Appendix B for details of the derivatives and the implementation of least squares logistic regression. To demonstrate the application of logistic regression, we applied the algorithm to a subset of the hippocampal dataset, containing 179 subjects, and apply the regression to the hippocampal volumes and the Mini-Mental State Examination (MMSE) scores (Fuller et al., 1975; Perneczky et al., 2006)

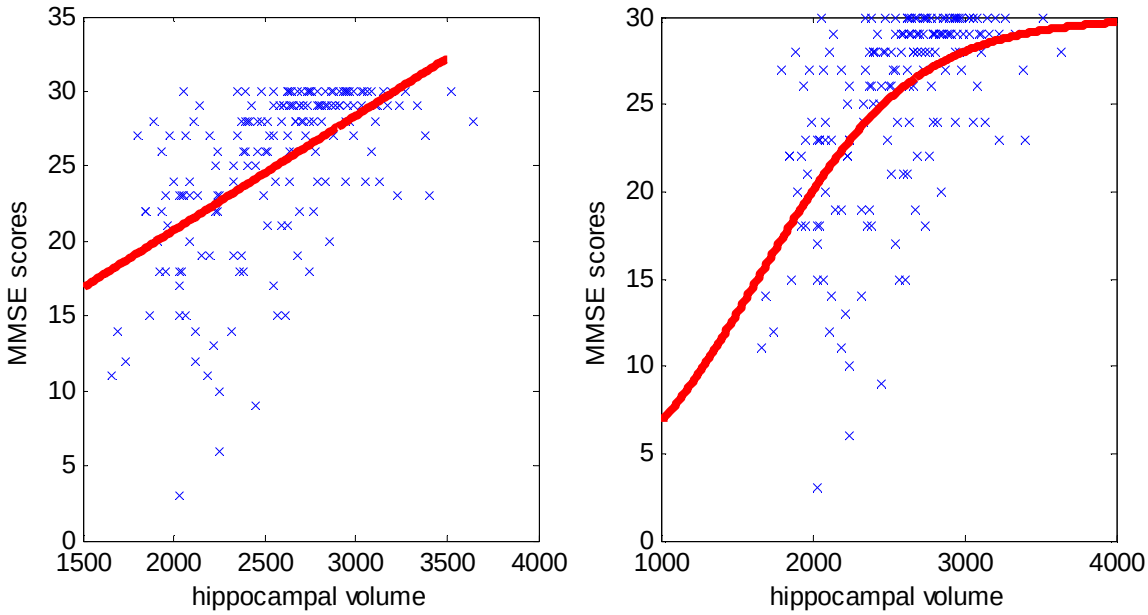


Figure 2.9 Linear regression versus logistic regression

The left figure shows a linear regression of the hippocampal volumes and the MMSE scores. The right figure shows a logistic regression. The maximum scores of MMSE is 30.

Because the MMSE scores range from 0 to 30, it would make sense to use a logistic regression to avoid the capping effect from linear regression. We also scale down the MMSE to between 0 and 1 before applying the logistic regression. The results are shown in Figure 2.8.

2.4.4 Linear discriminant methods for classification

In section 2.3, we described both the generative and discriminative approaches to the classification problem. In this section, we focused on the binary discriminant function, which requires a weighted linear combination of input features, as in equation (2.35). If we define the labels by $y \in \{-1,1\}$, then binary discriminant functions should have the form

$$y=\begin{cases} 1 & \mathbf{w}^{\mathrm{T}}\mathbf{x}+w_0>0 \\ -1 & \mathbf{w}^{\mathrm{T}}\mathbf{x}+w_0<0 \end{cases} \tag{2.53}$$

The simplest method to determine the weight vector could be just treating the labels as the target variables in a regression problem, and solving this using least squares. The solution in this formulation will minimize the square of the distance between each sample to the decision boundaries.

Another approach would be to use the perceptron algorithm (Rosenblatt, 1962), which is the primitive version of the artificial neural network. It usually uses a gradient based optimization to minimize the misclassification errors iteratively. The perceptron algorithm can guarantee a solution which has no classification error if the classes are separable, however, there are an infinite number of possible solutions, and the final solution would rely on the initial conditions and learning rate of the perceptron algorithm. Neither least squares, nor the perceptron methods, are based on a probabilistic framework.

For linear discriminant models, the decision boundary is defined by the subspace of the input space that satisfies $\mathbf{w}^{\mathrm{T}}\mathbf{x}+w_0 = 0$. The decision boundary is orthogonal to the weight vector **w**, hence it has one less dimension than **w** (and w_0). When the input space is two dimensional, the decision boundary is a line, and in three dimensional input space the boundary would be a plane. When the input space is high dimensional, then the decision boundary is sometimes referred to as a “hyper-plane”.

2.4.5 Fisher’s linear discriminant analysis (LDA)

Fisher’s linear discriminant method, also known as linear discriminant analysis (LDA), is the most well known linear discriminant algorithm. This comes from its simplicity and robustness when the distributions of the two classes are Gaussian. LDA projects all the data points to a one dimensional space and aims to maximize the inter-group separation, as well as minimize intra-class variation. Because the

magnitude of the projection vector is not important, we can calculate the vector

$$\mathbf{w} = \mathbf{S}_{\mathrm{w}}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \tag{2.54}$$

Where $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ are the mean vectors of both classes,

$$\boldsymbol{\mu}_1 = \frac{1}{N}\sum_{n\in C_1}\mathbf{x}_{\mathrm{n}}, \qquad \boldsymbol{\mu}_2 = \frac{1}{N}\sum_{n\in C_2}\mathbf{x}_{\mathrm{n}} \tag{2.55}$$

and $\mathbf{S}_{\mathrm{w}}$ is the within class covariance, given by

$$\mathbf{S}_{\mathrm{w}} = \sum_{n\in C_1}(\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}} + \sum_{n\in C_2}(\mathbf{x}_n - \boldsymbol{\mu}_2)(\mathbf{x}_n - \boldsymbol{\mu}_2)^{\mathrm{T}} \tag{2.56}$$

The bias term, w_0, is commonly defined in the way that the average of the means of both classes would lay on the boundary in the projected space. $w_0 = -\frac{1}{2}(\boldsymbol{\mu}_2 + \boldsymbol{\mu}_1)^{\mathrm{T}}\mathbf{w}$

We can also take a generative approach by assuming both class conditional probabilities are Gaussian distributed with equal covariances (Li, 2008)

$$p(\mathbf{x}\,|\,C_1) = N(\mathbf{x}\,|\,\boldsymbol{\mu}_1, \boldsymbol{\Sigma}), \qquad p(\mathbf{x}\,|\,C_2) = N(\mathbf{x}\,|\,\boldsymbol{\mu}_2, \boldsymbol{\Sigma}) \tag{2.57}$$

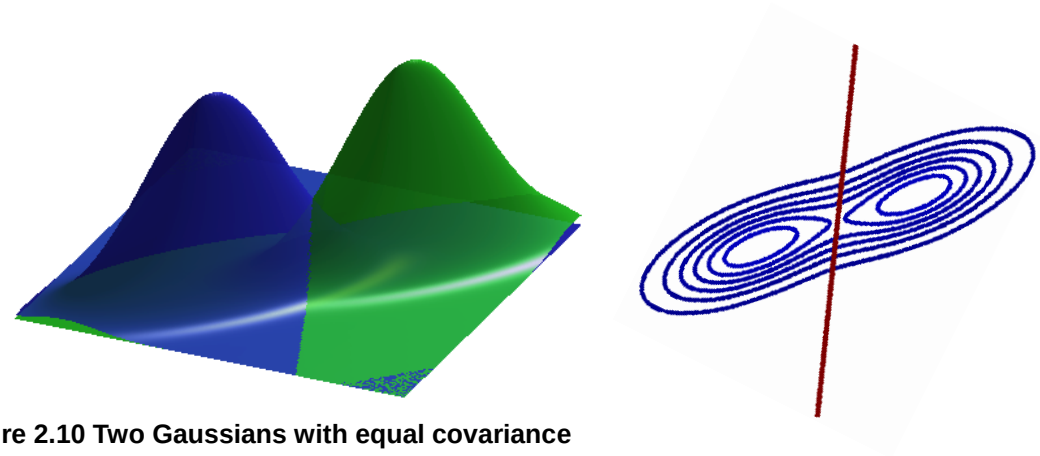


Figure 2.10 Two Gaussians with equal covariance

The left figure shows two Gaussian distributions with equal covariance, and the right figure shows their corresponding contours. The boundary, which is defined by having the same class conditional probabilities for both classes, is shown in the red line.

If we assume both classes have equal priors, then the decision boundary will be the subspace that satisfies $\frac{p(\mathbf{x}\,|\,C_1)}{p(\mathbf{x}\,|\,C_2)} = 1$ or $\log(p(\mathbf{x}\,|\,C_1)) = \log(p(\mathbf{x}\,|\,C_2))$. Therefore, the $\mathbf{x}$ in the space of the decision boundary must satisfy the condition that

$(\mathbf{x}-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)=(\mathbf{x}-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)$. Expanding both sides we can derived the following equation

$$\begin{aligned} -2\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1+\boldsymbol{\mu}_1{}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1+2\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2-\boldsymbol{\mu}_2{}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2=0 \\ \mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)-\frac{1}{2}(\boldsymbol{\mu}_2+\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)=0 \end{aligned} \tag{2.58}$$

If we reformulate the second equation in (2.58) into the linear discriminant equation, $y=\mathbf{w}^{\mathrm{T}}\mathbf{x}+w_0$, then we can derive the identical solution $\mathbf{w}=\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)$, and $w_0=-\frac{1}{2}(\boldsymbol{\mu}_2+\boldsymbol{\mu}_1)^{\mathrm{T}}\mathbf{w}$ as in (2.54). Different priors on the classes would be equivalent to changing the offset term w_0.

LDA generally yields good performance in low dimensional data. With high-dimensional imaging data, LDA has the drawback that the within-class covariance matrix, $\mathbf{S}_{\mathrm{w}}$, is often non-invertible when the number of samples is lower than the number of dimensions. This usually can be resolved by regularization methods (Chen et al., 2000; Thomaz and Gillies, 2005) to ensure that $\mathbf{S}_{\mathrm{w}}$ is invertible.

2.4.6 Logistic regression

We mentioned the least-square logistic regression model with a continuous target variable in 2.4.3. In this section, we introduce a logistic model with a different objective function to solve the linear classification problem. The logistic model for binary classification arises from the assumption that the log of the ratio of both posterior probabilities has a linear relationship.

$$\log\frac{p(C_1\mid x)}{p(C_2\mid x)}=\mathbf{w}^{\mathrm{T}}\mathbf{x}+w_0 \tag{2.59}$$

Then we can derive the logistic model for the posterior probability.

$$
\begin{aligned}
& z = \mathbf{x}^{\mathbf{T}}\mathbf{w} + w_0, \qquad \frac{p(C_1 \mid x)}{p(C_2 \mid x)} = \exp(z) \\
& \frac{p(C_1 \mid x)}{p(C_2 \mid x)} + \frac{p(C_2 \mid x)}{p(C_2 \mid x)} = \exp(z) + 1 = \frac{p(C_1 \mid x) + p(C_2 \mid x)}{p(C_2 \mid x)} \\
& \because\ p(C_1 \mid x) + p(C_2 \mid x) = 1 \\
& \therefore\ p(C_2 \mid x) = \frac{1}{\exp(z)+1}, \quad p(C_1 \mid x) = \frac{1}{\exp(-z)+1}
\end{aligned}
\tag{2.60}
$$

Because it is a binary classification, we can apply the Bernoulli distribution discussed in section 2.2.4 for assembling the likelihood function. For the convenience of the mathematical formulation, we changed the label from -1 and 1 to 0 and 1, $t \in \{0,1\}$.

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^{N} f(\mathbf{w}^{\mathbf{T}}\mathbf{x}_n)^{t_n}(1 - f(\mathbf{w}^{\mathbf{T}}\mathbf{x}_n))^{1-t_n} \tag{2.61}$$

The function f is the logistic function defined in (2.49). For the sake of clarity, we also take the augmented feature vector $\mathbf{x}$ to incorporate the constant offset in the feature set. The log-likelihood function is

$$\ln p(\mathbf{t} \mid \mathbf{w}) = \sum_{n=1}^{N} \{t_n \ln f(\mathbf{w}^{\mathbf{T}}\mathbf{x}_n) + (1 - t_n)\ln(1 - f(\mathbf{w}^{\mathbf{T}}\mathbf{x}_n)\} \tag{2.62}$$

As in the case of least squares logistic regression, there is no closed form solution to the log-likelihood function. Therefore, an iterative method such as the Newton-Raphson update method (2.52) is required. The gradient of the log-likelihood function is

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t} \mid \mathbf{w}) = \sum_{n=1}^{N} (t_n - f_n)\mathbf{x}_n = -\sum_{n=1}^{N} (f_n - t_n)\mathbf{x}_n = -\mathbf{X}^{\mathbf{T}}(\mathbf{f} - \mathbf{t}) \tag{2.63}$$

where $f_n = f(\mathbf{w}^{\mathbf{T}}\mathbf{x}_n)$. The second derivatives of the log-likelihood function, also know as the Hessian matrix, are

$$\nabla\nabla \ln p(\mathbf{t} \mid \mathbf{w}) = -\sum_{n=1}^{N} f_n(1 - f_n)\mathbf{x}_n\mathbf{x}_n^{\mathbf{T}} = -\mathbf{X}^{\mathbf{T}}\mathbf{R}\mathbf{X} \tag{2.64}$$

where the $\mathbf{R}$ is a diagonal matrix with elements $f_n(1 - f_n)$, which is also the gradient of the logistic function (2.50). Therefore, the update equation for the logistic model is

simply

$$\mathbf{w}_{new} = \mathbf{w}_{old} - (\mathbf{X}^{\mathbf{T}}\mathbf{R}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}(\mathbf{f} - \mathbf{t}) \tag{2.65}$$

See Appendix B for an implementation in MATLAB.

To show the classification capability, we applied both LDA and logistic regression to the Alzheimer's dataset. The decision boundaries determined by both methods are similar. In a simulation study (Pohar et al., 2004), it was found that when the normality assumption of both classes were not too badly violated, both LDA and logistic regression yield nearly identical solutions. However, when the assumption of normality fails, logistic regression would perform favourably compared with LDA. This is because logistic regression does not have the assumption of normality, and does not require the estimation of a covariance matrix. Logistic regression does still assume symmetry of the posterior distributions for both classes, and extension of logistic regression will be described in the section 5.1.2.

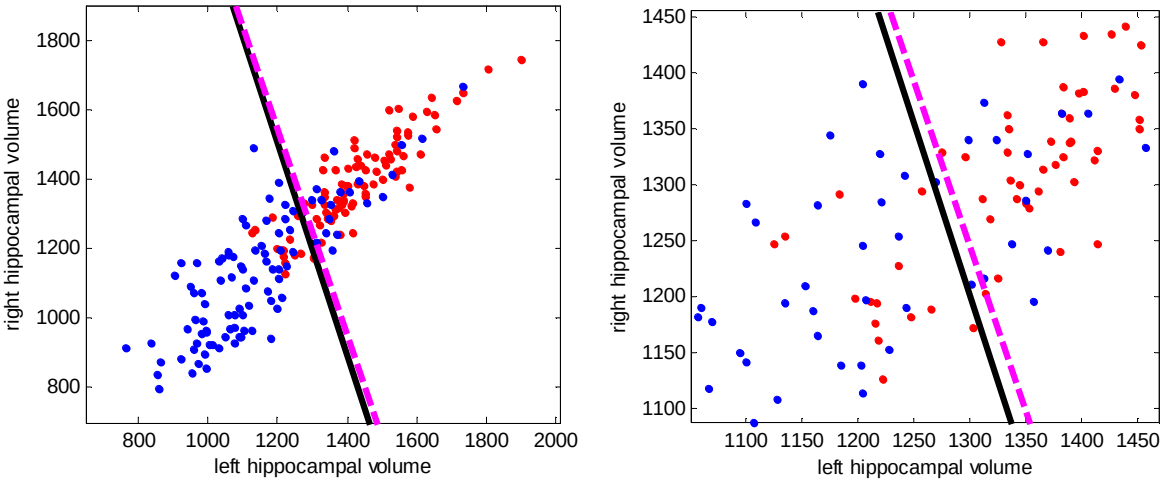


Figure 2.11 Fisher's linear discriminant

The figure shows the decision boundaries determined by both fisher's linear discriminant analysis (LDA) (Black line) and logistic regression (Magenta dashed line) for the Alzheimer's dataset, blue indicates the patient population, red indicates the controls. Since both populations satisfy the Gaussian distribution, and have similar covariance, the boundaries computed by both methods are also similar. The right figure zooms in closely to the decision boundary.

2.5 Cross validation and Model Comparison

In the context of machine learning and pattern recognition, the predictive ability (empirical success) of a model is essentially the measure of goodness for a particular problem. Different models and learning algorithms have different assumptions and learning strategies. Unbiased determination of prediction accuracy is crucial to selecting a desirable model for a specific pattern.

2.5.1 Cross validation and overfitting

When a model is trained, the accuracy with which it explains the training data does not necessarily reflect the true generalization error that would occur when it is used to make predictions - especially when the training set is small. Often, when an algorithm is finely tuned to minimize training errors, a phenomenon called "*overfitting"* would occur. Although training errors may be very small for an overly complex model, the generalization performance may be poor. Therefore to validate the performance of a learning algorithm, we sometimes use cross validation to provide an empirical measure of the generalization performance. Cross validation is not only used to estimate the performance of an algorithm, it is also often applied to tune parameters, such as the regularization parameter in ridge regression.

Cross validation techniques involve splitting the full dataset into a training set and a test set, repetitively. At each repetition, the algorithm is trained using the training set, and the trained model is applied to the test set. The average error over all the iterations, between the predicted outcome of the test set and the real target outcome, gives the test error.

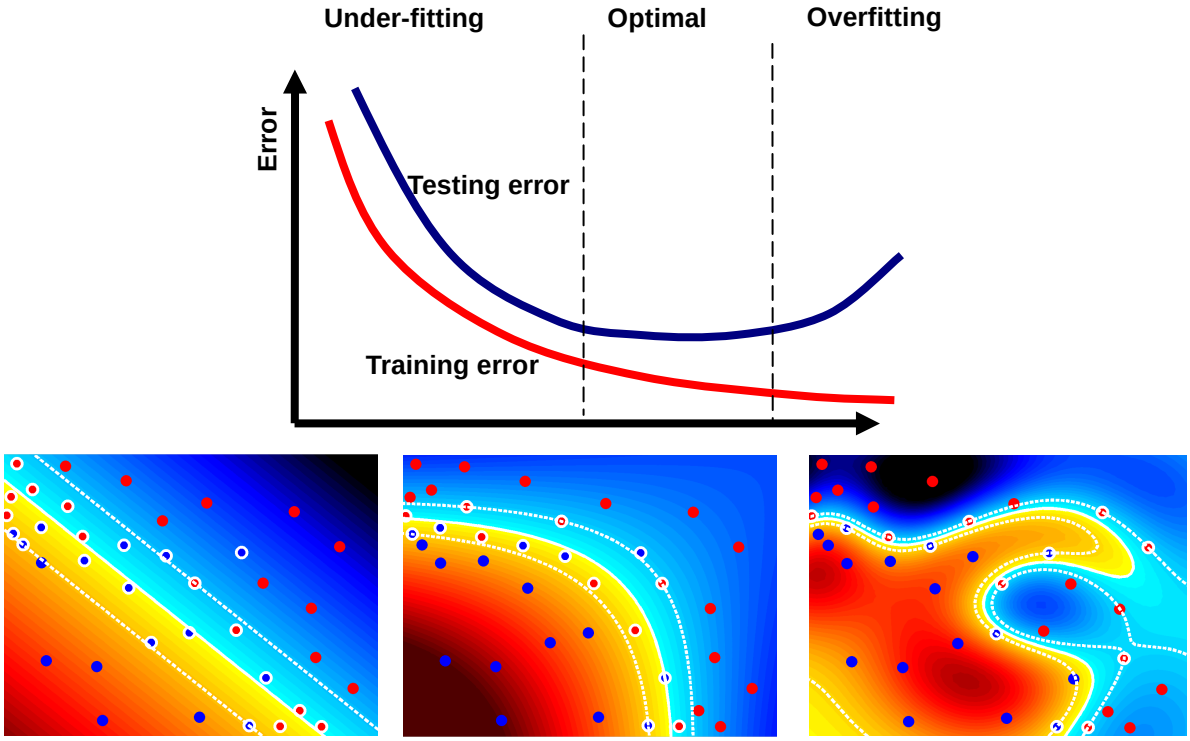


Figure 2.12 Risks of overfitting and underfitting

The above figure illustrates overfitting and underfitting. The three figures at the bottom demonstrate three possible models of binary classification with the same data points. Red and blue regions are learnt by the models after training. The left one is a linear model which under-fits the data, and the right model has no training errors, but it exhibits a complicated decision boundary. The middle figure is a compromise between training accuracy and the flexibility of the boundary, hence it may be the optimal model for this particular dataset. (The lightness of the colour indicate the strength to the classification likelihood, the darker the colour, the more likely that regions belong to the particular class.)

The most common method is called K-fold cross-validation. The procedure works by partition the dataset into *K* equal size subsets. For each validation, *K*-1 subsets (folds) are trained and the remaining fold is used for testing. The procedure will loop *K* times. At each iteration, a different subset will be chosen as the new testing set. This ensures all the samples will be including in the testing set at least once. If *K* equals the size of the training set, then at each validation run, only one sample will be left out, hence it is called the leave-one-out cross-validation (loocv).

Finding a desirable *K* involves a compromise between computational time and the training set size. If *K* is too small, the training set will be relatively small at each run. In practice, the choice of the number of folds depends on the size of the dataset.

In cases when a model has free parameters, or the training procedure includes feature selection, in order to estimate the true testing error, a three way split (Cheng et al., 2008; Ritchie et al., 2003; Su et al., 2007) should be employed. The data needs to be partitioned into three sets, namely the training set, the validation set, and the test set. The procedure works as following. First, the model is trained using a training set with specific parameters, and the prediction accuracy is evaluated using the validation set. This procedure is repeated for all the choices of parameters and models. The best model (and associated parameter set) is selected, and trained using the combination of training and validating set. The trained model is then applied to the test set to evaluate its testing accuracy. It is also possible to apply K-fold cross-validation for both validation and testing. K-fold three way splits will be a double layered loop, where the inner loop runs over a subset of data to select the best model and parameters, and the outer loop evaluates the testing error. The test set should always be intact when computing the performance of different models in the validation phase. Some published experiments did not use a full three way split. These papers reported only the best validating accuracy of the best model, which may be too optimistic and under estimate the generalization error.

2.5.2 Evaluating performance

After performing cross-validation, we can obtain predicted labels (for classification) or values (for regression). For regression algorithms we often evaluate the performance using root mean square error (RMSE) or correlation. RMSE is defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(t_n - y_n)^2} \tag{2.66}$$

where *t* are the observed, or true, values in the dataset, and *y* are the predicted values estimated by the model. If we are not interested in the offset and scale between the true values and the predicted value, we can take the correlation (2.14) of both as the measure of performance.

For evaluating classification results, the simplest measurements would be the classification accuracy rate, which is calculated from the number of correctly predicted samples divided by the total number of predicted samples. Often, a single measurement is not sufficient, especially in the cases of disease diagnosis, when the costs of classifying patients into normal and the reverse are not the same. To present more information, we often create a confusion matrix or table of confusion (Hastie et al., 2003).

	Predicted outcome	
True label	Positive (patient)	Negative (normal)
Positive (patient)	True positive (Tp)	False negative (Fn)
Negative (normal)	False positive (Fp)	True negative (Tn)

The sensitivity of the classifier is the number of true positives divided by the total number of real positives. In our example, it will be the number of patients. The specificity of the classifier is the number of true negatives divided by the total number of real negatives (controls).

$$\text{sensitivity}=\frac{\text{Tp}}{\text{Tp+Fn}}, \quad \text{specificity}=\frac{\text{Tn}}{\text{Tn+Fp}} \tag{2.66}$$

In our example, sensitivity is the accuracy of detecting patients of the given test, and

the specificity will be the accuracy of classify a non-diseased subject as normal. If we recall the decision function for the linear classifier (2.53), it is possible to adjust the bias, hence to trade between specificity and sensitivity. Considering figure 2.10, if the decision boundary is shifted orthogonally to the right, then more patients will be classified correctly, but some normal subjects will be classified as patient (false positive). To visualize the trade-off between sensitivity and specificity, we can plot the receiver-operating characteristic (ROC) curve, and use the area under the curve (AUC) as a measure of classifier performance (Huang and Ling, 2005).

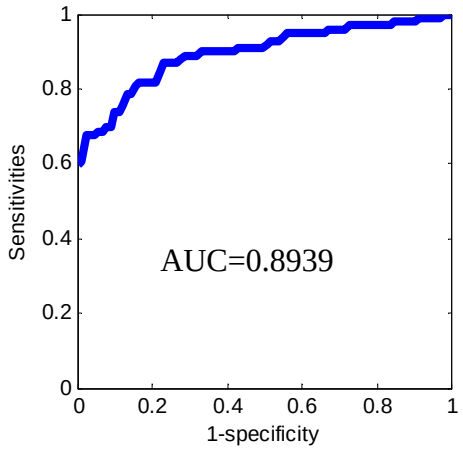


Figure 2.13 Receive-operating characteristic curve

The above figure illustrates the Receiver-Operating Characteristic (ROC) curve of the logistic regression classifier applied to the Alzheimer's dataset. The corresponding accuracy is 0.8158 and the area under the curve is 0.8939.

The ROC curve normally plots sensitivities along the vertical axis and 1-specificity along the horizontal axis. A random classification should yield a 45 degrees line from bottom left to the up right with an AUC of 0.5. A system that is better than random should have its ROC curve above the 45 degrees line with an AUC greater than 0.5.

2.5.3 Model selection

Good generalization performance involves a balance between model complexity and training accuracy. A complex model, such as one with little regularization and

many parameters, will yield low training errors. In the extreme case, the number of input features or basis functions may equal or exceed the number of training samples, and the model will explain the training data perfectly. Such a model is unlikely to make accurate predictions. To avoid overfitting, it is usually suggested to use models with less complexity. People often refer to Occam's razor (Domingos, 1998) to support the idea preferring simpler models[2]. However, Occam's razor did not specifically state sophisticated models should be avoided, but rather people should prefer the simpler model than the complex model when both have the same generalization performance. If a more complex model achieve better generalization performance than a simpler one, people should favour the model with better performance. In practice, cross-validation can provide empirical estimation of the generalization performance, but when the model has multiple complexity parameters, cross validating all the combinations of settings may be impractical. Therefore, it is necessary to find a measure of performance which depends on the training set only. One of the famous information criteria is the Akaike information criterion (AIC) (Akaike, 1974). It simply measures model complexity by the number of parameters and penalize it from the maximum likelihood estimates of the model.

$$\mathrm{AIC} = \ln p(D \mid \theta_{ML}) - M \tag{2.67}$$

where $\ln p(D \mid \theta_{ML})$ is the maximum log likelihood of the model, and M is the number of adjustable parameters. For regression, M could be the number of input features. It is often combined with principal component analysis (PCA) to orthogonalize the input features and rank them based on their contribution to total variance. Model selection based on AIC will try to achieve a compromise between the fitting of the regression and the number of principal components (Brickman et al., 2007). Another similar

[2] The original quote states "Nunquam ponenda est pluralitas sin necessitate", which is translated into "Entities should not be multiplied beyond necessity"

criterion is the Bayesian information Criterion (BIC), which penalizes model complexity more heavily

$$\mathrm{BIC} = \ln p(D \mid \theta_{ML}) - \frac{1}{2} M \ln N \tag{2.68}$$

Here the N is the number of samples.

We can also take the Bayesian approach, of which algorithms in the later chapters are based, to marginalize over the parameters and obtain the evidence function (2.5). Recall the Bayesian view on ridge regression in section 2.4.2. Because both the prior and the likelihood function are model by Gaussians, it is possible to find the analytic form of the marginal likelihood function

$$\begin{aligned} p(\mathbf{t} \mid \alpha, \sigma^2) &= \int p(\mathbf{t} \mid \mathbf{w}, \sigma^2) p(\mathbf{w} \mid \alpha) d\mathbf{w} \\ &= N(\mathbf{t} \mid 0, \mathbf{C}) \end{aligned} \tag{2.69}$$

Here, $\mathbf{C} = \sigma^2 \mathrm{I} + \alpha^{-1} \mathbf{X}\mathbf{X}^{\mathbf{T}}$ is the covariance of the marginal likelihood. Therefore, we can estimate the hyper-parameters by maximizing the evidence function without dividing the data into training and validating sets. More details are in chapter 4 on the topic of Relevance Vector Regression (RVR). For cases when the integration cannot be achieved analytically, approximation methods such as Laplace's method and variational Bayes can be applied (Friston et al., 2007a; Mackay, 1992).

Chapter 3.

Kernel Methods and Kernel Construction from Neuroimaging Data

Contents

This Chapter will describe the foundation of kernel methods, data preprocessing, kernel construction, and basic kernel algorithms. Kernel methods are a specific category of machine learning algorithms. The procedures can be broadly divided into two components: the construction of the kernels and the actual kernel algorithms themselves. The two parts are mostly independent, so kernel algorithms are not constrained to particular data types or input features. The same kernel algorithms can be used with kernels generated from images, but also many other types of data, such as documents, genetic data, etc. One of the advantages of kernel methods is that kernel functions take care of the conversion from raw data into the desirable kernel matrix. When the input features are in a high dimensional space, such as with image data, the kernel algorithms work in the dimensionality of the input kernel. This dimensionality is the number of training samples, rather than the number of dimensions in the original high dimensional samples.

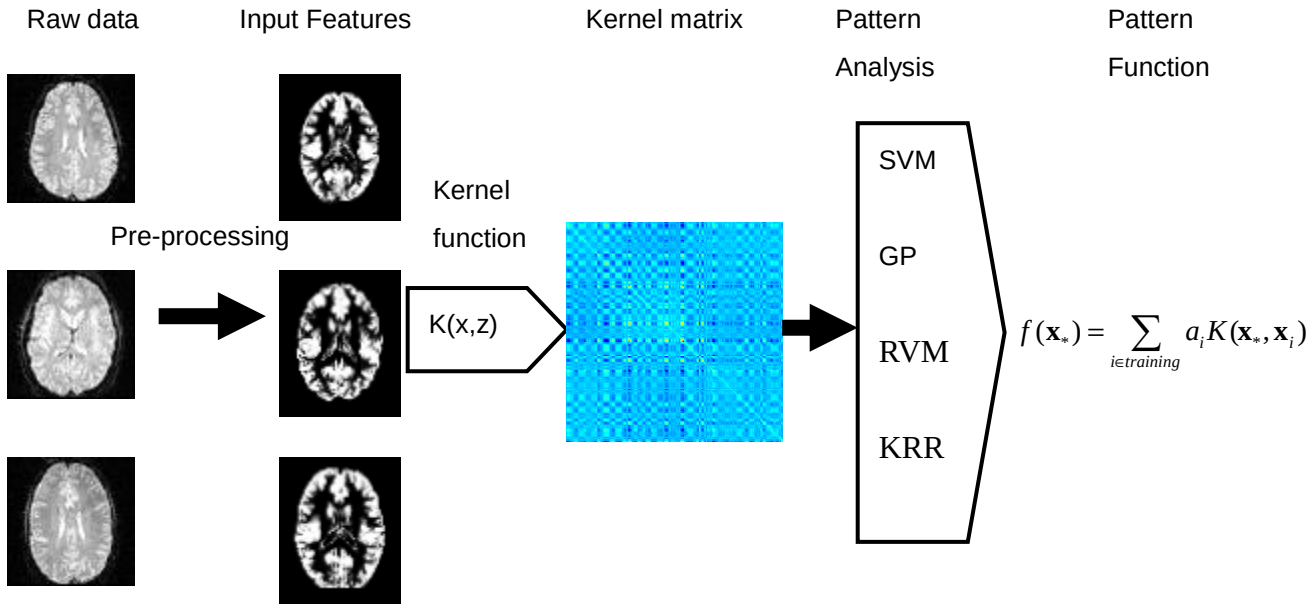


Figure 3.1 The pipeline of kernel methods

The different stages of standard procedures for pattern analysis with kernel methods

The general pipeline for pattern analysis using kernel methods involves five stages:

1. Extract measurements from the observed data, for instance, by converting magnetic resonance images (MRI) into tissue class images.
2. Select the most relevant features for pattern analysis.
3. Choose the desired kernel function to convert the input features into the kernel space.
4. Train the kernel algorithm with the kernel or kernels.
5. Obtain the pattern function and apply the function to predict the new data.

Unlike most of the memoryless algorithms mentioned in chapter 2, kernel methods require the training data to be retained after training. Exceptions to this rule include algorithms that require only a sparse subset of the training data to be retained, or algorithms using linear kernels. The pattern recognition algorithms themselves will be described in chapters 4 and 5, whereas this chapter will concentrate on kernel generation procedures.

3.1 Introduction to Feature Projection and Kernels

Chapter 2 introduced linear methods for classification and regression. Linear methods make predictions using weighted linear combinations of input features. However, data may exhibit non-linear patterns in the input space. To characterize such patterns, kernel methods use the approach of mapping the input space into a higher dimensional feature space, via a mapping function ϕ.

$$\phi : \mathbf{x} \in \Re^D \rightarrow \phi(\mathbf{x}) \in F \subseteq \Re^M \tag{3.1}$$

Nonlinear characterizations in the input space are achieved by linear characterizations in the new space. Theoretically, if the mapped space has equal or higher dimensions

than in number of training samples, the algorithm can find an exact linear fit. Although such solutions may explain the training data exactly, they may not generalize well for making predictions based on new data. A simple example of a mapping is the polynomial mapping function. For a one-dimensional input space $X \subseteq \Re^1$, a third order polynomial mapping will result in the new feature set $\phi : x \to \phi(x) = (x, x^2, x^3) \in \Re^3$. Such polynomial basis functions are often used in least squares regression, to fit non-linear patterns in data.

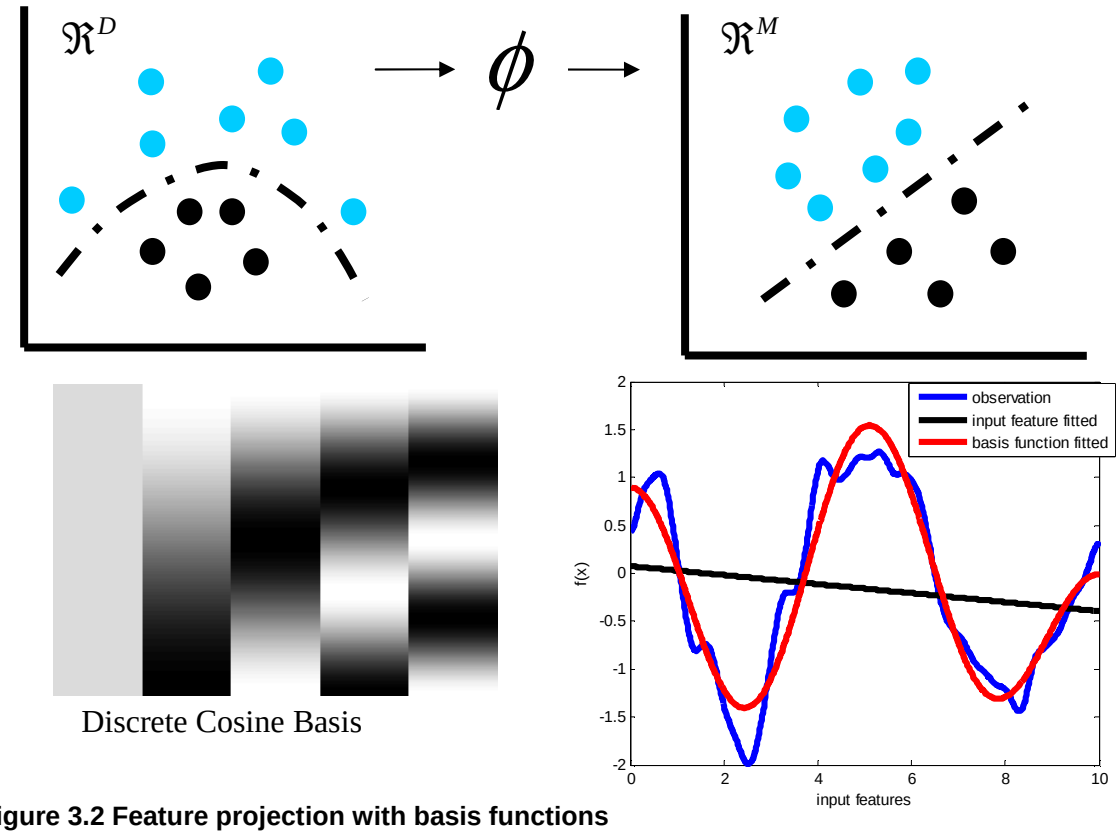


Figure 3.2 Feature projection with basis functions

This figure illustrates possible feature mappings to resolve classification and regression problem for non-linear patterns in the original input space. The top row shows how a non-linear decision boundary, between two classes in the input space, may appear linear after mapping to a new space. For illustration purposes, the data points are projected into a two-dimensional subspace for both input and feature space. The bottom row shows a regression example, using cosine basis functions to fit one-dimensional sinusoidal data. The input features were mapped into a five dimensional space shown at the bottom left. At the bottom right, the black line is the linear fit through the input features. The red line shows the fit from the five dimensional functions.

Given a feature map ϕ, its associated kernel function is $K:\Re^D\times\Re^D\to\Re$ as

$$K(\mathbf{x},\mathbf{z})=<\phi(\mathbf{x}),\phi(\mathbf{z})>,\qquad \mathbf{x},\mathbf{z}\in\Re^D \tag{3.2}$$

Here $<\cdot,\cdot>$ symbolizes the dot-product operation, or the sum of the element-wise products. $<\mathbf{x},\mathbf{z}>=\sum_{i=1}^{D}x_i\,z_i,\quad \mathbf{x},\mathbf{z}\in\Re^D$ .Such a kernel function is symmetric, so that *K*(**x**,**z**)=*K*(**z**,**x**). Notice that the feature map can also be an identity map $\phi:x\to x$. In such cases, the kernel is called a linear kernel and $K(\mathbf{x},\mathbf{z})=<\mathbf{x},\mathbf{z}>=\mathbf{x}^{\mathbf{T}}\mathbf{z}$. For each input feature pair and the corresponding mapping function, there is a unique value determined by the kernel function. However, the feature space defined by the mapping function is not uniquely determined by the kernel function. For example, the feature mapping defined by $\phi:\mathbf{x}=(x_1,x_2)\to\phi(\mathbf{x})=(x_1^2,x_2^2,\sqrt{2}x_1x_2)$ yields the inner product $<\phi(\mathbf{x}),\phi(\mathbf{z})>=x_1^2z_1^2+x_2^2z_2^2+2x_1x_2z_1z_2=<\mathbf{x},\mathbf{z}>^2$, which is identical to the inner product generated from the mapping function defined by $\phi:\mathbf{x}=(x_1,x_2)\to\phi(\mathbf{x})=(x_1^2,x_2^2,x_1x_2,x_2x_1)$. Strictly defined, the kernel is in a reproducing kernel Hilbert space (RKHS), which is an inner product space with additional properties. For mathematical details, please refer to the textbooks (Schlkopf and Smola, 2001; Shawe-Taylor and Cristianini, 2004). Linear kernels were mostly used for this thesis, with only limited exploration of non-linear kernels. For simplification, the thesis will refer to the kernel in the inner produce space.

We define the input matrix **X**, where $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\cdots\mathbf{x}_N]^T$, and each row of **X** is one vector of input features with *D* elements. We also define the feature matrix $\mathbf{\Phi}$ from a particular mapping function ϕ, where $\phi(\mathbf{X})=\mathbf{\Phi}=[\boldsymbol{\varphi}_1,\boldsymbol{\varphi}_2,\cdots\boldsymbol{\varphi}_N]^{\mathbf{T}}$. Each row of $\mathbf{\Phi}$ is a vector of mapped features with *M* elements. The gram matrix, or kernel matrix, is defined as the *N* by *N* matrix **K,** whose entries are $\mathbf{K}_{ij}=<\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)>$ and $\mathbf{K}=\mathbf{\Phi}\mathbf{\Phi}^{\mathbf{T}}$.

The kernel matrix is a symmetric positive-semi-definite matrix, which means that for any non-zero vector **x**, $\mathbf{x}^{\mathbf{T}}\mathbf{K}\mathbf{x} \geq 0$. This implies that all the eigenvalue of the matrix are non-negative.

Intuitively, the kernel matrix can be conceptualized as a matrix of similarity measures between each pair of input training points. The kernel contains all the information available about the relative positions of the inputs in the feature space. In other words, if we rotate and translate the data points in the feature space, the information contained in the kernel matrix will not change, although the values of the kernel matrix may change. Most learning algorithms use only information about relative positions. It should be noted that although most kernel algorithms will maintain identical solutions after rotating and translating the data points in feature space, some of them (such as Gaussian process models) may not.

3.1.1 Dual representation

Many linear classification or regression algorithms can be formulated into either primal or dual forms. In the primal form, we seek the linear weights for each feature, whereas in the dual form, we try to find the weights for each training point. Both weights are interchangeable. The commonly used example to illustrate the dual representation is ridge regression (Bishop, 2006b; Shawe-Taylor and Cristianini, 2004). The primal form was described in section 2.4.2, where we derived the primal weights $\mathbf{w} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{t}$. Alternatively, we can rewrite the second line in equation (2.44) to obtain $\mathbf{w} = \lambda^{-1}\mathbf{X}^{\mathbf{T}}(\mathbf{t} - \mathbf{X}\mathbf{w}) = \mathbf{X}^{\mathbf{T}}\mathbf{a}$. This shows that **w** can be written as a linear combination of the training points, $\mathbf{w} = \sum_{i=1}^{N} a_i \mathbf{x}_\mathbf{i}$, with $\mathbf{a} = \lambda^{-1}(\mathbf{t} - \mathbf{X}\mathbf{w})$. By substituting **w** into this new dual representation, it can be shown that

$$\begin{aligned} \lambda \mathbf{a} &= (\mathbf{t} - \mathbf{X}\mathbf{X}^{\mathbf{T}}\mathbf{a}) \\ (\mathbf{X}\mathbf{X}^{\mathbf{T}} + \lambda \mathbf{I})\mathbf{a} &= \mathbf{t} \\ \mathbf{a} &= (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{t} \end{aligned} \tag{3.3}$$

Here $\mathbf{K} = \mathbf{X}\mathbf{X}^{\mathbf{T}}$ is the linear kernel matrix, mentioned in an earlier section of this chapter. This formulation makes the computation much easier when the input features are high dimensional, as **K** is only *N* by *N*. Ridge regression can be extended into a high dimensional feature space by apply a mapping function ϕ. Hence this formulation is also called “kernel ridge regression” (KRR).

To make a prediction (*y*) from a new data point

$$y = \mathbf{w}^{\mathbf{T}}\mathbf{x}_* = \sum_{i=1}^{N} a_i K(\mathbf{x}_i, \mathbf{x}_*) = \mathbf{a}^{\mathbf{T}}\mathbf{k}_* \tag{3.4}$$

The vector $\mathbf{k}_* = [K(\mathbf{x}_1, \mathbf{x}_*), K(\mathbf{x}_2, \mathbf{x}_*), \cdots K(\mathbf{x}_N, \mathbf{x}_*)]^{\mathbf{T}}$ is the kernel of the new input point $\mathbf{x}_*$ with all the training points in the training set. In the dual formulation, the algorithms do not need the input features or the mapped features. Only the kernel is needed, which describes the relative positions within the feature space. This can be an advantages of kernel methods when the input space or feature space is very large *D*>>*N* or *M*>>*N*. Utilizing the kernel formulation also implies that it is not necessary to compute the features mapped by the function ϕ. Because only the kernel function is needed, it is even possible to use mapping functions with infinite dimensions.

3.1.2 Constructing kernels

The beginning of this chapter showed the construction of a kernel matrix by a pair-wise dot product $\mathbf{K}=\mathbf{\Phi}\mathbf{\Phi}^{\mathbf{T}}$. In practice, it is often not necessary to compute the mapped feature matrix, because the new kernel matrix can be computed from a linear kernel. There are a number of rules to describe the construction of valid kernel matrices. Here, only those rules are listed that are most relevant to this work.

A kernel, $K_1(\mathbf{x}_1, \mathbf{x}_2)$, scaled by a positive constant is also a valid kernel.

$$K(\mathbf{x}_1, \mathbf{x}_2) = cK_1(\mathbf{x}_1, \mathbf{x}_2), \qquad c > 0 \tag{3.5}$$

The sum of two valid kernels, $K_1(\mathbf{x}_1, \mathbf{x}_2)$ and $K_2(\mathbf{x}_1, \mathbf{x}_2)$, is also a valid kernel.

$$K(\mathbf{x}_1, \mathbf{x}_2) = K_1(\mathbf{x}_1, \mathbf{x}_2) + K_2(\mathbf{x}_1, \mathbf{x}_2) \tag{3.6}$$

Combining both rules (3.5) and (3.6), it can be shown that a positive linear combination of valid kernels is also a valid kernel. The element-wise product of valid kernels is also a valid kernel

$$K(\mathbf{x}_1, \mathbf{x}_2) = K_1(\mathbf{x}_1, \mathbf{x}_2)K_2(\mathbf{x}_1, \mathbf{x}_2) \tag{3.7}$$

The polynomial kernel can be derived by applying the above rule (3.7)

$$K_{poly}(\mathbf{x}_i, \mathbf{x}_j) = < \mathbf{x}_i, \mathbf{x}_j >^d \tag{3.8}$$

The polynomial kernel can be further generalized to include a non-negative constant *c*

$$K_{poly}(\mathbf{x}_i, \mathbf{x}_j) = (< \mathbf{x}_i, \mathbf{x}_j > + c)^d \tag{3.9}$$

Expansion of the kernel (3.9) using the binomial theorem gives $K_{poly}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i=0}^{d} \binom{d}{i} c^{d-i} < \mathbf{x}_i, \mathbf{x}_j >^i$. The constant term works as a control of the relative weighting of the different degree monomials. Increasing *c* decreases the relative weighting of the higher order polynomials.

A very popular non-linear kernel is the radial basis function (RBF) kernel, which is sometimes called the squared exponential kernel.

$$\begin{aligned} &K_{rbf}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \| \mathbf{x}_i - \mathbf{x}_j \|^2) \\ &= \exp\{-\gamma(< \mathbf{x}_i, \mathbf{x}_i > -2 < \mathbf{x}_i, \mathbf{x}_j > + < \mathbf{x}_j, \mathbf{x}_j >)\} \\ &= \exp(-\gamma < \mathbf{x}_i, \mathbf{x}_i >) \exp(2\gamma < \mathbf{x}_i, \mathbf{x}_j >) \exp(-\gamma < \mathbf{x}_j, \mathbf{x}_j >) \end{aligned} \tag{3.10}$$

Recalling the Taylor expansion of the exponential function, $\exp(x) = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$, we see that the RBF kernel is a valid kernel with infinite features.

Another commonly used kernel is the normalized kernel, also known as the Cosine

similarity.

$$K_{norm}(\mathbf{x}_i, \mathbf{x}_j) = \frac{< \mathbf{x}_i, \mathbf{x}_j >}{\sqrt{< \mathbf{x}_i, \mathbf{x}_i >< \mathbf{x}_j, \mathbf{x}_j >}} \tag{3.11}$$

Because the normalization in (3.11) depends on the origin in the feature space, we often centre the data and set the origin in the feature space to the mean of the training set. By defining an N element column vector of ones, $\mathbf{l}$, the centred kernel matrix can be computed by[3]

$$\mathbf{K}_{centered} = \mathbf{K} - \frac{1}{N}\mathbf{l}\mathbf{l}^{\mathbf{T}}\mathbf{K} - \frac{1}{N}\mathbf{K}\mathbf{l}\mathbf{l}^{\mathbf{T}} + \frac{1}{N^2}(\mathbf{l}^{\mathbf{T}}\mathbf{K}\mathbf{l})\mathbf{l}\mathbf{l}^{\mathbf{T}} \tag{3.12}$$

If we recall the definition of the covariance matrix of the features in section 2.1.4, the centred kernel can also be viewed as a covariance matrix of the data points. Hence, a centred and normalized kernel is also the correlation matrix between data points of a particular feature space. For the above non-linear kernel, centring, and normalization can be done without directly computing the data in the feature space. All of these new kernels can be computed from linear kernels generated from dot product pairs of training data points. However, there are a number of other popular non-linear kernels, which are not mentioned or applied in this thesis, and not all of them can be computed directly from linear kernels. So far, most investigators have used either linear or RBF kernels for neuroimaging data (Fan et al., 2007a; Fan et al., 2008b).

3.2 Pre-processing and Generating Kernels from Imaging Data

The previous section introduced theoretical aspects of kernel methods and some mechanisms to compute non-linear kernels from linear kernels. This section will

[3] This will be generalized later, by using a residual forming matrix.

describe practical aspects of generating kernels, from either structural or functional MRI data. From Ugly Duckling theorem (Watanabe, 1970), prior knowledge is still required to define the similarity measure. With background knowledge about the task of pattern recognition, for instance, discriminating patients with Alzheimer's disease and normal controls, one may be in favour of kernel generated from grey matter density map than kernel generated from raw T1-image.

3.2.1 Data pre-processing for structural MRI data

Briefly speaking, MRI techniques utilize the properties of the nuclear spin of protons in water molecules, to create contrast among different body tissues (McRobbie et al., 2007). A variety of sequences of radiofrequency pulses and magnetic field gradients, produced in the MR machine, make it is possible to create different types of images with different tissue contrasts. Structural (or anatomical) MRI scans are often acquired using T1-weighed sequences, but it is important to note that many pulse sequences can also be used to image brain structure. T1-weighted sequences have relatively short TR (repetition time) and short TE (echo time), and are often used to image brain structure because they give reasonable separation between the intensities of grey and white matter (GM and WM), as well as between grey matter and cerebro-spinal fluid (CSF). T2-weighted images are often used to detect brain lesions. To most users, the main distinction between T1-weighted and T2-weighted images are their intensity distributions for GM, WM and CSF. In T1-weighted images, the intensity of CSF is less than that of GM, and GM is less intense than WM. In T2-weighted images, the order is reversed. The work in this thesis mostly concerns degenerative diseases, so the focus is on GM.

Although images may be acquired using the same modality, each sequence and machine may still have some variations. This may result in different baseline intensities and intensity scaling of the same tissues in the raw MRI data. Therefore, it

may not be feasible to use the raw MRI intensities as input features for the machine learning algorithms. A more reliable measure, which is invariant to the intensity distribution given the same tissue type, should be used. The procedure called “tissue classification” or “segmentation” is applied to generate tissue class images from the original scans.

Besides variability among the intensity distributions of tissues, different brains have different shapes and sizes. Theoretically, given a large training set containing all possible variations of patient and control brains, we could still learn the pattern of difference; with minimum noise induced by inter subject variability. This would require a nonlinear kernel to accurately encode the complicated shape variability that may be encountered. In practice, available datasets tend to be rather small, so it is necessary to reduce inter subject variability and increase the with-in group similarity. To achieve this goal, images would be warped into a standard space. This procedure is often called “spatial normalisation”, and has the effect of modelling out much of the shape variability.

There are many publicly available packages to do segmentation and spatial normalisation, but the current work uses SPM5 (http://www.fil.ion.ucl.ac.uk/spm/), which the well-known package developed in our lab. Pre-processing of images prior to using pattern recognition can be done in a similar way to pre-processing for Voxel Based Morphometry (VBM) (Ashburner and Friston, 2000, 2001).

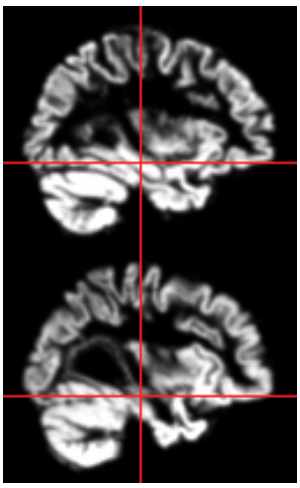
The first stage of the pre-processing is to segment the original MRI data into tissue class images of GM and WM. In these tissue class images, the voxel values range between 0 and 1 to represent the probability of a voxel belonging to a particular class. In principle, the “unified segmentation” function in SPM5 (Ashburner and Friston, 2005) works for different modalities, so it is also possible to segment T2-weighted or proton density-weighted images. T1-weighted images normally give a

reasonable estimation of the GM tissue map. The new segmentation routine (which is part of SPM8) could also be used to achieve multi-channel segmentation. Segmentation in SPM is based on a mixture of Gaussians clustering method (introduced in section 2.2.3), which is guided by tissue probability maps representing the prior probability of encountering various tissue types at each voxel. The algorithm incorporates some nonlinear registration so that the tissue probability maps can be overlaid, and also models out a smooth intensity inhomogeneity artefact.

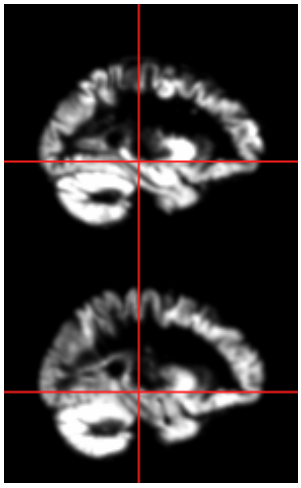
Because registration is built into the segmentation model, the output of the routine can include spatially normalized versions of the tissue class images. To preserve the tissue volumes, an additional scaling (by the Jacobian determinants of the nonlinear deformation) is applied to the normalized images (Davatzikos et al., 2001; Good et al., 2001a; Good et al., 2001b). This is colloquially known as “modulation”. The integral of the values from a GM tissue class image in the native space, should equal the integral of the modulated normalized image. Theoretically, if the registration was perfect, all the spatially normalized images would be identical without the modulation. In practice however, because there is regularization imposed on the registration, the normalization is not perfect and the residual differences between the non-modulated normalized image and the template are a function of the regularization parameters. From the pattern recognition perspective, we could use either the Jacobian determinants or the residuals as input features for kernel constructions. However, using Jacobian determinants as the input features often yields similar results to those obtained using the modulated images.

One slight drawback is that the nonlinear deformations in the segmentation routine are based on a model with only about 1,000 parameters. Another is that only a small-deformation approximation is used (Ashburner and Friston, 1999), so the registration is only approximately invertible. The template space (MNI space) (Chau

and McIntosh, 2005; Evans et al., 1993), is also known to be slightly larger than brains from the general population (Lancaster et al., 2007). To decrease the within group variability, an iterative template generating method was used (Ashburner and Friston, 2008) from the “DARTEL” (Ashburner, 2007) toolbox of SPM5/SPM8. The processing works as follows. Firstly, the segmented GM and WM tissue class images of each subject in the native space are rigidly aligned using a Procrustes method. Initial template data are generated by averaging the GM over all subjects, and doing the same for the white matter. The individual GM and WM maps are then simultaneously registered with their corresponding templates, and their respective weighted averages are recomputed. This iterative warping and averaging procedure is repeated 18 times. The regularization of the warping is reduced slightly at each iteration. The outputs of this procedure are the population templates of GM and WM and the deformation parameters of each individual to this template. The deformation parameters are used to generate the modulated and normalized images, which serve as features for the subsequent pattern recognition.



Small field deformation in MNI space (conventional SPM normalization)



Large field deformation in population space (DARTEL normalization

Figure 3.3 Normalised brain by conventional SPM and DARTEL

The above figures show the results of two subjects with two different normalization methods. The left figure shows the normalization using about a thousand discrete cosine transform basis functions to parameterize the warp to the MNI template. The right figure shows the spatial normalization using the DARTEL toolbox, with iterative registration and template generation. The conventional SPM normalization clearly resulted in larger brains than the DARTEL normalization due to the larger MNI template. The DARTEL normalization also performed more accurate registration. The hippocampus shown at the bottom left of the figure is clearly mis-registered

There are a number of other outputs generated by the DARTEL toolbox, which could also be used as data for generating kernel matrices. Each of them conveys slightly different information about the shape and local density. The Jacobian determinants of the nonlinear spatial transforms may reflect most information on global and local shape differences. Alternatively, the parameterisations of the deformations could be used as features. There is not yet an established strategy to determine which of those feature sets are most salient to tissue degeneration in the context of multivariate pattern recognition. The “Ugly Duckling” Theorem explains why prior knowledge of the data should really be used to formulate the similarity measures.

The current implementation of the DARTEL toolbox is based on generating diffeomorphic mappings via a constant velocity framework. It is fast because it allows a larger one-to-one mapping to be generating by repeatedly composing a very small deformation with itself, using a scaling and squaring procedure. From a theoretical perspective, a variable velocity approach, such as that used by the Large Deformation Diffeomorphic Metric Mapping (LDDMM) algorithm (Miller, 2004; Qiu et al., 2007; Wang et al., 2007), would be superior. LDDMM tries to minimize the difference between the source and the template images as well as minimize the geodesic distance of the deformation. It can also be shown, through the conservation of momentum, that knowledge of the initial conditions (initial velocity) is sufficient to derive the entire deformation trajectory. Such a "geodesic shooting" method is currently being developed for inclusion within the SPM software. The variable velocity framework has some unique mathematic properties that make it a good metric system. The initial momentum, which can be computed by a linear operator from the initial velocity, can encode the information of the full deformation in a spatially compact fashion (Ma et al., 2008).

The pre-processing pipeline is often finished by a spatial smoothing step. The idea is to suppress higher spatial frequency signal, which is more likely to be uninformative for the pattern recognition (ie noise). Conventionally, people doing VBM analysis convolve their images with Gaussian kernels of between about 8 and 12mm FWHM. The aim is to reduce the errors induced by mis-registration, and also to satisfy the assumptions of Random Field Theory. After the more accurate inter-subject alignment of DARTEL, the data would typically be convolved with Gaussians of 6mm FWHM or less. For pattern recognition, the optimal amount of smoothing is an empirical problem, and requires cross-validation to justify.

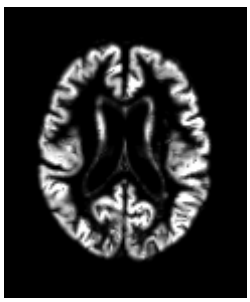
To generate a linear kernel matrix of the cohort, we simply treat each image as a

long one dimensional vector, and compute dot products between each pair of images. As when fitting a mass-univariate general linear model (GLM) through the data, it is sometimes desirable to remove some of the confounding inter-subject variability that could be explained by variables such as sex, education, gender and a constant term. We can apply the same method used by the GLM to remove the linear effects of the confounding factors at each voxel across subjects. Here, we take the same definition of $\mathbf{X}$ as in the previous section, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_N]^T$ and each row of $\mathbf{X}$ is one vector of input features with *D* voxels. We can define an *N* by *K* matrix of confounds, $\mathbf{C}$, where each column is one covariate to remove from the data, and *K* is the number of covariates. From the general equation of ordinary least squares (2.38), we can compute the contribution of each confound at each voxel by $\mathbf{W} = (\mathbf{C}^\mathbf{T}\mathbf{C})^{-1}\mathbf{C}^\mathbf{T}\mathbf{X}$, where $\mathbf{W}$ is a *K* by *D* matrix. The input data with the confounds removed is computed by

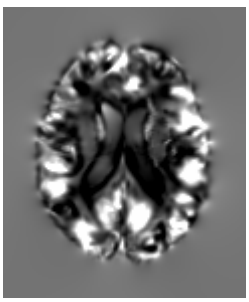
$$\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{C}\mathbf{W} = \mathbf{X} - \mathbf{C}(\mathbf{C}^\mathbf{T}\mathbf{C})^{-1}\mathbf{C}^\mathbf{T}\mathbf{X} = (\mathbf{I} - \mathbf{C}\mathbf{C}^+)\mathbf{X} \tag{3.12}$$

where $\mathbf{C}^+ = (\mathbf{C}^\mathbf{T}\mathbf{C})^{-1}\mathbf{C}^\mathbf{T}$ is the pseudo-inverse. We often define a residual forming matrix as

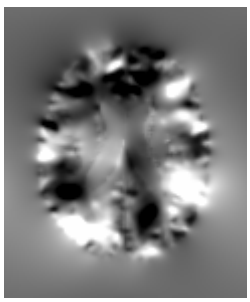
$$\mathbf{R} = (\mathbf{I} - \mathbf{C}\mathbf{C}^+) \tag{3.13}$$



Normalised modulated grey matter map



Jacobian determinants of the deformation



Velocity field (3D vector field) which parameterize the deformation. Only the x component is shown.

Figure 3.4 Output images from DARTEL

The three images show different maps generated from registration by the DARTEL toolbox. Each of these maps encodes the information about shape and GM density in a different way.

Recalling that the linear kernel is calculated as $\mathbf{K}=\mathbf{X}\mathbf{X}^{\mathbf{T}}$, a linear kernel from data with confounds removed can be computed by

$$\tilde{\mathbf{K}}=\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathbf{T}}=\mathbf{R}\mathbf{X}\mathbf{X}^{\mathbf{T}}\mathbf{R}^{\mathbf{T}}=\mathbf{R}\mathbf{K}\mathbf{R}^{\mathbf{T}} \quad (3.14)$$

This shows that it is not necessary to directly remove the confounding covariates at each voxel, especially when the images having millions of voxels. We can simply compute a kernel using the original data, and then factor out the confounds from it. This proves to be computationally efficient, and also adds flexibility when we may want to remove different numbers of covariates. In addition, centering the kernel using (3.12) can also be achieved by factoring out a covariate consisting in a column of ones, $\mathbf{C}=[1,...,1]^{\mathbf{T}}$.

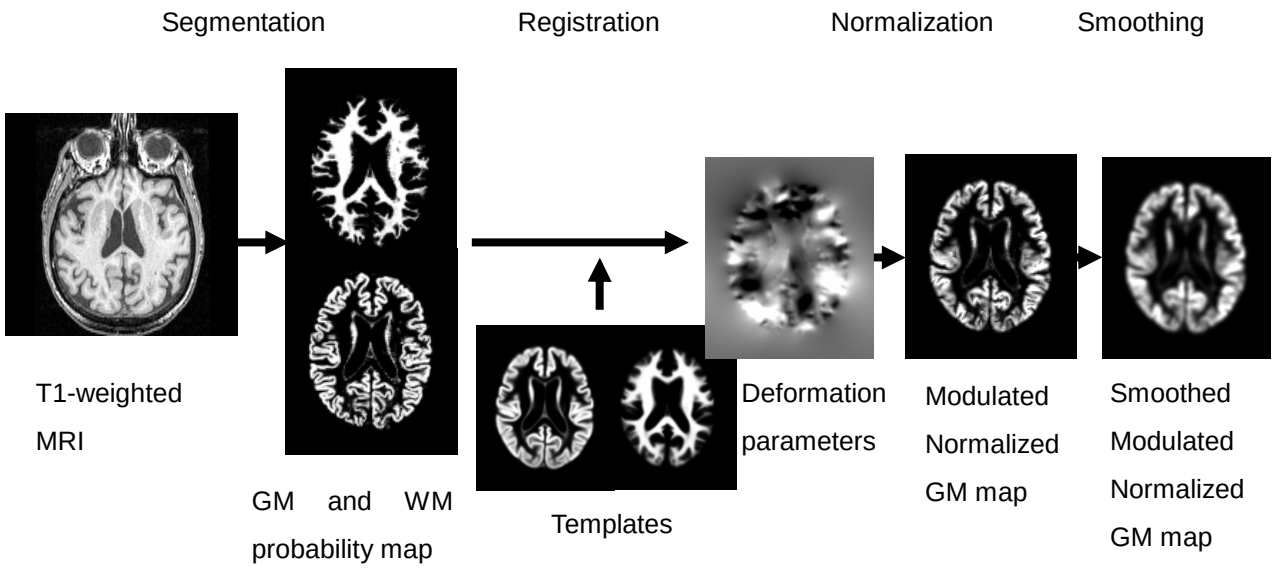


Figure 3.5 Pipeline of structural MRI pre-processing
The pipeline of pre-processing for structural MRI data.

In practice, the size of $\mathbf{X}$ is sometimes beyond the allowable memory. Therefore, kernel construction involves loading only part of the field of view of all subjects' images into memory at a time. Because dot products are linearly additive, the sum of the kernels generated from each part is equivalent to the kernel generated from the full

field of view.

To summarize the pre-processing and kernel generation for structural MRI data, we firstly segment the raw MRI data into tissue class images in the cohort. Secondly, the GM and WM maps are iteratively registered to the population template. The deformation parameters are then used to generate the modulated and normalized GM maps, which are in a standard space, and conserve global GM volumes. Sometimes spatial smoothing is applied to remove image noise and registration error. The most common input features for pattern analysis are the smoothed modulated normalized images. The linear kernel is computed first, and residual forming matrices are later applied to remove the confounding covariates. For non-linear patterns, conversion to an RBF kernel (3.10) or polynomial kernel (3.9) may be done. The final kernel is then used by kernel methods, such as the support vector machine.

3.2.2 Data pre-processing for functional MRI data

Functional imaging generally refers to imaging modalities that are capable of measuring regional neuronal activity, and includes electroencephalography (EEG) and magnetoencephalography (MEG). When referring to functional magnetic resonance imaging (fMRI), people often mean Blood Oxygenation Level Dependent (BOLD) imaging, which is an MR-based non-invasive technique to measure signals related to brain activity. The most recognized theory about the origin of the BOLD signal is based on changes to concentrations of deoxygenated-hemoglobin in the draining veins (McRobbie et al., 2007). When regions in the brain are invoked in cognitive tasks, in order to provide the energy for local action potential and synapse activity, the local blood flow will increase to bring more oxygenated blood (Attwell and Iadecola, 2002; Logothetis, 2008; Logothetis et al., 2001). Because the fully oxygenated blood and

deoxygenated blood have different magnetic susceptibilities, higher concentrations of oxygenated blood will increase the BOLD signals. Physiological constraints also induce delay and dispersion into the measured signals. In other words, a response at the neuronal level does not cause immediate BOLD signal changes, but changes that are often characterized by a hemodynamic response function (HRF). The HRF peaks about 5 seconds after the stimulation, and may reach an undershoot after about 15 seconds. The overall duration of the response function is around 30 seconds, and sometimes a initial dip can be observed (Malonek and Grinvald, 1996), which may be due to initial increase of deoxygenated blood. In SPM, a "canonical HRF" is modelled by two Gamma functions, with seven parameters to control the overall form (Friston et al., 2007c) (figure 3.6). The precise shape of the HRF has been shown to vary across different regions in the brain as well as vary across different people (Aguirre et al., 1998; Schacter et al., 1997). However, for the convenience of modelling, we often use the canonical HRF or sometimes vary the delay of the onset, but maintain the shape of canonical HRF.

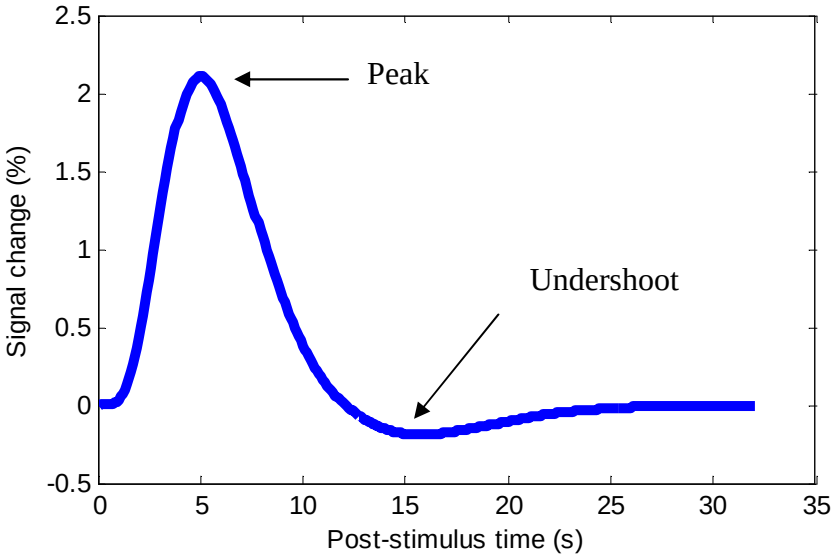


Figure 3.6 Hemodynamic response function

The canonical hemodynamic response function (HRF) of the BOLD signal modelled in SPM5. The delay of the peak is 6 seconds, and the delay of the undershoot is 16 seconds,

Echo planar imaging (EPI) is the usual imaging strategy to measure BOLD signal, as it provides relatively strong signal to noise ratio with short acquisition times. However, EPI suffers from susceptibility-induced image distortion and several artifacts. In the imaging field, the majority of investigators use gradient-echo echo planar imaging (GE-EPI) , which has a T2* weighted contrast (Logothetis, 2008).

In fMRI studies, sequences of EPI are acquired for each subject in a particular experiment. The first stage in pre-processing is realignment and re-sampling of images to remove movement artifacts. This procedure rigidly transforms the images to match the template, which can be the first image in the sequence or the average of the images. Rigid-body transforms in 3D are parameterised by translations in all x, y and z directions and rotation around all x, y and z axes. From the perspective of pattern recognition, the variability arising from rigid body motion lies on a six-dimensional manifold embedded within a space having the same dimensionality as the number of voxels. Therefore, removing motion effects can be seen as a form of dimensionality reduction, and increases the within group similarities.

To further reduce dimensionality, those brain regions that are, a priori, considered non-informative to the pattern recognition should be masked out (or at least down-weighted relative to more informative regions). BOLD signal change is generally believed to occur mainly in grey matter, as its major cause should be the local neuronal activity. Masks defining grey matter can be generated for each subject by segmenting one of the EPIs using (for example) the unified segmentation approach implemented in SPM5. A practical reason for masking out non-grey matter tissue is that it accelerates the speed of kernel generation. By masking out other tissues, only 20% of the whole image is used. It may also have been possible to coregister the anatomical image with the fMRI, and identify grey matter from this. Nevertheless, functional images tend to suffer from spatial distortions, especially in the frontal

region due to the air in the frontal sinus, so it may not have been possible to accurately overlay grey matter masks derived from the anatomical scans.

If the aim is to apply pattern analysis methods across subjects, we will have to spatially normalize the fMRI data to minimize dissimilarity due to inter-subject variation. There are three commonly used ways to spatially normalize fMRI series using the SPM5 software.

1. Match the fMRI data to an EPI template image, by minimising the mean squares difference of the intensities.
2. Coregister the functional images to a structural image of the same subject, and apply the normalization parameters estimated during the unified segmentation routine in SPM5 to the functional data.
3. Coregister the functional image to the structural image, apply DARTEL to the structural images and use the estimated velocity fields mentioned in 3.2.1 to warp the functional data.

There is a less common way to normalize the fMRI data (figure 3.6), which has been shown to perform slightly better in our empirical results of pattern predictions. This is to segment the EPI using unified segmentation, and later apply DARTEL to the tissue class images segmented from the EPI. The estimated deformation fields can then be used to spatially normalise all the functional data.

Usually, when fMRI data are spatially normalised, investigators do not adjust the data to account for the relative expansion or contraction incurred by the warping. However, incorporating such a Jacobian scaling step may prove useful, although we have not yet collected empirical evidence to test this idea.

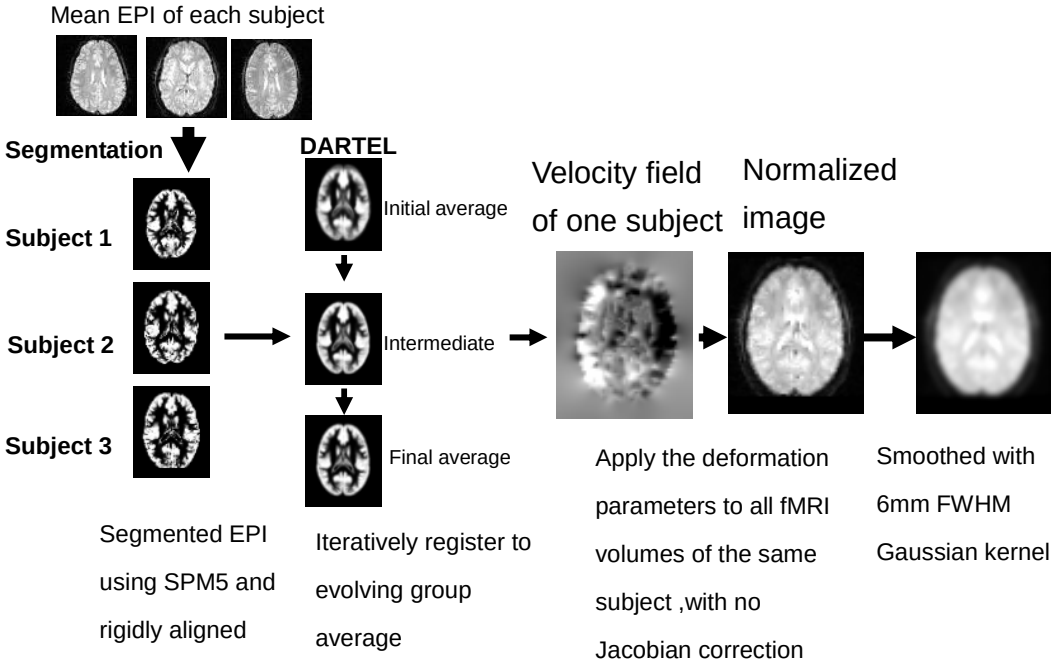


Figure 3.7 Pipeline of spatial normalisation for fMRI data

The pipeline shows the unconventional way to spatially normalize fMRI data. The EPI of each subject are segmented into tissue probability maps. Those maps are used to create the population template using DARTEL toolbox. The normalization parameters are then applied to the original fMRI data.

Signal changes in fMRI that are due to brain activity tend to be slightly lower frequency over space than the much of the noise. From a Wiener filtering perspective, the signal to noise ratio can be increase by spatially smoothing the images. Empirically, we found that accuracy could often be increased by convolving the scans with a 6mm FWHM Gaussian Kernel. Another reason for applying spatial smoothing was to suppress interpolation errors from fMRI time series realignment (Grootoonk et al., 2000).

3.2.3 Temporal modelling for functional MRI data

Low frequency drift has often been reported in fMRI time series. This drift has been attributed to physiological noise or subject motion, but few studies have been done to test this assumption (Smith et al., 1999). The drift models currently dominating fMRI analysis are linear subspaces spanned by a set of polynomial or

discrete cosine transform (DCT) basis functions (Friman et al., 2004; Tanabe et al., 2002). In the context of fMRI decoding, low frequency drift affects the prediction accuracy significantly. The optimal amount of low frequency component to be removed sometimes varies from experiment to experiment, and can only be determined empirically. Often, for event related stimuli with short durations, the cut-off frequency can be set at higher value for the high-pass filtering. In contrast to this, removing only the linear and quadratic drift is often sufficient for block design experiments.

In SPM, the low frequency drift is removed by including DCT basis function as confounding variables in the design matrix, and the default cut-off frequency is 1/128 Hz. DCT is an invertible frequency transform for discrete data. Mathematically, for each voxel v, the time series $\mathbf{v} = \{v_n\}_{n=0}^{N-1}$ is collected from *N* time points and can be transformed into a frequency sequence $\mathbf{f} = \{f_l\}_{l=0}^{N-1}$

$$f_l = \begin{cases} \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} v_n & l = 0 \\ \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} v_n \cos[\frac{\pi}{N}(n+\frac{1}{2})l] & l = 1,..., N-1 \end{cases} \tag{3.15}$$

After pruning the low frequency drift terms (i.e. frequency components less than or equal to a particular number of minimum basis sets, say L) in the original voxel-time series, the detrended sequence $\tilde{\mathbf{v}} = \{v_n\}_{n=0}^{N-1}$ is obtained by the inverse transforms

$$\tilde{v}_n = \sqrt{\frac{2}{N}} \sum_{l=L+1}^{N-1} f_l \cos[\frac{\pi}{N} l(n+\frac{1}{2})] \qquad n = 0,..., N-1 \tag{3.16}$$

Note that the DCT can be represented as a matrix multiplication. Let **G** be the *N* by *L* matrix with $g_{n,l} = \sqrt{\frac{2}{N}} \cos[\frac{\pi}{N}(n+\frac{1}{2})l] \quad l = 1,...L$, $g_{n,1} = \sqrt{\frac{1}{N}}$, where L denotes the number of the minimum DCT basis which are meant to be removed. It can be shown

that the detrending operation is

$$\tilde{\mathbf{v}} = \mathbf{v} - \mathbf{G}(\mathbf{G}^{\mathrm{T}}\mathbf{v}) = (\mathbf{I} - \mathbf{G}\mathbf{G}^{\mathrm{T}})\mathbf{v} = \mathbf{R}\mathbf{v} \tag{3.17}$$

The **R** matrix is the residual forming matrix mentioned in section 3.2.1. Notice in (3.17) we use the transpose of the **G** matrix rather than the pseudo-inverse in the equation (3.13). This is because **G** is an orthonormal basis set $\mathbf{G}^{\mathrm{T}}\mathbf{G} = \mathbf{I}$. Since this procedure is equivalent to removing confounding covariates, we can generalize matrix **G** with any basis functions that model the drift. For example a quadratic basis set will be $\mathbf{G} = \begin{pmatrix} 1 & 1^2 & 1 \\ \vdots & \vdots & \vdots \\ N & N^2 & 1 \end{pmatrix}$ and we can apply (3.13) to calculate the residual forming matrix.

To compute the linear kernel from the fMRI data series, we can take the same definition of **X** as for structural MRI, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_{\mathrm{N}}]^{\mathrm{T}}$ and each row of **X** is one vector of input features with *D* voxels at one time point. Conventionally, we use ascending order for the order of the row i.e. the first row in **X** is the first image in the fMRI time series, and the last row in **X** is the last image in the series. The linear kernel is then computed by $\mathbf{K}=\mathbf{X}\mathbf{X}^{\mathrm{T}}$. Recall equation (3.14), we can apply this equation to detrend the linear kernel and avoid the expensive computation of detrending all the voxels.

Researchers often apply additional pre-processing for block design experiments in fMRI pattern classification, so there would be only one representative image per block. The most common pre-processing strategies include averaging the image volumes over the duration of the block (Cox and Savoy, 2003a; Mourao-Miranda et al., 2005). This can be generalized to the more elegant approach of obtaining the “beta map” (parameter image) for each block, which are the regression parameters from a general linear model (GLM) (Eger et al., 2008; Kriegeskorte et al., 2008).

Both approaches are linear operations, so it is possible to formulate them as matrix operations. In fact, both “average maps” and “beta maps” are a weighted linear combination of the images in the time series. So far, a square residual forming matrix has been described for removing uninteresting signal from the kernel. Such a procedure does not necessarily require the matrix to be a square residual forming matrix. Instead, it could be one for converting a kernel generated from the original data, into a kernel that would be obtained by generating dot products from the parameter images.

Mathematically, we can define a vector of weighting coefficient, **p**, which has the same number of elements as the number of images in the time series. This weighting vector is generated by taking the pseudo-inverse of the regressor in the design matrix of the corresponding block. Usually, the regressor is the HRF convolved block (See figure 3.8) or a boxcar functions with 6 seconds delay after the onset of the stimulus. The 6 seconds delay comes from the delay of the peak of the HRF (figure 3.6). We can also take a more general approach by including the confounding covariates in the design matrix, and take the pseudo-inverse of the design matrix (all the regressors) corresponding to the specific block. The **p** is the transpose of the row of the pseudo-inversed matrix corresponding to the specific beta values (parameters of the regressors) in which we are interested. If every block has the same length, we can use the Kronecker product to generate the “average forming matrix” or “beta map forming matrix” (temporal compressing matrix) by $\mathbf{P} = \mathbf{I} \otimes \mathbf{p}^{\mathbf{T}}$, where **I** is the number of blocks by number of blocks identity matrix. This approach can be extended to event related fMRI as well. If each event is modeled as a separated regressor in the design matrix, the temporal compressing matrix **P** is simply the pseudo inverse of the design matrix. The new data matrix can be evaluated by $\tilde{\mathbf{X}} = \mathbf{PX}$ and the compressed kernel can also be evaluated directly from the original

linear kernel generated from all the image volumes, $\tilde{\mathbf{K}}=\tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\mathbf{T}}=\mathbf{PXX}^{\mathbf{T}}\mathbf{P}^{\mathbf{T}}=\mathbf{PKP}^{\mathbf{T}}$. The dimension of this new kernel will be the number of blocks or events, rather than number of fMRI volumes in the series.

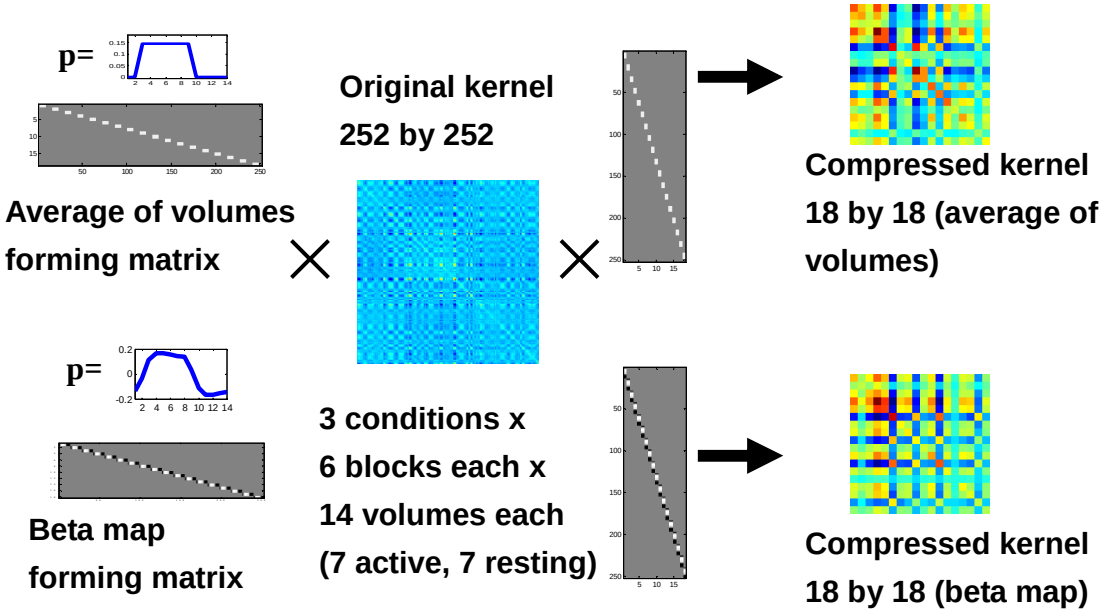


Figure 3.8 Temporal compression using matrix operation

This figure illustrates the matrix operation to compress the original kernel into the reduced kernel whose dimension is the number of blocks. The experiment in the example had a total of 252 image volumes, with 3 different types of stimuli using the block design. Each condition has 6 repeats, and each block contains 14 volumes (7 volumes of active condition, followed by 7 volumes of resting). The averaging operation computes the kernel by averaging the 3rd to the 9th image volumes in each block. And the operation to generate the equivalent kernel from beta maps can be realized as a weighted averaging by the mean removal HRF (see the profile of **p** at the lower left corner in the figure)

There is also another formulation called “spatial-temporal” (Mourao-Miranda et al., 2007)”. In this formulation, images in each block are concatenated into one long vector, hence the input features contain both spatial and temporal information, i.e. the temporal information is not averaged. Unfortunately, this formulation cannot be arranged into the same matrix operation. Therefore, we express each element in the condensed kernel as a sum of weighted kernel elements in the original kernel (figure

3.9)

$$\tilde{k}_{kl} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} k_{n(k-1)+i,n(l-1)+j} \tag{3.18}$$

where $\tilde{k}_{kl}$ is the element at row k and column l in the condensed kernel, n is the number of image volumes in each block. **A** is the n by n weighting matrix containing the coefficients a_{kl} for each of the elements in the original kernel. Because the kernel matrix **K** is symmetric, the weighting matrix is also symmetric.

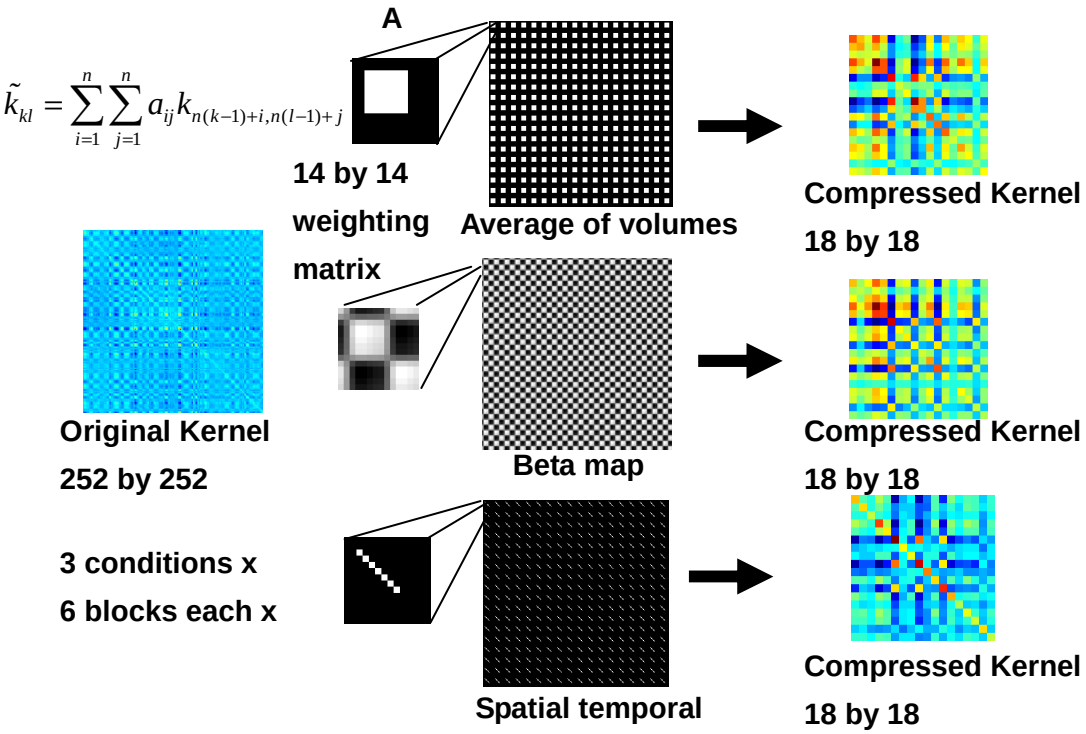


Figure 3.9 Temporal compression using generalised operation

This figure illustrates the generalized operation to compress the original kernel into the reduced kernel whose dimension is the number of blocks. The experiment is the example used in figure 3.8. The new elements in the compressed kernel matrix are the sum of the weighted elements in the original kernel, computed using equation (3.18). The weighting matrix is shown on the second column of the figure. Brighter colours indicates higher values.

The weighting matrix for the beta map can be computed directly from their weighting vector **p**, by $\mathbf{A} = \mathbf{p}\mathbf{p}^{\mathrm{T}}$. For the spatial-temporal operation, the weighting matrix is a partial diagonal matrix, such that $\mathbf{A}_{i,i} = \begin{cases} 0 & i \notin S \\ 1 & i \in S \end{cases}$, where S is the set of images

concatenated in the block, and it is often selected to be the same set as the averaging operation. Generally speaking, the full kernel matrix from the entire time series is often utilized in the kernel regression framework, whereas the condensed kernel matrix is used in classification problems, where the objective is to categorise events.

3.3 Introduction to Basic Kernel Algorithms

Before going into more sophisticated kernel algorithms in chapters 4 and 5, this section introduces some basic algorithms for clustering and classification. Data decomposition methods will also be introduced, such as singular value decomposition (SVD), principal component analysis (PCA), as well as the more general kernel principal component analysis (KPCA).

3.3.1 Singular Value Decomposition and dimensionality reduction

In linear algebra, Singular Value Decomposition (SVD), is a factorization to decompose an *N* by *D* matrix **X,** into an *N* by *N* unitary matrix **U**, an *N* by *D* rectangular diagonal matrix **S**, and a *D* by *D* unitary matrix **V** (Lay, 1997; Moler, 2006).

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}} \tag{3.19}$$

U is often called the matrix of left singular vector vectors, and contains orthonormal basis set $\{\mathbf{u}_1,...,\mathbf{u}_n\}$, so $\mathbf{U}^{\mathbf{T}}\mathbf{U} = \mathbf{I}$. Similarly, **V** is called the matrix of right singular vectors, containing the orthonormal basis sets $\{\mathbf{v}_1,...,\mathbf{v}_d\}$, where $\mathbf{V}^{\mathbf{T}}\mathbf{V} = \mathbf{I}$. The diagonal entries in **S** are the first r singular values of **X**, $s_1 \geq s_2 \geq ... \geq s_r > 0$, where r is the rank of the matrix **X**. If we define the linear kernel as $\mathbf{K}=\mathbf{X}\mathbf{X}^{\mathbf{T}}$, then it can be represented by $\mathbf{K}=\mathbf{U}\mathbf{S}\mathbf{S}^{\mathbf{T}}\mathbf{U}^{\mathbf{T}}$. If we recall the definition of eigenvalue and eigenvector, $\mathbf{X}\mathbf{u}=\lambda\mathbf{u}$, we can realize that the orthonormal vectors $\{\mathbf{u}_1,...,\mathbf{u}_n\}$ are actually the

eigenvectors of the linear kernel, and the square of the singular values $\{s_1^2,...,s_r^2\}$ are the eigenvalues.

Section 3.1.1 mentioned that one advantage of working in the dual form is the computational efficiency when the number of dimension is much higher than the number of samples, $D >> N$. In fact, the data points do not span across the whole feature dimension *D*, but rather span the subspace that is bound by the rank of the data matrix, $r = rank(\mathbf{X}) \le N$. Therefore, we can reduce the data matrix **X**, while still maintaining all the information regarding the relative distance between data points. In other words, we can reduce the *N* by *D* data matrix into an *N* by *N* matrix

$$\tilde{\mathbf{X}} = \mathbf{US} = \mathbf{X}\tilde{\mathbf{V}}, \tilde{\mathbf{V}} = \{\mathbf{v}_1,...,\mathbf{v}_N\} \tag{3.20}$$

This often reduces the computation in the primal form to be the same as that in the dual form. However, this technique of dimensionality reduction still has a shortcoming compared with when dealing with high dimensional data in the dual form. The drawback appears when we want to preserve the full information and maintain a lossless dimensionality reduction while additional data points are added. If we apply SVD on the initial training set, the effective dimension is bounded by the number of training points. When new training points arrive, we can still apply the first *N* right singular vectors to project the new data points into the subspace defined by the original training set. Unfortunately, this will reduce the information carried in the new data. If we want to preserve the information, we would have to apply SVD every time new data arrives. In practice, if the initial training set is sufficiently large, we can be quite confident of obtaining adequate information after dimensionality reduction is applied. Hence we can fix the number of singular vectors to a desirable value.

For memory reasons, it is usually impractical to decompose the huge matrix **X**. To overcome this issue, we can generate the linear kernel **K** by sequentially summing

up kernels computed from subsets of the full data, and then apply SVD to the matrix **K**. The SVD can decompose the matrix into its eigenvectors and eigenvalues, $\mathbf{K}=\mathbf{U}\boldsymbol{\lambda}\mathbf{U}^{\mathbf{T}}, \boldsymbol{\lambda}=\mathbf{S}\mathbf{S}^{\mathbf{T}}, \boldsymbol{\lambda}_{i,i}=\mathbf{S}_{i,i}^{2}$. After obtaining the left singular vectors, and the eigenvalues we can compute successively to find the right singular vectors $\mathbf{v}_i=\mathbf{X}^{\mathbf{T}}\mathbf{u}_i s_i^{-1}, \tilde{\mathbf{V}}=\mathbf{X}^{\mathbf{T}}\mathbf{U}\mathbf{S}^{-1}$. Because it is a linear operation, part of the columns in the data matrix can be sequentially loaded to compute the corresponding elements in the right singular vectors. The reduced matrix of right singular vectors $\tilde{\mathbf{V}}$ is *N* by *N* .

SVD can be applied not only to dimensionality reduction, but also if there is enough redundancy in the data matrix (i.e. only relatively few singular vectors can represent the original data matrix adequately), incremental SVD can be employed to estimate missing entries (Brand, 2002; Kurucz et al., 2007). In the context of imaging data, we can use this method to replace voxels in image volumes containing artifacts, by treating those entries as missing in the full data matrix **X**.

3.3.2 Principal Component Analysis and Kernel Principal Component Analysis

Principal Component Analysis (PCA) is a very popular unsupervised learning method for data visualization, lossy dimensionality reduction, and feature selection. Intuitively, PCA can be understood as a technique to rotate and sometimes flip the data points, without translation and scaling such that the data points in the new coordinate system are orthogonal, i.e. there are no off-diagonal components in the sample covariance matrix of this new coordinate system. This leads to the maximum variance formulation of PCA (Bishop, 2006b; Jolliffe, 2002). To simplify the notation, we assume the means in each dimension of data matrix **X** have been removed, as described in section 2.1.4. We can define a projection unit vector $\mathbf{v}_1$, such that $\mathbf{v}_1^{\mathbf{T}}\mathbf{v}_1=1$.

The projected data points in this one dimensional space are computed by $\mathbf{X}\mathbf{v}_1$, so the variance of the projected data is $\mathbf{v}_1^T\mathbf{X}^T\mathbf{X}\mathbf{v}_1 = \mathbf{v}_1^T\boldsymbol{\Sigma}\mathbf{v}_1$, where $\boldsymbol{\Sigma}$ is the data covariance matrix defined in equation (2.13). The objective will be to maximize $\mathbf{v}_1^T\boldsymbol{\Sigma}\mathbf{v}_1$ with respect to $\mathbf{v}_1$, under the constraint $\mathbf{v}_1^T\mathbf{v}_1 = 1$. To solve this optimization problem, we introduce a Lagrange multiplier λ_1 to convert into the unconstrained optimization of $\mathbf{v}_1^T\boldsymbol{\Sigma}\mathbf{v}_1 + \lambda_1(1-\mathbf{v}_1^T\mathbf{v}_1)$. Setting the derivative with respect to $\mathbf{v}_1$ to zero, and solving, leads to the characteristic equation

$$\boldsymbol{\Sigma}\mathbf{v}_1 = \lambda_1\mathbf{v}_1 \tag{3.21}$$

This means that $\mathbf{v}_1$ and λ_1 are the eigenvector and eigenvalue of the sample covariance matrix. We often define the largest eigenvalue and its corresponding eigenvector as the first principal component. The additional principal components are the rest of the eigenvectors and eigenvalues, ordered such that the eigenvalues are decreasing. Because the covariance matrix is positive definite, the eigenvectors are orthonormal to each other. In fact, PCA can be implemented by SVD. When SVD is applied to the covariance matrix, the singular vectors are the essentially the eigenvectors, and the singular values are the eigenvalues. When the dimensionality is very large, computing the sample covariance matrix is infeasible. We often first generate the linear kernel, and apply equation (3.12) to centre the data, and then apply SVD to the centred kernel. The data projected to the principal components can be evaluated using equation (3.20). Notice that the projected data points will be different if the kernel is not centred. Sometimes, investigators remove the projected components with lower eigenvalues, and retain only those principal components that contribute most (e.g. 96%) of the total variance (Ashburner et al., 1998). A related technique, called Principal Component

Regression (PCR), selects a few principal components with high eigenvalues as the regressors in the model to avoid over-fitting. However, sometimes the low variance components may be important (Jolliffe, 1982). PRC can also be combined with model selection criteria introduced in section 2.5.3. We can use AIC (2.67) or BIC (2.68) to select a desirable number of principal components (Brickman et al., 2007).

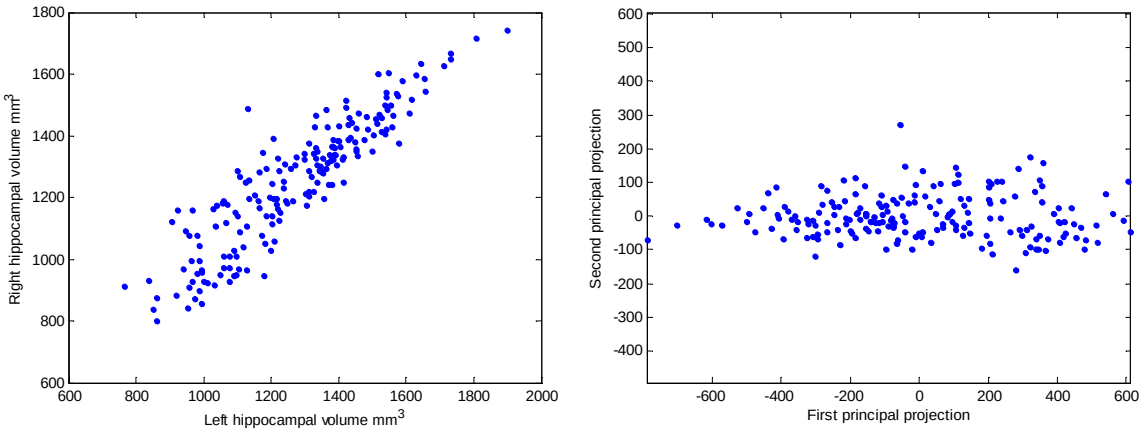


Figure 3.10 Principal component analysis

The left figure is the original plot of right hippocampal volume versus left hippocampal volume, and the right figure is the data project to the first and second principal components. The relative distance between each data points in both figures is the same, however, the data points were mean removed, rotated and flipped.

Because the projected data can be computed by equation (3.20), and the left singular vectors, **U,** computed without needing to work on the original input features, we can apply the kernel trick to project the data into a higher dimensional space. Typically, an RBF kernel function would be used, and the principal components computed in the high dimensional projected feature space. This is called Kernel PCA, and is a technique that may sometimes reveal interesting structure among the data points.

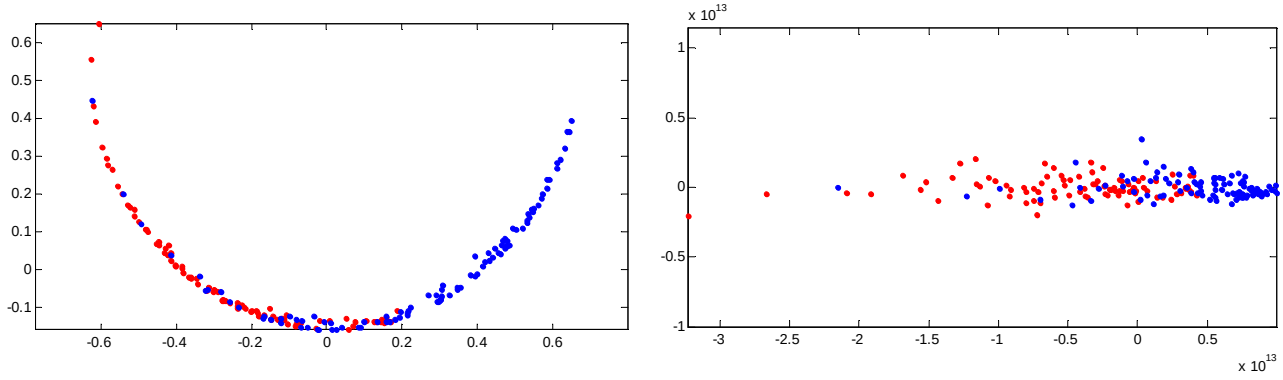


Figure 3.11 Kernel Principal component analysis

The left panel shows the first two principal projections of the implicit feature space of the RBF kernel with the parameter value γ=2.5e-6, using the same data as in figures 2.6 and 3.10 (hippocampal volumes). The circular shape is due to the property of RBF kernels. Notice that the diagonal elements in the RBF kernel are all one, which implies that the data points are projected on to a hyper-sphere. The right panel shows the principal projections of a fifth order polynomial kernel. The red colour indicates controls and blue indicates patients.

3.3.3 Basic kernel algorithms

This section introduces some elementary algorithms, which sometimes assist the visualization or analysis of data. The first would be using kernels to calculate the distance to the group average in the kernel space. To simplify the notation, we replace $\phi(\mathbf{x})$ by $\boldsymbol{\varphi}$ as the features of one data point in the projected feature space. The distance between any two data points can be calculated by

$$\begin{aligned} \|\boldsymbol{\varphi}_i - \boldsymbol{\varphi}_j\|^2 &= <\boldsymbol{\varphi}_i - \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i - \boldsymbol{\varphi}_j> = <\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_i> -2<\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j> + <\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_j> \\ &= K(\mathbf{x}_i, \mathbf{x}_i) - K(\mathbf{x}_j, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) \end{aligned} \tag{3.22}$$

This leads to the equation for computing the distance of any point from the mean of a particular set of samples. $\|\boldsymbol{\varphi}_* - \boldsymbol{\varphi}_m\|^2 = <\boldsymbol{\varphi}_*, \boldsymbol{\varphi}_m> -2<\boldsymbol{\varphi}_*, \boldsymbol{\varphi}_m> + <\boldsymbol{\varphi}_m, \boldsymbol{\varphi}_m>$, where $\boldsymbol{\varphi}_m$ is the centre of mass of the set, and the norm is given by $\|\boldsymbol{\varphi}_m\|^2 = <\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\varphi}_i, \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\varphi}_i> = \frac{1}{N^2}\sum_{j,i=1}^{N} K(\mathbf{x}_i, \mathbf{x}_j)$. Therefore, the distance to the centre of mass is given by

$$\| \boldsymbol{\varphi}_* - \boldsymbol{\varphi}_m \|^2 = K(\mathbf{x}_*, \mathbf{x}_*) + \frac{1}{N^2} \sum_{j,i=1}^{N} K(\mathbf{x}_i, \mathbf{x}_j) - \frac{2}{N} \sum_{i=1}^{N} K(\mathbf{x}_*, \mathbf{x}_i) \qquad (3.23)$$

This is often useful when the aim is to detect outliers in the training groups, and removing those that are more than (say) three standard deviations away from the mean.

Another simple classification algorithm is called K-Nearest Neighbour classification (KNN). In KNN classification, the class of a new data point is determined according to the majority class membership of the K closest training data points (Bishop, 2006b; Duda et al., 2000). For binary classification, K is typically an odd number to avoid equal numbers of neighbours in both classes. Because the distance between any data points can be calculated using (3.22), this algorithm is very easy to implement. Another advantage of this algorithm is that it does not require any training. However, cross validation may be needed to determine a suitable setting for K.

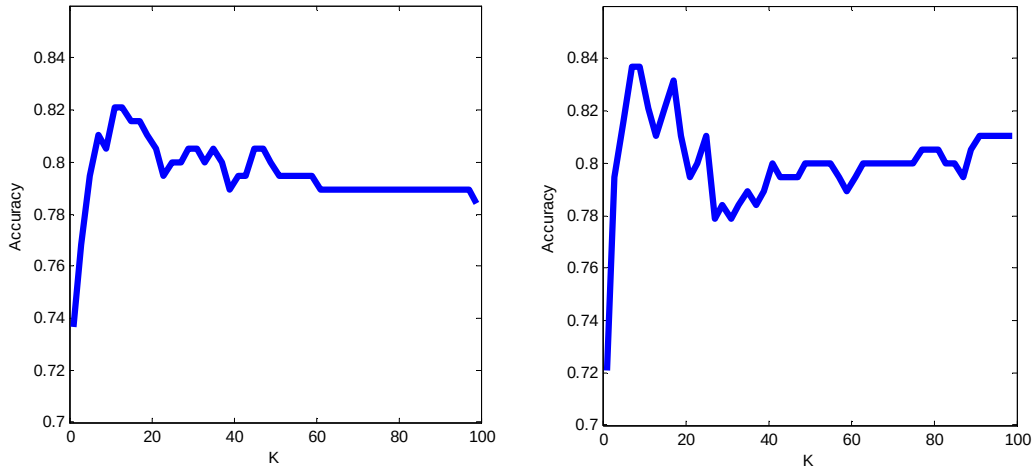


Figure 3.12 K-near neighbour classification

This figure shows the leave one out cross-validation accuracy with different K for the K nearest neighbour classification. The left panel shows the accuracies using a linear kernel generated from the left and right hippocampal volumes, and the right panel shows the accuracies with fifth-order polynomial kernel. The maximum accuracy of 83.68% is achieved with k=7 using the non-linear kernel in this example.

The distance matrix, computed with equation (3.22), can also be used for clustering. The hierarchical clustering method and the dendrogram can reveal hidden structures of similarities among subjects or conditions. For example, if we have several experimental stimuli in an fMRI experiment, we can use hierarchical clustering to group similar conditions together automatically. This may tell us which conditions are similar in terms of their BOLD patterns. Hierarchical clustering works by initially assuming that there are as many clusters as data points. Then each cluster will merge with the nearest cluster until only one cluster remains, or when the minimum number of clusters is reached. The dendrogram, which is the visualization tool for the hierarchical clustering tree, consists of many upsides down U-shape lines connecting different clusters. The height of each reverse U represents the distance between the two clusters being connected (see Figure 3.13).

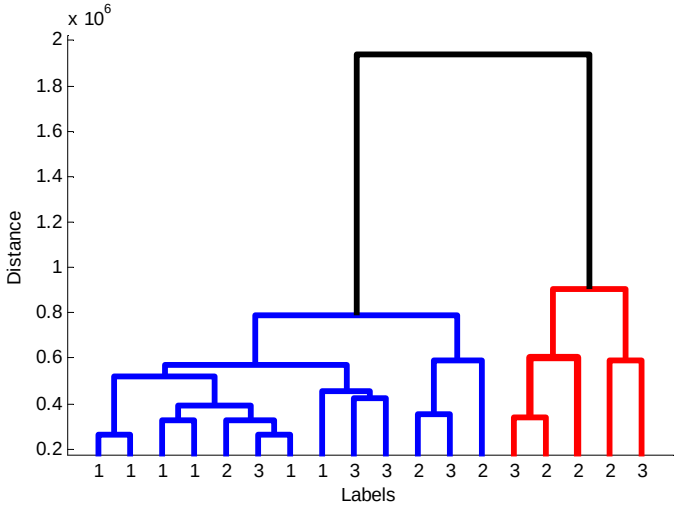


Figure 3.13 Cluster analysis using dendrogram

This is an example of a dendrogram applied to the fMRI experiment mentioned in figure 3.8. We used the volume averaging kernel introduced in section 3.2.2, so each label represents one type of experimental stimulus. Label '1' represents unpleasant stimuli, label '2' represents neutral stimuli, and label '3' represents pleasant stimuli. From this dendrogram, we can see unpleasant stimuli (label 1) are quite distinctive from the neutral stimuli (label 2), and that the unpleasant stimuli seem to be less dispersed.

For the current example, we used the Matlab function 'linkage' and

'dendrogram' in the statistics toolbox. The only required input is the pair-wise distance, which can be computed using equation (3.22). There are various methods to evaluate the distances between clusters. The common ones are:

- 'Single-linkage', which measure the distance between clusters by finding the distance between the closest points in both clusters. i.e. the shortest possible distance between members in cluster one to members in cluster two. $dist(C_1, C_2) = \min(dist(i, j)), i \in C_1, j \in C_2$.
- 'Complete-linkage', which is the opposite of the Single-linkage, finds the furthest points in both clusters. $dist(C_1, C_2) = \max(dist(i, j)), i \in C_1, j \in C_2$.
- 'Average-linkage' evaluates the overall average of the distances between all possible pairs in both clusters. $dist(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{i \in C_1, j \in C_2} (dist(i, j))$.
- 'Centroid-linkage' measures the distance between the centroids of the clusters.

Besides hierarchical clustering, there is a well-known clustering algorithm called "K-means" clustering which is famous for its simplicity. The algorithm is a greedy method; hence it only guarantees to find a local optimum, so the solution may change with different initial estimates. The algorithm works by iteratively evaluating the distance of all data points to the cluster centres, and assigning cluster membership according to which centre is closest. This scheme is iterated until convergence. The free parameter K, which is the number of clusters, has to be specified before running the algorithm. From equation (3.23), we can directly utilize the kernel formulation and calculate the distance to the cluster centres without explicitly evaluating the cluster centres in the feature space. We first initialize an N by K indicator matrix $\mathbf{A}$, which specifies the membership of each data point, so that $\mathbf{A}_{ik} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is in cluster } k \\ 0 & \text{otherwise} \end{cases}$. We can compute the cluster centre by $\mathbf{X}^{\mathbf{T}}\mathbf{AD}$, where

X is the data matrix, and the **D** is *K* by *K* diagonal matrix having the inverse of the column sums of **A** in the diagonal entries. The distance of all data points to the cluster centres can therefore be computed by $\mathbf{l}\, diag(\mathbf{DA}^{\mathrm{T}}\mathbf{KAD})^{\mathrm{T}} - 2\mathbf{KAD}$, where **l** is an *N* element column vector, and diag() indicates the diagonal entries of the matrix. Once the distances to cluster centres has been evaluated, the cluster memberships in **A** can be reassigned based on the shortest distance. This procedure is iterated until convergence, when no element in **A** is reassigned.

Chapter 4

Kernel Regression Methods and their Application in Functional and Structural MRI

Contents

This chapter will introduce the methodological aspects of kernel regression methods, namely Relevance Vector Regression (RVR), Support Vector Regression (SVR), Gaussian Processes Regression (GPR), and the Kernel Ridge Regression (KRR) mentioned in section 3.1.1. Projects employing these methods to both fMRI and structural MRI data will be presented in detail. The fMRI related works are the "Pittsburgh Brain Activity Interpretation Competition 2006 (PBIAC)", in which we achieved 5th place, and the "PBIAC 2007", in which we came first (Carlton Chu et al., 2009; Ni et al., 2008). The project using structural MRI is on "Regression analysis for clinical scores of Alzheimer's Disease using multivariate machine learning method". It is a collaborative work with Drs. Stefan Klöppel and Cynthia Stonnington, using data from Dr. Clifford Jack at the Mayo Clinic. I presented this work as both a poster and an oral presentation at the Organisation for Human Brain Mapping (OHBM) conference in 2007.

The general regression framework was introduced in section 2.4, and assumes that a training set containing input/output pairs, $S = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ..., (\mathbf{x}_N, t_N)\}$, where t is a continuous number $t \subseteq \Re$. The general model for a linear regression problem is $t = \sum_{d=1}^{D} w_d x_d + offset + error$, which means that the output is a weighted linear combination of input features, plus a constant offset and noise. The general model for a kernel regression is $t_* = \sum_{i \in training} \beta_i K(\mathbf{x}_*, \mathbf{x}_i) + offset + error$, which means the output is a weighted linear combination of the kernel generated from the input sample, with all the training samples - plus offset and error. Some models may not include an offset term. Both SVR and RVR are in the category of sparse kernel machines, which implies that some of the kernel weights, β, are zero. In other words, not all training samples contribute to the prediction of the testing samples. One advantage of kernel methods is that the kernel algorithms are invariant to the type of kernels used.

Therefore, all the issues relating to non-linear patterns were previously encapsulated in Chapter 3 (kernel construction). However, if a linear kernel is used, we can obtain the weights in the input feature space by $\mathbf{w} = \sum_{i=1}^{N} \beta_i \mathbf{x_i}$, which can be useful for gaining insight into which features are informative.

4.1 Introduction to Kernel Regression Algorithms

KRR was previously described in section 3.1.1, so no more needs to be said about it here. RVR and GP both use a Bayesian framework. SVR will be introduced first. Usually when describing the Support Vector Machine (SVM), the classification form is introduced first and the regression form afterwards. Classification is directly linked to the fundamental core of SVM (Vapnik, 1998), and is also the most popular form. In fact, SVM is often used to refer only to Support Vector Classification (SVC). SVR is often viewed as a regression model motivated by the philosophy of SVC. Because of the structure of this thesis, SVR is described prior to SVC, but it is advisable to read about SVC first in Chapter 5.

4.1.1 Support Vector Regression

In the SVM framework, the optimisation problem is convex, so the solution it finds is the global optimum. Another feature of SVM is its property of sparseness (Vapnik, 1998). To achieve both sparseness and the global optimum, SVR is motivated by defining a loss function, which ignores errors within a certain range between the predicted and the true (target) values. This is called an ε-insensitive loss function (Bishop, 2006b; Cristianini and Shawe-Taylor, 2000; Schlkopf and Smola, 2001; Smola and Olkopf, 2003). A typical linear loss function is $L_1(\mathbf{x}, t) = |\, y - t \,|, y = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$ and a quadratic loss function is $L_2(\mathbf{x}, t) = (y - t)^2$, where t is target, or true value, and y is the predicted value. The linear ε-insensitive loss is

defined by

$$L_1^{\varepsilon}(\mathbf{x},t) = \max(0,|\, y-t\,| -\varepsilon) \tag{4.1}$$

and the quadratic ε-insensitive loss is defined by

$$L_2^{\varepsilon}(\mathbf{x},t) = (\max(0,|\, y-t\,| -\varepsilon))^2 \tag{4.2}$$

This means that if the conventional linear loss is below a threshold value ε, the loss is ignored. If we adapt the regularised linear regression in (2.42), the objective function for SVR with quadratic ε-insensitive loss is given by

$$\sum_{i=1}^{N} L_2^{\varepsilon}(\mathbf{x}_i, y_i) + \lambda \,\|\, \mathbf{w} \,\|^2 \tag{4.3}$$

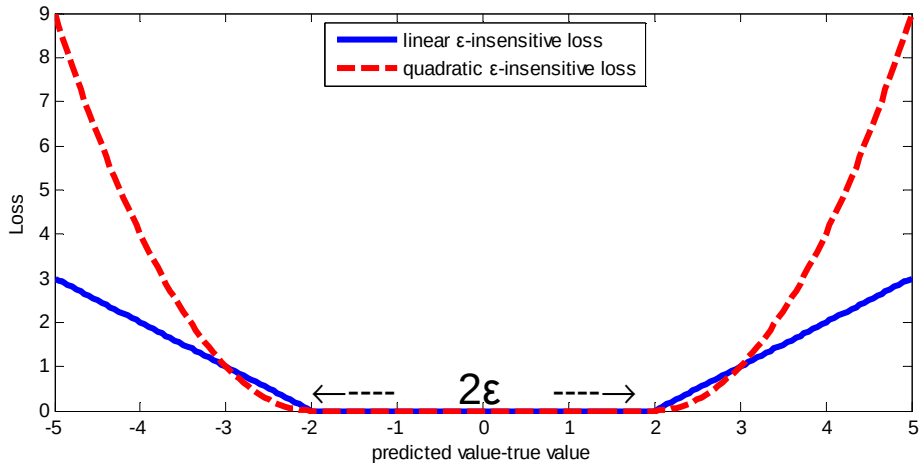


Figure 4.1 e-insensitive loss function

Both linear and quadratic ε-insensitive loss functions with the value of ε set to 2.

This optimisation problem can be reformulated by introducing slack variables. Two slack variables are required for each data point, such that $\xi_i \geq 0$ corresponds to the point where $t_i > \varepsilon + y_i$ (the predicted value is more than ε below the target) and $\hat{\xi}_i \geq 0$ corresponds to point where $t_i < y_i - \varepsilon$ (the predicted value exceeds the target value by more than ε). In addition, the slack variables satisfy the condition that $\hat{\xi}_i \xi_i = 0$. This implies that for each point, either both slack variables are zero (when the data point is within the insensitive zone), or one of the slack variables is

zero. In the one dimensional regression example (figure 4.2), if the data point is outside the insensitive zone, ξ_i indicates the error from the data point to the upper boundary of the insensitive zone, and $\hat{\xi}_i$ indicates the error from the data point to the lower boundary of the insensitive zone.

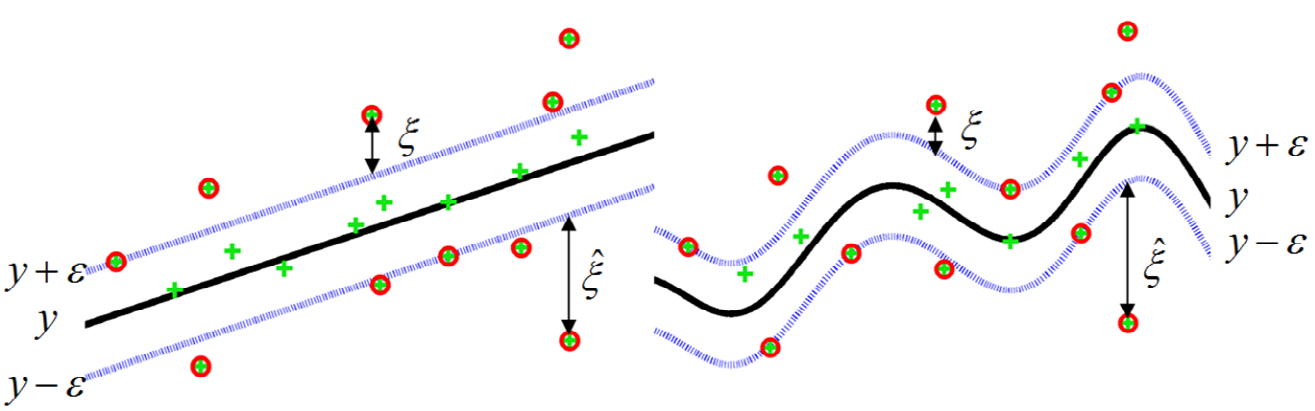


Figure 4.2 1D Support Vector Regression

Illustration of a one dimensional Support Vector Regression problem, with a linear ε-insensitive loss function. The left panel shows the solution with a linear kernel and the right panel shows the result of a RBF kernel with the same data points. The dotted lines indicate the insensitive "tube", defined by the ε parameter. The data points are indicated by crosses, and the circles are the so called "support vectors", which are the data points defining the solution. In other words, they are the data points with non-zero kernel weights. Data points within the insensitive tube do not contribute to the solution.

The primal objective function of the regularised SVR with quadratic ε-insensitive loss function is given by

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i^2 + \hat{\xi}_i^2) \\ \text{subject to} \quad & (\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) - t_i \le \varepsilon + \hat{\xi}_i, i = 1,\ldots,N \\ & t_i - (\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \le \varepsilon + \xi_i, i = 1,\ldots,N \\ & \xi_i, \hat{\xi}_i \ge 0, i = 1,\ldots,N \end{aligned} \tag{4.4}$$

The C and ε settings are often chosen using cross validation. If $\varepsilon = 0$, this objective function is equivalent the ridge regression in (2.42). Notice the difference between the free parameter C and the λ used in conventional ridge regression. In ridge

regression, λ controls the amount of regularisation. In SVR, the regularisation is fixed and *C* controls the amount of penalty from the training errors. Therefore higher *C* indicates less regularisation, and lower C indicates stronger regularisation. The dual problem can be derived by introducing the Lagrange multipliers, $\hat{a}_i, a_i$.

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{N} t_i(a_i - \hat{a}_i) - \varepsilon \sum_{i=1}^{N}(a_i + \hat{a}_i) - \frac{1}{2}\sum_{i,j=1}^{N}(a_i - \hat{a}_i)(a_j - \hat{a}_j)(<\mathbf{x}_i, \mathbf{x}_j> + \frac{1}{C}\delta_{i,j}) \\ &\text{subject to} \quad \sum_{i=1}^{N}(a_i - \hat{a}_i) = 0, \quad \hat{a}_i \geq 0, a_i \geq 0, i = 1, ..., N \end{aligned} \tag{4.5}$$

where $\delta_{i,j}$ is the delta function, having a value of 1 only if *i=j*, and zero otherwise. We can then substitute $\beta = a_i - \hat{a}_i$, and use the relation $\hat{a}_i a_i = 0$, which is inherited from the corresponding slack variables. We can simplify (4.5) to the following

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{N} t_i\beta_i - \varepsilon \sum_{i=1}^{N}|\beta_i| - \frac{1}{2}\sum_{i,j=1}^{N}\beta_i\beta_j(<\mathbf{x}_i, \mathbf{x}_j> + \frac{1}{C}\delta_{i,j}) \\ &\text{subject to} \quad \sum_{i=1}^{N}\beta_i = 0, \quad i = 1, ..., N \end{aligned} \tag{4.6}$$

This is essentially a quadratic programming problem. This equation $\sum_{i,j=1}^{N}\beta_i\beta_j(<\mathbf{x}_i, \mathbf{x}_j> + \frac{1}{C}\delta_{i,j})$, can also be reformulated into the matrix form $\boldsymbol{\beta}^{\mathbf{T}}(\mathbf{K} + \frac{1}{C}\mathbf{I})\boldsymbol{\beta}$, where **K** is the *N* by *N* kernel matrix, and **I** is an identity matrix of the same dimensions. We can see the similarity between this formulation and KRR (3.3): both methods adjust the regularisation by adding diagonal entries to the kernel. After fitting, predictions can be made by

$$f(\mathbf{x}_*) = \sum_{i=1}^{N}\beta_i K(\mathbf{x}_*, \mathbf{x}_i) + b \tag{4.7}$$

where *b* is chosen so that $f(\mathbf{x}_i) - t_i = -\varepsilon - \beta_i / C$ for any *i* that $\beta_i > 0$ or $f(\mathbf{x}_i) - t_i = \varepsilon + \beta_i / C$ for any *i* that $\beta_i < 0$. In practice, *b* is chosen by averaging the solution of b, for data points satisfying $\beta_i \neq 0$. This reduces numerical rounding errors.

Although the quadratic loss function is more closely related to ridge regression, the linear ε-insensitive loss function is more popular for SVR. The primal objective function of the regularised SVR with a quadratic ε-insensitive loss function is given by

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi + \hat{\xi}_i) \\ \text{subject to} \quad & (\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) - t_i \le \varepsilon + \hat{\xi}_i, i = 1,...,N \\ & t_i - (\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) \le \varepsilon + \xi_i, i = 1,...,N \\ & \xi_i, \hat{\xi}_i \ge 0, i = 1,...,N \end{aligned} \tag{4.8}$$

The corresponding dual form can be derived by introducing Lagrange multipliers

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{N} t_i(a_i - \hat{a}_i) - \varepsilon\sum_{i=1}^{N}(a_i + \hat{a}_i) - \frac{1}{2}\sum_{i,j=1}^{N}(a_i - \hat{a}_i)(a_j - \hat{a}_j) < \mathbf{x}_i, \mathbf{x}_j > \\ \text{subject to} \quad & \sum_{i=1}^{N}(a_i - \hat{a}_i) = 0, \quad \hat{a}_i \ge 0, a_i \le C, i = 1,...,N \end{aligned} \tag{4.9}$$

We can also substitute $\beta = a_i - \hat{a}_i$ to derive

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{N} t_i\beta_i - \varepsilon\sum_{i=1}^{N}|\beta_i| - \frac{1}{2}\sum_{i,j=1}^{N}\beta_i\beta_j < \mathbf{x}_i, \mathbf{x}_j > \\ \text{subject to} \quad & \sum_{i=1}^{N}\beta_i = 0, \quad -C \le \beta_i \le C, i = 1,...,N \end{aligned} \tag{4.10}$$

In the linear loss function model, b is chosen so that $f(\mathbf{x}_i) - t_i = -\varepsilon$ for any i that satisfies $0 < \beta_i < C$, or $f(\mathbf{x}_i) - t_i = \varepsilon$ for any i satisfying $-C < \beta_i < 0$. These data points are right on the boundary of the insensitive "tube". The data points with $\beta_i = 0$ are inside the insensitive "tube" (see figure 4.2), and the data points with either $\beta_i = C$ or $\beta_i = -C$ are outside the tube. Notice that both (4.6) and (4.10) are very similar. For the formulation of quadratic loss, the only difference is the extra diagonal elements added to the kernel. Therefore, when third party SVR software does not explicitly support the quadratic loss function, which is often the case, we can simply use an implementation of the linear ε-insensitive SVR to compute the kernel weights β_i for the quadratic ε-insensitive SVR. This is achieved by setting C to infinity and adding regularisation to the diagonal elements of the kernel, prior to

passing it as an argument to the algorithm. If this strategy is adopted, it is essential to note that the offset term, *b*, is calculated differently for both formulations.

If training samples with zero weights are removed from of the original training set, the new result will yield an identical solution. This means that any additional training samples, which locate in the insensitive tube of a previously trained solution, will not benefit the training. However, in practice, a cross validation using the extended training dataset may suggest a different "width" for the insensitive tube. Generally speaking, training SVR is very fast, and the solution is also guaranteed to be globally optimal. The sparse solution suggests that SVR is faster when predicting new data points, than is KRR - especially when the training set is very large. However, the two free parameters (C and ε) in SVR increase the computation time by requiring cross-validation to optimise them. In practice, the solutions obtained using SVR are not very sparse for fMRI data. A larger insensitive parameter can increase the sparsity, but it may also harm the performance.

For this thesis, SVR was performed by passing a pre-computed kernel matrix to the LIBSVM toolbox (http://www.csie.ntu.edu.tw/~cjlin/libsvm/).

4.1.2 Relevance Vector Regression

Relevance Vector Regression (RVR) is also a sparse kernel method, but it is formulated in a Bayesian framework. While the general expression takes the form of a dual formulation, RVR treats the kernel as a set of linear basis functions in order to obtain the form of equation (4.7). $\phi : \mathbf{x}_* \in \Re^D \rightarrow \phi(\mathbf{x}_*) = (k(\mathbf{x}_*, x_1), ..., k(\mathbf{x}_*, x_N)) \in \Re^N$. RVR is not strictly a kernel algorithm because its input is not required to be a kernel satisfying Mercer's condition. In other words, the kernel need not be symmetric and positive definite (Tipping, 2000, 2001). In fact, we could also use input features for RVM rather than kernels, and enabling sparsity in the feature space to be achieved

(Peng et al., 2008). It is also possible to take only a few "representative samples", and use the similarity measures, i.e. kernel values, or dissimilarity measures of those samples for the basis functions (Pekalska and Duin, 2005). In such a case, the input will be the *N* by *M* matrix, where *N*>*M*. The general RVM takes the full kernel for the input, and prepends a column of ones to model the offset. We will denote the *N* by *N*+1 basis functions by $\mathbf{\Phi} = [\mathbf{l}, \mathbf{K}]$, where $\mathbf{l}$ is an *N* element column vector of ones. The likelihood function of the data set can be modeled by a Gaussian distribution, $p(\mathbf{t} \mid \boldsymbol{\beta}, \sigma^2) = N(\mathbf{t} \mid \mathbf{\Phi}\boldsymbol{\beta}, \sigma^2\mathbf{I})$. Similar to the Bayesian view of ridge regression, each of the weights, $\boldsymbol{\beta}$, are assigned a unique zero mean Gaussian prior. This differs from ridge regression, where all the elements of the weight have the same variance, α^{-1}. The RVR models the prior of $\boldsymbol{\beta}$ with independent variance, $p(\boldsymbol{\beta} \mid \boldsymbol{\alpha}) = \prod_{i=0}^{N} N(\beta_i \mid 0, \alpha_i^{-1})$. This formulation is similar to the Bayesian view of ridge regression mentioned in section 2.4.2, so the posterior distribution is also similar to equation (2.46). It is given by $p(\boldsymbol{\beta} \mid \mathbf{t}, \boldsymbol{\alpha}, \sigma^2) = N(\boldsymbol{\beta} \mid \boldsymbol{\mu}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = (\sigma^{-2}\mathbf{\Phi}^{\mathbf{T}}\mathbf{\Phi} + \mathbf{A})^{-1}$ is the posterior covariance and $\mathbf{A} = diag(\alpha_0, \alpha_1, ..., \alpha_N)$ is the diagonal matrix with the precision or the inverse of the variance for each weight. $\boldsymbol{\mu} = \sigma^{-2}\mathbf{\Sigma}\mathbf{\Phi}^{\mathbf{T}}\mathbf{t} = (\mathbf{\Phi}^{\mathbf{T}}\mathbf{\Phi} + \sigma^2\mathbf{A})^{-1}\mathbf{\Phi}^{\mathbf{T}}\mathbf{t}$ is the maximum posterior weight. This is nearly identical to the maximum posterior solution for ridge regression, except the diagonal matrix does not have identical diagonal elements. Intuitively, this can be viewed as using different amounts of regularisation for each of the "training samples", where the amount of regularisation is controlled by the hyper-parameters. In the Bayesian framework, finding an optimum solution involves maximising the marginal likelihood (type-II maximum likelihood) with respect to the hyper-parameters $\boldsymbol{\alpha}$ and a noise variance σ^2. Because both the likelihood and the prior are modeled by Gaussian distributions, it is analytically feasible to derive the marginal likelihood function by

integrate over the parameters. The marginal likelihood is also a Gaussian

$$p(\mathbf{t} \mid \boldsymbol{\alpha}, \sigma^2) = \int p(\mathbf{t} \mid \boldsymbol{\beta}, \sigma^2)\, p(\boldsymbol{\beta} \mid \boldsymbol{\alpha}) d\beta = N(\mathbf{t} \mid 0, \mathbf{C}) \tag{4.11}$$

where $\mathbf{C} = \sigma^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^{\mathbf{T}}$ is the covariance of the marginal likelihood. The objective of the optimisation is to find the hyper-parameters, $\mathbf{A}, \sigma^2$, which maximise the "evidence" of the data (Mackay, 1992; Tipping, 2001). This is closely related to restricted maximum likelihood (ReML) and estimation of covariance components in the statistical literature. (Friston et al., 2002; Harville, 1977; Henderson, 1953). The covariance matrix that maximises the marginal likelihood can be obtained by iterative re-estimation or expectation maximisation (EM). For optimising the hyper-parameters, we can differentiate (4.11) and set the derivative to zero, based on the approach in (Mackay, 1992). The update is given by

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2}, \qquad (\sigma^2)^{new} = \frac{\| \mathbf{t} - \boldsymbol{\Phi} \boldsymbol{\beta} \|^2}{N - \sum_i \gamma_i} \tag{4.12}$$

where $\gamma_i = 1 - \alpha_i \Sigma_{ii}$ represents a measure of how well the corresponding parameter β_i is determined by the data, and $\Sigma_{ii} = (\sigma^{-2} \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{\varphi}_i + \alpha_i)^{-1}$ is the *i*th diagonal element of the posterior covariance. When α_i is large, β_i is heavily regularised, so $\Sigma_{ii} \approx \alpha_i$, and it follows that $\gamma_i \approx 0$. This indicates that the corresponding weight is less well determined by the data. When maximising the marginal likelihood, some of the α will grow very large, implying a small prior variance. Because the prior is zero mean, a parameter with an extremely small variance results will have its posterior probability sharply peaked at zero. This property allows irrelevant columns of the basis functions to be pruned out, and is known as automatic relevance determination (ARD)(MacKay, 1995). Because the solution is sparse, it means that only some of the training scans are used for prediction. Those scans are called "relevance vectors", and are analogous to "support vectors" in the SVM framework.

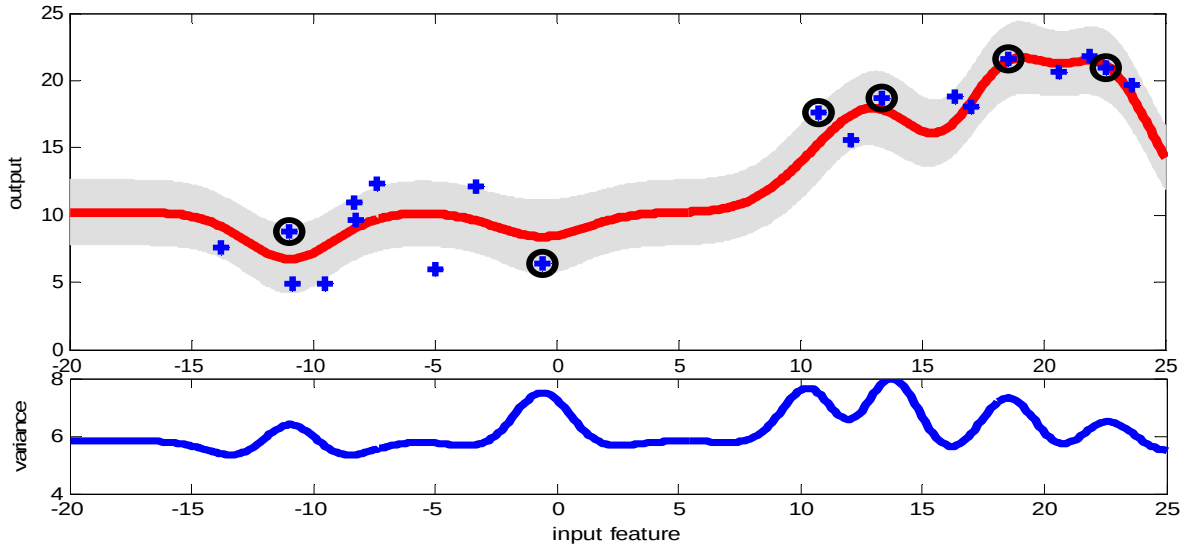


Figure 4.3 1D Relevance Vector Regression

Illustration of a one dimensional Relevance Vector Regression with an RBF kernel (the free parameter $\gamma = 0.3$). In the top plot, the solid line is the mean of the predictive distribution, and the grey stripe has the width of two standard deviations. The data points are shown as crosses, and the relevance vectors as circles. The bottom plot shows the variance of the predictive distribution. Notice that the variance estimate from RVR is actually higher around the relevance vectors, and smaller away from them.

When making the prediction, RVR has a similar form to other kernel methods, except that the test point is augmented as $\phi(\mathbf{x}_*) = [1, k(\mathbf{x}_*, \mathbf{x}_1), ..., k(\mathbf{x}_*, \mathbf{x}_N)]^{\mathbf{T}}$, where $\mathbf{x}_i, i = 1, ..., N$ are the data points in the training set. The prediction is given by

$$f(\mathbf{x}_*) = \sum_{i=0}^{N} \phi_i(\mathbf{x}_*)\beta_i \tag{4.13}$$

with a predictive variance of $\sigma_*^2 = \sigma^2 + \phi(\mathbf{x}_*)^{\mathbf{T}} \boldsymbol{\Sigma} \phi(\mathbf{x}_*)$. This is the variance in the data plus the uncertainty of the predicted maximum posterior weights. Unfortunately, there is a property of the predictive variance estimates from RVM, which is that if the test point $\mathbf{x}_*$ is far away from the centres of all relevance vectors, the values of the basis function $\phi(\mathbf{x}_*)$ become small. This formulation results the second term in the predictive variance going to zero, $\phi(\mathbf{x}_*)^{\mathbf{T}} \boldsymbol{\Sigma} \phi(\mathbf{x}_*) \approx 0$. Therefore if the test points are far from the training points, their estimated predictive variances are actually smaller

(Rasmussen and Quiñonero-Candela, 2005)(see figure 4.3). This behaviour is undesirable, and is not shared with Gaussian Process methods.

Although both SVR and RVR are sparse kernel methods, they exhibit different sparsity properties. Section 4.1.1 mentioned that removing non-support vectors from the training set will result the same training result for SVR, but this is not the case for RVR. Removing any non-relevance vectors from the training set may result in a different solution, and even a different set of relevance vectors.

Work in this thesis used in-house RVR code written in MATLAB® (http://www.mathworks.com/).

4.1.3 Gaussian Processes Regression

A Gaussian process is defined by a collection of random variables, any finite number of which is normally distributed (Rasmussen and Williams, 2006). Gaussian process approaches for machine learning were introduced relatively recently (MacKay, 1998, 2002; Williams, 1999). There are usually two views of Gaussian Process Regression (GPR): the weight-space view and the function-space view.

The weight-space view is from a similar perspective to the description of RVR and ridge regression, where a standard linear regression model with Gaussian noise is used, $p(\mathbf{t}\,|\,\mathbf{\Phi},\mathbf{w}) = N(\mathbf{\Phi w},\sigma^2\mathbf{I})$. This is a more general formulation than (2.40), where $\mathbf{\Phi} = \phi(\mathbf{X})$ encodes the projected features, and has dimensions of N by M. The weights are modelled by a zero mean Gaussian prior, $p(\mathbf{w}) = N(0,\Sigma_{prior})$. By applying the Bayes' rule (2.5) we can obtain the following relationship

$$p(\mathbf{w}\,|\,\mathbf{t},\mathbf{\Phi}) = \frac{p(\mathbf{t}\,|\,\mathbf{\Phi},\mathbf{w})\,p(\mathbf{w})}{p(\mathbf{t}\,|\,\mathbf{\Phi})}, \text{ where } p(\mathbf{t}\,|\,\mathbf{\Phi}) = \int p(\mathbf{t}\,|\,\mathbf{\Phi},\mathbf{w})\,p(\mathbf{w})\,d\mathbf{w} \qquad (4.14)$$

The posterior distribution is given by $p(\mathbf{w}\,|\,\mathbf{t},\mathbf{\Phi}) = N(\sigma^{-2}\mathbf{\Sigma}_{post}\mathbf{\Phi}^{\mathbf{T}}\mathbf{t},\mathbf{\Sigma}_{post})$, where $\mathbf{\Sigma}_{post} = (\sigma^{-2}\mathbf{\Phi}^{\mathbf{T}}\mathbf{\Phi} + \Sigma_{prior}^{-1})^{-1}$. To make a prediction of a test sample, t_*, we integrate

over all parameter values, which are weighted by the corresponding posterior probability.

$$\begin{aligned} p(t_* \mid \phi(\mathbf{x}_*), \mathbf{t}, \mathbf{\Phi}) &= \int p(t_* \mid \phi(\mathbf{x}_*), \mathbf{w}) p(\mathbf{w} \mid \mathbf{t}, \mathbf{\Phi}) d\mathbf{w} \\ &= \int \phi(\mathbf{x}_*)^{\mathbf{T}} \mathbf{w} p(\mathbf{w} \mid \mathbf{t}, \mathbf{\Phi}) d\mathbf{w} = N(\sigma^{-2} \phi(\mathbf{x}_*)^{\mathbf{T}} \mathbf{\Sigma}_{post} \mathbf{\Phi}^{\mathbf{T}} \mathbf{t}, \sigma^2 + \phi(\mathbf{x}_*)^{\mathbf{T}} \mathbf{\Sigma}_{post} \phi(\mathbf{x}_*)) \end{aligned} \tag{4.15}$$

All of the formulations above are identical to those for RVR. We can also re-write the equation using the kernel formulation by

$$\begin{aligned} p(t_* \mid \phi(\mathbf{x}_*), \mathbf{t}, \mathbf{\Phi}) = N(&\boldsymbol{\varphi}_*^{\mathbf{T}} \Sigma_{prior} \mathbf{\Phi}^{\mathbf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{t}, \\ &\boldsymbol{\varphi}_*^{\mathbf{T}} \Sigma_{prior} \boldsymbol{\varphi}_* + \sigma^2 - \boldsymbol{\varphi}_*^{\mathbf{T}} \Sigma_{prior} \mathbf{\Phi}^{\mathbf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{\Phi} \Sigma_{prior} \boldsymbol{\varphi}_*) \end{aligned} \tag{4.16}$$

where $\mathbf{K} = \mathbf{\Phi} \Sigma_{prior} \mathbf{\Phi}^{\mathbf{T}}$, and $\phi(\mathbf{x}_*)$ is denoted by $\boldsymbol{\varphi}_*$ for convenience. We often assume $\Sigma_{prior} = \alpha^{-1} \mathrm{I}$, so $\mathbf{K} = \frac{1}{\alpha} \mathbf{\Phi} \mathbf{\Phi}^{\mathbf{T}}$, which is a scaled kernel matrix, also known as the covariance function in the context of Gaussian Processes. Further details of the derivation can be find in the text book (Rasmussen and Williams, 2006), which is freely available online at http://www.gaussianprocess.org/gpml/chapters/.

In the function-space view of GPR begins with the definition of a simple Bayesian linear regression model, $\mathbf{y} = \mathbf{\Phi} \mathbf{w}$, with the prior $p(\mathbf{w}) = N(0 \mid \Sigma_{prior})$. Because $\mathbf{y}$ is a linear combination of Gaussian distributed variables, it is itself also Gaussian distributed, and defined by the following mean and covariance.

$$\begin{aligned} \mathrm{E}[\mathbf{y}] &= \mathbf{\Phi} \mathrm{E}[\mathbf{w}] = 0 \\ \mathrm{cov}(\mathbf{y}) &= \mathbf{\Phi} \mathrm{E}[\mathbf{w} \mathbf{w}^{\mathbf{T}}] \mathbf{\Phi}^{\mathbf{T}} = \mathbf{\Phi} \Sigma_{prior} \mathbf{\Phi}^{\mathbf{T}} = \mathbf{K} \end{aligned} \tag{4.17}$$

For the regression model with Gaussian noise, the likelihood probability is given by $p(\mathbf{t} \mid \mathbf{y}) = N(\mathbf{y} \mid \sigma^2 \mathbf{I})$, and the marginal distribution is $p(\mathbf{y}) = N(0, \mathbf{K})$. Integration over $\mathbf{y}$ gives the marginal distribution

$$p(\mathbf{t}) = \int p(\mathbf{t} \mid \mathbf{y}) p(\mathbf{y}) d\mathbf{y} = N(\mathbf{t} \mid 0, \mathbf{C}) \tag{4.18}$$

where the covariance matrix $\mathbf{C}$ is defined by

$$\mathbf{C} = \mathbf{K} + \sigma^2\mathbf{I} \tag{4.19}$$

This equation resembles the covariance matrix for the marginal likelihood for RVR in (4.11). In fact, RVR is a special case of GPR. The covariance matrix in RVR is defined by $\mathbf{C} = \alpha_1^{-1}\boldsymbol{\varphi}_1^{\mathbf{T}}\boldsymbol{\varphi}_1 + \alpha_2^{-1}\boldsymbol{\varphi}_2^{\mathbf{T}}\boldsymbol{\varphi}_2 + ...\alpha_N^{-1}\boldsymbol{\varphi}_N^{\mathbf{T}}\boldsymbol{\varphi}_N + \sigma^2\mathbf{I}$, where $\boldsymbol{\varphi}_i$ is the *i*th row in the feature matrix $\boldsymbol{\Phi}$. A simple covariance matrix with a linear kernel, a constant and the noise term is given by $\mathbf{C} = \theta_1\mathbf{X}\mathbf{X}^{\mathbf{T}} + \theta_2 + \theta_3\mathbf{I}$, where $\theta_i \geq 0$ models the weighting for each component. More generally, we can define the kernel matrix by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 K_1(\mathbf{x}_i, \mathbf{x}_j) + \theta_2 K_2(\mathbf{x}_i, \mathbf{x}_j) + \theta_3 K_3(\mathbf{x}_i, \mathbf{x}_j)... + \theta_k \tag{4.20}$$

For example, the kernel matrix **K** can be a positive weighted combination of an RBF kernel, a linear kernel, a polynomial kernel and a constant. Here we take advantage of the kernel trick without needing to explicitly define the feature matrix $\boldsymbol{\Phi}$.

For *N* training samples, we denote the distribution of the training data plus a new test data point by $p(\mathbf{t}_{N+1}) = N(0, \mathbf{C}_{N+1})$, where $\mathbf{t}_{N+1} = [t_1, ..., t_N, t_*]^{\mathbf{T}}$ and $\mathbf{C}_{N+1}$ is an *N*+1 by *N*+1 covariance matrix

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C} & \mathbf{k} \\ \mathbf{k}^{\mathbf{T}} & c \end{pmatrix} \tag{4.21}$$

Covariance matrix **C** is defined by (4.19), the vector **k** has elements $\mathbf{k}_i = K(\mathbf{x}_i, \mathbf{x}_*), i = 1, ..., N$, and the scalar value $c = K(\mathbf{x}_*, \mathbf{x}_*) + \sigma^2$. Using the properties of conditional Gaussian distributions, we can compute the conditional distribution of $p(t_* \mid \mathbf{t})$ by

$$p(t_* \mid \mathbf{t}) = N(\mathbf{k}^{\mathbf{T}}\mathbf{C}^{-1}\mathbf{t}, c - \mathbf{k}^{\mathbf{T}}\mathbf{C}^{-1}\mathbf{k}) \tag{4.22}$$

This is equivalent to (4.16) by defining $c = \boldsymbol{\varphi}_*^{\mathbf{T}}\Sigma_{prior}\boldsymbol{\varphi}_* + \sigma^2$ and $\mathbf{k} = \boldsymbol{\varphi}_*^{\mathbf{T}}\Sigma_{prior}\boldsymbol{\Phi}^{\mathbf{T}}$. However, the formulation in (4.22) does not require the feature space to be specified directly. Note that the mean of the predictive distribution can be included in the

general kernel regression equation $y_* = \sum_{i=1}^{N} \beta_i K(\mathbf{x}_*, \mathbf{x}_i) = \mathbf{k}^{\mathrm{T}}\boldsymbol{\beta}$, where $\boldsymbol{\beta} = \mathbf{C}^{-1}\mathbf{t}$. Because $\mathbf{C}$ is the combination of the kernel matrix and a diagonal matrix, this resembles the kernel ridge regression (3.3), where the kernel weights are given by $\boldsymbol{\beta} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{t}$. However, in the usual ridge regression framework, the free parameters are often determined by cross validations. In the Bayesian framework, those free parameters, or hyper-parameters, can be learned by maximising the marginal likelihood (4.18). This is also known as the "model evidence", whose log likelihood function is given by

$$\ln(p(\mathbf{t} \mid \theta, \sigma^2)) = -\frac{1}{2}\ln|\mathbf{C}| - \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t} - \frac{N}{2}\ln(2\pi) \tag{4.23}$$

It is therefore possible to optimise the evidence in order to find the optimum regularisation for a ridge regression, without resorting to cross validation. This framework has many other uses, for example, it could be used to find the optimum threshold for high-pass filtering of fMRI time series. Furthermore, the evidence framework can allow comparisons among different pattern mapping models (Friston et al., 2008). In this work, which I am a co-author of, we mapped input voxels into different linear pattern spaces. This included the use of linear kernels, principal components, and spatially smoothed input data. A greedy search was applied to find sparse solutions that maximise the model evidence in those pattern spaces. Instead of making predictions, we made inferences based on the model evidence from different pattern spaces, and compared the models this way.

The major advantage of using model evidence to select hyper-parameters is its computational efficiency. When the number of free parameters increases, the computation for cross validation grows exponentially. In practice, hyper-parameters learned through maximising marginal likelihood often provide reasonably good performance, but they are not always better than hyper-parameters learned by cross-validation. Examples will be described in the next section on practical

applications.

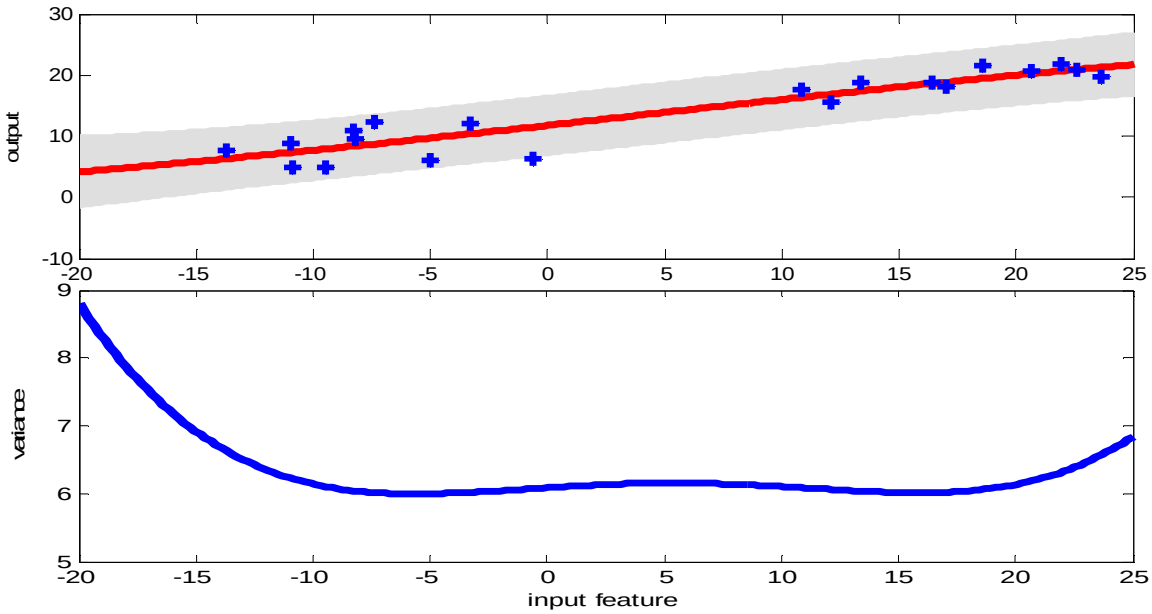


Figure 4.4 1D Gaussian Process Regression

Illustration of a one dimensional Gaussian Process Regression using a RBF kernel. The hyper-parameters of the RBF kernel and the data noise were learnt through maximising marginal likelihood or the "model evidence". The input data features are the same as in figure 4.3. In the top plot, the solid line is the mean of the predictive distribution, and the grey stripe has a width of two standard deviations. The data points are shown by crosses. The bottom plot shows the variance of the predictive distribution. Notice that the variance is actually lower around the data points and higher away from them. This property is the converse of RVR, and is more accurate.

For the implementation, we used the GPML Matlab toolbox (http://www.gaussianprocess.org/gpml/code/matlab/doc/) with some modifications to allow the use of pre-computed kernel. We also add some codes to enable generating linear combination of kernels. There is another in house implementation by Dr. Ashburner. In this implementation, Powell's line search method (Press et al., 1992) is applied to optimize the marginal likelihood.

4.2 Application: Pittsburgh Brain Activity Interpretation Competition 2006

In the spring of 2006, the psychology department in the University of Pittsburgh held the international "Pittsburgh Brain Activity Interpretation Competition" (PBAIC), in conjunction with the Organization for Human Brain Mapping conference, which was held in Florence later that year. All details about the competition can be found at http://www.lrdc.pitt.edu/ebc/2006/competition.html. The competition was primarily organised by the "Experienced Based Cognition" (EBC) project, of which the principal investigator was Prof. Walter Schneider. The goal of the competition was to "challenge multiple groups to use state-of-the-art techniques to infer subjective experience from a rigorously collected set of fMRI." In other words, the competition aimed to test the ability of different pattern recognition methods for mapping the BOLD patterns in fMRI data to the subjects' experience during scanning. These techniques are often referred as "fMRI decoding". There were 273 teams registered to take part in the competition and 40 teams submitted their results at the end. Our entry came 5th in the competition. The prize money was $10,000 for the first place, $5,000 for the 2nd place, and $2,000 for the 3rd place.

4.2.1 Overview of the competition: data, goals, and scoring system

The fMRI data was provided by the competition organisers. Data were collected from three subjects viewing three 20 minute long videos, which were re-edited segments of the American comedy "Home Improvement". There were also resting (blank) periods for each run of the video. Rating data were collected on 13 subjective features rated by each subject, plus seven actor ratings and seven location ratings. The subjective ratings were collected after the fMRI session from each subject separately, which means subjects watched the videos repeatedly to provide the 13 different

ratings. Thirteen of the ratings were compulsory for the participants of the competition, and there were an additional 14 optional ratings, which would benefit the participant if they scored higher than the average of compulsory ratings. The ratings were named “amusement”, “attention”, “arousal”, “body parts”, “environmental sounds”, “faces”, “food”, “language”, “laughter”, “motion”, “music”, “sadness”, and “tools”. For further details, please refer to the competition websites. All the ratings were scaled between 0 and 1, and the baseline for most of the ratings was 0, except for arousal and attention, which had a baseline of 0.5. The organiser provided all the fMRI volumes for the three subjects viewing the three videos segments, but only the ratings for the first two videos were provided. The objective for each team was to use data collected from the first two movies in order to learn the mappings from fMRI volumes to the subjective ratings, and then apply the learnt mapping to predict the ratings from the third video. To resemble the more conventional GLM analysis, each rating was convolved by the canonical hemodynamic response function (HRF) (figure 3.6). Instead of predicting the original ratings, the objective was to predict the HRF convolved ratings i.e. regressors in the design matrix.

The fMRI volumes were acquired in a Siemens 3T Allegra scanner with TR=1.75s, TE=25ms, and flip angle=76 degrees. The slice thickness was 3.5mm and xy voxel size was 3.28mm. Each video was approximately 20 minutes, so there were 858 volumes for the first run (corresponding to the first video), 868 volumes for the second run, and 900 volumes for the third run. The video and volumes were approximately synchronised across the three subjects. Additional structural data was also collected in the same scanner, using a MPRAGE sequence with 1mm slice thickness and 0.82 xy voxel size. The organisers kindly provided pre-processed data done using Brain Voyager. We chose to pre-process the raw data with SPM5 ourselves.

The score for each rating was evaluated using Pearson's correlation between the predicted rating and the true HRF convolved rating. To encourage participants to improve their predictions and achieve higher accuracy, the scores were converted into Fisher's z score, $z_r = \tanh^{-1}(r_{xy}) = \frac{1}{2}\log(\frac{1+r_{xy}}{1-r_{xy}})$. In order to rank the performance from different teams, the competition committee developed a single "competition score" by averaging all the z scores for thirteen ratings of all three subjects, and then convert the averaged z scores back to correlation coefficients. Because z scores are non-linearly related to correlations, improving a correlation from 0.65 to 0.75 increases the scores by twice as much as improving a correlation from 0.1 to 0.2. Each team was allowed to submit their prediction of run three only three times. The best competition score among the three submissions was compared with the best scores of the other teams.

4.2.2 Our approaches to tackle PBAIC 2006

This competition was held less than six months after I began my PhD study. In addition, the release of the full dataset was delayed, so each team had only about five weeks to complete their entries. As I was very inexperienced, I teamed up with a senior researcher, Dr. Janaina Mourao-Miranda, from the Institute of Psychiatry, Kings College. At this stage, I still had much to learn, and spent more than a week pre-processing the dataset, using various SPM5 pre-processing options. I worked on RVR with my own pre-proceesed data, and Dr. Mourao Mirand worked on SVR with spatially normalised data pre-processed by the competition organisers.

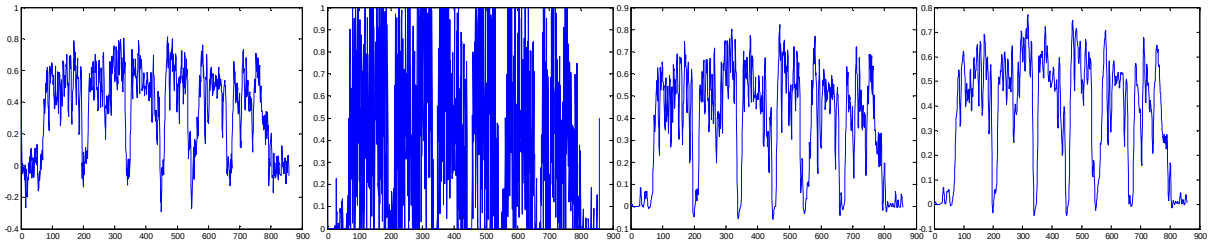
For our own pre-processing steps, all functional data was realigned and resliced, and the T1 structural image was coregistered with the fMRI data. The grey matter (GM) was segmented from the structural image, and then resampled to have the same dimension as the functional data. This GM map was used to mask out non-grey matter tissues. The time series of each voxel were detrended using a piecewise linear

regression model. The breakpoints of each linear segment were set at the middle of each resting period. At this point, I had not realised that detrending could be done to the kernel using a residual forming matrix, as described in equation (3.14). I took an inefficient approach of detrending all voxels for all three runs of each user. Then different amounts of Gaussian smoothing were applied to the data. We also tried using the ROI time series provided by the competition committee as input features.

Standard RVR was used to predict the subjective ratings. We used each rating as the target **t** and a kernel generated from the pre-processed fMRI volumes as the input. In this work, we treated each rating independently and trained them separately. The prediction accuracies for different settings, such as different spatial smoothing and different choices of input features, were determined by two fold cross validation i.e. train using ratings from run one, and then predict the ratings for run two, and vice versa. An interesting phenomenon was observed, in that we could actually predict one subject's subjective rating using another subject's fMRI data. We believed that this was due to synchronisation among the three subjects in terms of when the videos were viewed. Therefore, we also combined input features from different subjects when predicting the rating of one subject. i.e. summing up the three linear kernels generated from each individual.

A further processing step was applied in order to improve their accuracy. In most cases, the range of the raw feature ratings $\mathbf{z}_{raw}$, prior to convolution with the haemodynamic response function (HRF), was between zero and one. To utilise this prior knowledge, a constrained de-convolving strategy was applied. The "canonical HRF", which the competition used to convolve the raw ratings with, was generated. The convolution can be implemented as a matrix multiplication of the raw rating by a toeplitz matrix, such that $\mathbf{t} = \mathbf{H}\mathbf{z}_{raw}$. The objective is to recover the raw rating $\mathbf{z}_{raw}$ fulfilling the constraints by minimising the sum of square loss between the

re-convolves solution $\mathbf{Hz}_{raw}$ and the predicted rating $\mathbf{t}_*$.



Deconvolved data constrained from 0 to 1.

Smoothed data Correlation =0.8854

Figure 4.5 Constrained deconvolusion

Illustration of constrained deconvolution using quadratic programming. This was used because we know the original rating (before convolved by HRF) is between 0 and 1. Cross validation showed that this technique generally improved the prediction accuracy slightly. The additional temporal Gaussian smoothing also boosted the prediction accuracy.

Quadratic programming (the same optimisation used by SVM) was used to deconvolve the HRF from the predictions (t*) by

$$\begin{aligned} &\underset{\mathbf{z}_{raw}}{\operatorname{argmin}}\{(\mathbf{Hz}_{raw}-\mathbf{t}_*)^{\mathbf{T}}(\mathbf{Hz}_{raw}-\mathbf{t}_*)\} = \underset{\mathbf{z}_{raw}}{\arg\min}\{\mathbf{z}_{raw}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\mathbf{Hz}_{raw}-2\mathbf{t}^{\mathbf{T}}\mathbf{Hz}_{raw}\} \\ &\text{subject to } 0 \le \mathbf{z}_{raw} \le 1 \end{aligned} \tag{4.24}$$

The new predicted rating is then $\mathbf{t} = \mathbf{Hz}_{raw}$. Ideally, some sort of procedure such as Weiner filtering would have been used. Estimates of the expected temporal power spectrum of the predicted time course (derived from the smoothness of the scores used for training), and the power spectrum of the errors (obtained by making use of the probabilistic nature of the RVR), would have allowed more accurate deconvolution to be performed. As a compromise, we smoothed the reconvolved data slightly using a Gaussian smoothing kernel of 3 TRs FWHM. The FWHM was based on empirical testing.

4.2.3 Our result in PBAIC 2006

Before we submitted our predictions for run three, we did some cross validations. The only variation in our attempts was the input features. For training and testing, we used SVR and RVR throughout the competition, and hence the results of each attempt will be listed here with four different ranks from poor (scores<0.3), neutral (0.3<scores<0.4), good (0.4<scores<0.45), to excellent (0.45<scores). These are based on the ranking results. Because of the time constraints, not all features were tested with all three subjects, for both video one and video two. The two free parameters for SVR were also selected empirically (Dr. Mourao-Miranda was responsible for all the SVR training)

Input features type	**Learning methods (SVR or RVR or both)**	**Subject**	**Rank**
Individual pre-processed grey matter masked	RVR	All 3	Poor
Individual Smoothed pre-processed grey matter masked	RVR	All 3	Neutral
Individual normalised & pre-processed whole brain	SVR & RVR (RVR slightly better)	All3	Good for subject 1,2, excellent for subject 3
Individual ROI time series (ROI data was provided by the competition committee)	RVR & SVR	All 3	Neutral
Combining ROI of 3 subjects	RVR & SVR	All 3	Good for subject 1,2 Neutral for subject 3
Combining normalised & pre-processed whole brain of 3 subjects	RVR	Subject 1,2	Good
Combining normalised & pre-processed whole brain of 3 subjects	SVR	Subject 1,2	Neutral

Combining smoothed grey matter masked brain of 3 subjects	RVR	Subject 1, 2	Excellent
Combining smoothed grey matter masked brain of 3 subjects	RVR	Subject 3	Neutral

The finding was fairly interesting. In most cases, RVR seemed to perform better than SVR according to the cross validations. We found out that by adding the kernels of three subjects together, we could predict subject one and subject two better, but not subject three. The ratings could be modelled as a “true rating” with noise added. For subjects one and two, the variance maybe high compared with that from subject three. The optimal combination of data from several people may help RVR estimate the variance components better, but since the rating of subject three has lower variance already, this procedure may actually increase the variance of the prediction for subject three. Due to the time constraint, we did not predict any optional ratings.

Because we had three chances to submit, we submitted the prediction from Dr. Mourao-Miranda’s work in the first attempt. For our second attempt, we submitted the predictions based on our cross-validation results, hence for subject one, the prediction was from “combining smoothed grey matter masked brain of three subjects” with RVR, for subject two, the prediction was from “combining smoothed grey matter masked brain of three subjects” with SVR, and for subject three, the prediction was using “Individual pre-processed whole brain” with RVR. For the first two submissions, the competition organiser returned our results within 24 hours. The results had shown that RVR was superior than SVR (perhaps we did not find the optimum parameters for SVR). Therefore in our final attempt, we submitted the same predictions as our second submission, except for subject 2, where the prediction was from “combining smoothed grey matter masked brain of 3 subjects” with RVR instead of SVR. We

achieved a competition score of 0.477 and ranked fifth place, which was higher than the first two submissions. The competition scores from the first place to the fourth place are as following, 0.515, 0.509, 0.493, and 0.484. Here we can see that the scores in the top five places are very close. The results for others teams can be found at http://www.lrdc.pitt.edu/ebc/2006/2006results.html. Figure 4.6 shows our final competition score relative to the other teams. Generally speaking, we did relatively well, and had four ratings predicted within the top 5%. Our work was also presented in the poster session of the 2006 OHBM conference.

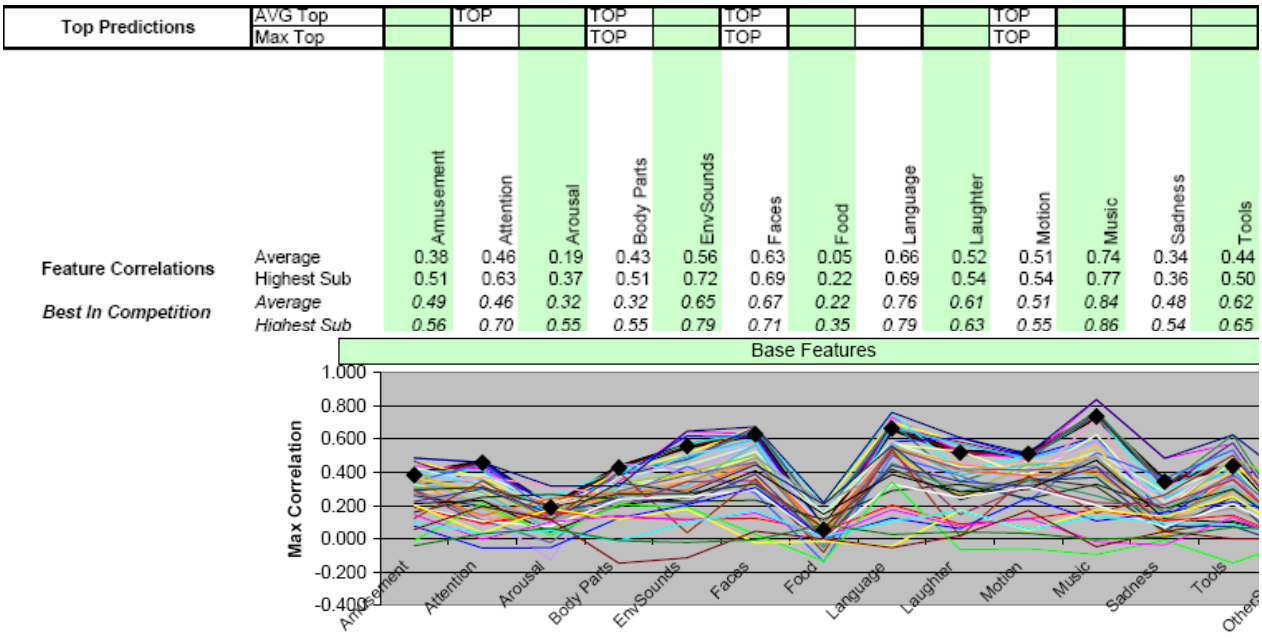


Figure 4.6 Our result of PBAIC 2006

The figures show the summary of our competition scores across 13 compulsory ratings. The bottom plots show the maximum correlation from one of the subject at each feature rating for all the teams. Our team was shown with the black line with dots. Any feature in which the entry was within the top 5% of prediction is indicated as TOP at the top of this figure.

4.2.4 Post-competition analysis

A few months after the competition, the organising committee reopened their scoring system without disclosing the true ratings for run three. The system allowed

each registered user to submit one prediction every seven days. During the OHBM conference, the top three teams gave oral presentations as well as presenting their work by poster. Instead of pooling subject's fMRI data to predict each subject's ratings, all three teams averaged their prediction from three subjects. Inspired by this idea, I simply averaged our final submission and submitted the same averaged prediction for all three subjects. This achieved a score **0.497**, which was higher than the team in third place (0.493).

Because we knew there would be another competition in 2007, we took the opportunity to test different algorithms and different input features. We submitted new predictions for run three every week, and the table below indicates the different trials we did. In the new trials, the predictions were averaged for three subjects by default. If not specified otherwise, the input used a linear kernel.

Input feature and pre-processing	Algorithm description	Score
Normalised, pre-processed by the organiser, whole brain	Modified RVR algorithm with single prior variance. The formulation is similar to GPR with a linear kernel.	0.485
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian, whole brain	Modified RVR algorithm with single prior variance. The formulation is similar to GPR with a linear kernel.	0.495
Combining, normalised, spatially smoothed with 6mm Gaussian, grey matter masked brains of 3 subjects (summation of three linear kernels from three subjects)	Standard RVR	0.395
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian,	Standard RVR	0.480

whole brain		
Normalised, pre-processed by the organiser, whole brain. Temporal smoothing by 2 TR Gaussian	Standard RVR	0.472
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian, grey matter	Standard RVR	0.491
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian, grey matter	Standard RVR with RBF kernel (free parameter determined by peak histogram of RBF kernel matrix at 0.5)	0.487
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian, grey matter	Standard RVR with second order polynomial kernel ($\mathbf{K}_{poly} = (\mathbf{K} + 10^6)^2$)	0.483
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian, grey matter	Modified RVR algorithm with single prior variance using second order polynomial kernel ($\mathbf{K}_{poly} = (\mathbf{K} + 10^6)^2$)	0.483
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian, grey matter	SVR with fixed free parameters (C= 0.00001, epsilon=0.1)	0.479
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian, grey matter	SVR free parameters optimised through cross validation	0.472
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian, grey matter	Standard RVR with linear kernel, except for “Attention”, “Environmental Sounds”, “Sadness“, and “Tools” we used RBF kernel (free parameter determined by peak histogram of RBF kernel matrix at 0.5)	0.521
Normalised, pre-processed	Standard RVR with	0.486

by the organiser, spatially smoothed with 6mm Gaussian, grey matter	normalised kernel	
Normalised, pre-processed by the organiser, spatially smoothed with 6mm Gaussian, grey matter	KRR, regularisation determined through cross validation	0.414

From the table, it is clear that most of the results are better than our competition score. RVR seems to be slightly better than SVR, but the difference is tiny, and may not be statistically significant. Oddly, the free parameters of SVR optimised by cross validation for each rating and each subject did not work as well as the fixed parameters. It might due to the bias from two fold cross validation. Generally speaking, Gaussian smoothed and grey matter masked images were the most preferred input features. RVR was also favoured, not because its prediction accuracy was better, but because it did not require additional cross validation. Notice that when we combined the linear kernel and the RBF kernel, we achieved a score of 0.521, which was higher than the score achieved by first prize team (0.515).

For the ratings predicted by a linear kernel, we can also compute "weighting maps", which are the weights in voxel space. This enables visualisation of which regions contribute most to the prediction.

$$y = \sum_{i=1}^{N} \beta_i \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_* + b = \mathbf{w}^{\mathrm{T}} \mathbf{x}_* + b, \qquad \mathbf{w} = \sum_{i=1}^{N} \beta_i \mathbf{x}_i \tag{4.25}$$

The weighting in voxel space often resembles the maps generated from the conventional mass-univariate analysis; however, it is essential to understand the differences between mass-univariate and multivariate approaches. Regions with low p-values in a mass-univariate analysis should have high absolute weighting in the "weight map", but this does not necessary apply the other way around.

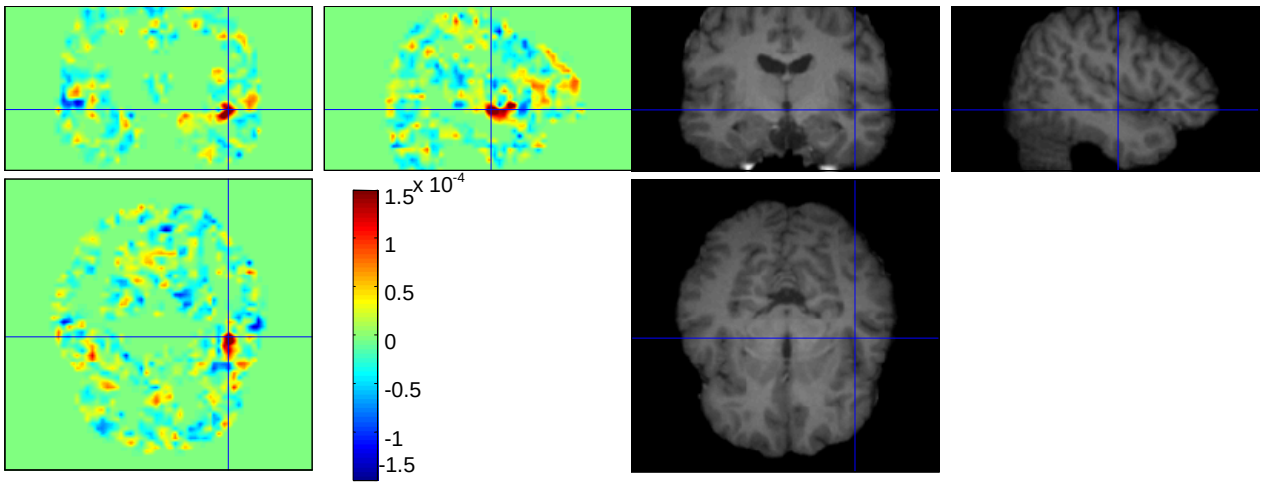


Figure 4.7 Weight map of the rating "Music"

The figure on the left shows the projected weighting in the grey matter masked voxel-space. This is referred to as the "weighting map", computed from the RVR and equation (4.25). The right figure is the corresponding structural MRI. This is the predicted map of subject three training on "Music" with both runs one and two. We can see clearly that voxels in the right auditory cortex have high weights, indicating those voxels that contribute more in the prediction of the rating "Music".

4.3 Application: Pittsburgh Brain Activity Interpretation Competition 2007

In the spring of 2007, the same competition organisers which organised the PBAIC 2006 held the second PBAIC in conjunction with the Organization for Human Brain Mapping conference at Chicago later that year. This time we were much better prepared. After gaining experience from re-analyzing the dataset from PBAIC 2006, we had clear knowledge about which algorithms and pre-processing procedures are most suitable. I also invited a friend of mine, Mr. Yizhao Ni, who was a PhD candidate at the University of Southampton, specialising in kernel methods, into our team. In this competition, we won the $10,000 first prize. All the competition details can be found at http://www.lrdc.pitt.edu/ebc/2007/competition.html.

4.3.1 Overview of the competition

The main difference of the competition in 2007, compared with that from 2006, was that most of the ratings in PBAIC 2007 were objective rather than subjective. The aim was to understand how the brain represents and manipulates information dynamically during real-world behaviour. The organisers developed a virtual reality (VR) system. Instead of receiving passive stimulation, as in the previous competition, subjects were asked to navigate freely around the VR environment. A few tasks were also given to the subjects (or players, since it is basically a video game). The players were asked to collect items on the ground or to take pictures of people with piercings. When subjects played the game in the MR scanner, the system could collect the activities of the players and the fMRI data synchronously. Eye trackers were also used to monitor the field of view of the players. This information was later converted into ratings related to what the players were seeing during the game. There were 11 objective ratings, which are “hits” (collecting items), “search people”, “search weapons”, “search fruit”, “listening to instructions”, “dog barking”, “seeing faces”, “seeing fruit and vegetables”, “seeing weapons and tools”, “interior or exterior of the buildings”, and “velocity”. The two subjective ratings were “arousal” and “valence”, which were rated by reviewing the “game play” after the fMRI scanning of each subject

The format of PBAIC 2007 was the same as that of PBAIC 2006. There were three subjects and three runs. The competition committee disclosed the ratings for runs one and two, and the objective was to predict the ratings for run three. The rating system (raw and HRF convolved), MRI scanner, MR sequence, scoring system, and data format were all the same as PBAIC 2006, except that all three runs had exactly the same length of 704 fMRI volumes. From the contestants’ point of view, PBAIC 2007 is the same as PBAIC 2006.

4.3.2 Pre-processing and feature selection

We initially tried various pre-processing options for subject two, and estimated prediction accuracies through two fold cross validation. The fMRI volumes in this competition had a higher level of non-linear drift in the signal intensities, but the pre-processed data provided by the organisers had only been linearly detrended. This may be the reason why some teams who performed well in PBAIC 2006 did not perform as well in PBAIC 2007. For example, the Italian group who won 1st place in PBAIC 2006 were not even among the top 10 of PBAIC 2007.

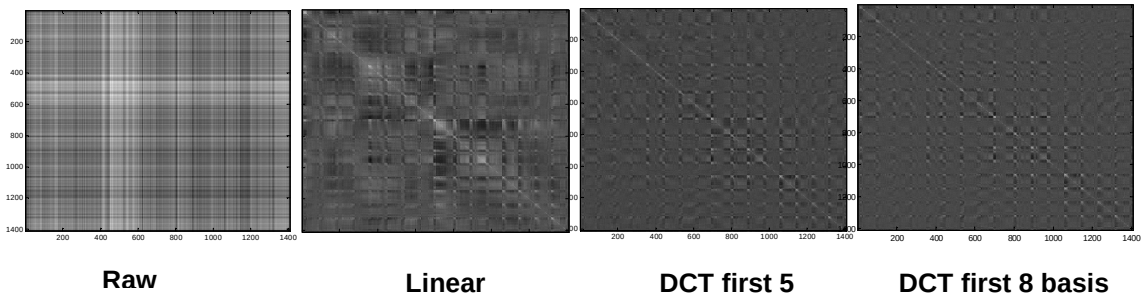


Figure 4.8 Linear kernel with different level of detrending

The images show the linear Kernel with raw data (no detrending) and different degrees of detrending. The raw kernel without temporal detrending and the linear detrended kernel have less uniform intensities than those kernels with more low frequency components removed. Visualisation of the linear kernel can provide rough guidance of the quality of the data. Ideally, the "grids" in the linear kernel should be prominent, as they indicate the resting or fixation periods in the scanning runs. (Notice: the raw data here was still motion corrected).

Vsual inspection of the linear kernel (figure 4.8) shows a patchy looking effect, even after the linear detrending done by the competition organisers. Because of this, we did our own pre-processing using SPM5. All the scans were first realigned and resampled to remove the effects of subject motion. We did not attempt to correct for the fact that all slices of an fMRI volume are collected at different times, although some such adjustments had been made to the pre-processed dataset provided by the organisers.

To further reduce dimensionality, those voxels that were, a priori, considered

non-informative were removed. Selecting informative voxels can be seen as a form of feature selection (Guyon and Elisseeff, 2003), which can often increase the signal to noise ratio. In the context of fMRI, BOLD signal change is generally believed to occur mainly in grey matter, as its major cause should be the local neuronal activities (Logothetis et al., 2001). Masks defining grey matter were generated for each subject by segmenting one of the fMRI scans (Ashburner and Friston, 2005). It may also have been possible to coregister the anatomical image with the fMRI, and identify grey matter from this. However, this was not done because EPI data tend to suffer from spatial distortions, especially in the frontal region due to the air in the frontal sinus.

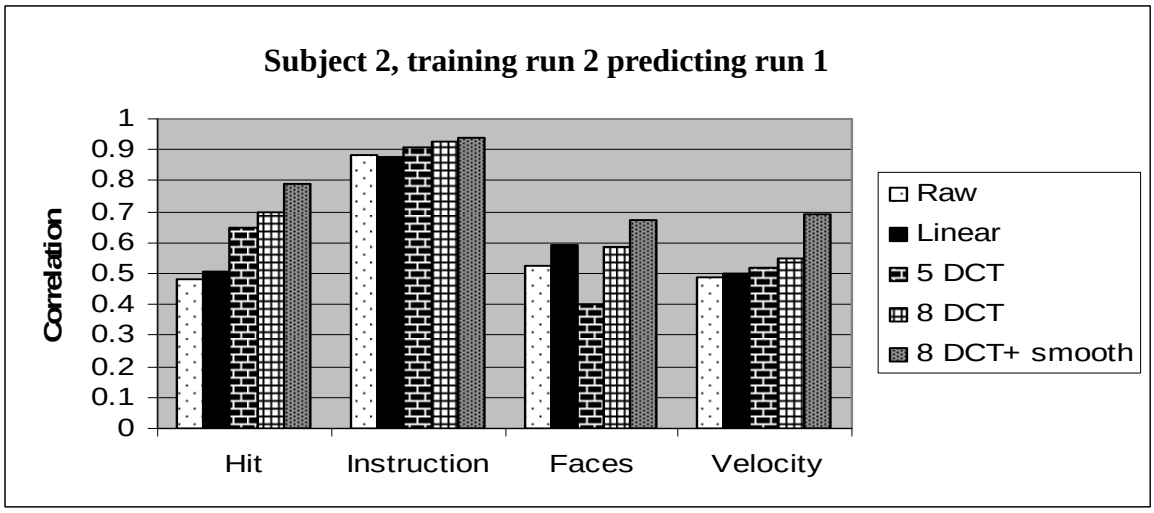


Figure 4.9 Prediction accuracy with different pre-processing

This figure shows the prediction accuracy of subject two for predicting the first run of the VR game by training the second run with different detrending settings. It is clear that the raw data without any detrending performed poorly. From our empirical results, removing the lower 8 DCT components resulted the best cross validation accuracy. Spatial smoothing with 6mm FWHM Gaussian also boosted the prediction accuracy significantly. Spatial smoothing seemed to result extensive improvement in PBAIC 2007 than the effect of smoothing in PBAIC 2006.

To remove low frequency components (high pass filtering), we relied on the cross validation of subject two, and empirically determined that removing the lower eight DCT components was optimal (figure 4.9). This was equivalent to a high pass

filtering with the cut-off frequency at 0.0057Hz. (SPM default cut-off frequency is 1/128=0.0078). Additional smoothing with a 6mm FWHM Gaussian kernel was also applied spatially. No temporal smoothing was applied.

Ugly Duckling Theorem (Duda et al., 2000; Watanabe, 1970) tells us that prior knowledge is essential for quantifying the similarity between things, so knowledge about human brain function was used to further increase the signal to noise ratio and suppress those features that were believed, a priori, to be less informative. It is known from the functional brain mapping literature that some cognitive functions and sensory perceptions are regionally localised. Hence, masks were used to weight the kernels when predicting the two feature ratings: “dog barking” and “interior or exterior of the building”. It was believed that most of the fMRI pattern resulting from the barking sound would be localised in auditory cortex. Similarly, the major discrimination between the inside and outside of the buildings would be the illumination differences. Therefore, a mask of visual cortex could mask out a large amount of irrelevant signal. In order to generate the mask of functional regions for all three subjects, first, the cytoarchitectonic maps of visual and auditory in stereotaxic space were downloaded from the McConnell Brain Imaging Center (http://www.bic.mni.mcgill.ca/cytoarchitectonics/). Then the deformation map was generated using the spatial normalisation routine in SPM5, but rather then warping the individual to the MNI space, the cytoarchitectonic maps in MNI space were warped to match the individual subjects fMRI data. Finally, (although not really necessary) a threshold of 0.3 was used to convert the probability maps into binary masks. We could also have weighted each voxel by the corresponding probabilities.

4.3.3 Predicting general ratings and details on how to achieve nearly perfect predictions for some ratings

In this competition, there were 13 compulsory ratings and 10 optional ratings.

Initially, we applied both standard KRR and RVR to all the ratings. It was identical to what we did for PBAIC 2006. All the ratings were treated independently, hence the same kernel was used as the input with different ratings as the target variable, except “dog barking” and “interior or exterior of the building”. For those two ratings, we used kernels generated from auditory cortex and visual cortex respectively. Most ratings were predicted using linear kernels, except for “Valance” and “Arousal”. For those two ratings, we used RBF kernels, and selected the parameter that resulted in the peak of the histogram of the RBF kernel at 0.5. (Note, all elements in the RBF kernel are between 0 and 1). The regularisation for KRR was determined by cross validation, and both methods showed similar results.

Because the competition scoring was based on Z-scores, we found that increasing a correlation from 0.8 to 0.9 resulted in three times as much improvement in the final scores as raising a correlation from 0.2 to 0.3. The goal was therefore to focus attention on those ratings that could be predicted reasonably well, and improve them further. By watching the re-play of the VR games provided by the competition organiser, we found a few insights into the VR games. It was observed that for each run, the “instructions” ratings had seven spikes, all of which had similar shapes across all subjects and all runs. It became apparent that an ad hoc model fitting strategy could be used to further improve what were already high correlations. Firstly, kernel regression was applied to predict the rating, and then the prediction was convolved with the model shape, which was generated by averaging all the spikes in all runs of all subjects. This is equivalent to match filtering, and the peak values in the convolved ratings indicate the location where the average shape fits best. After finding the estimated peak location, the average shape was placed in (Figure 4.10). Without this procedure, the correlation of the predicted rating was 0.8, whereas by adopting it, the final correlation reached 0.988, which increased the Z-score from 1.0986 to 2.555.

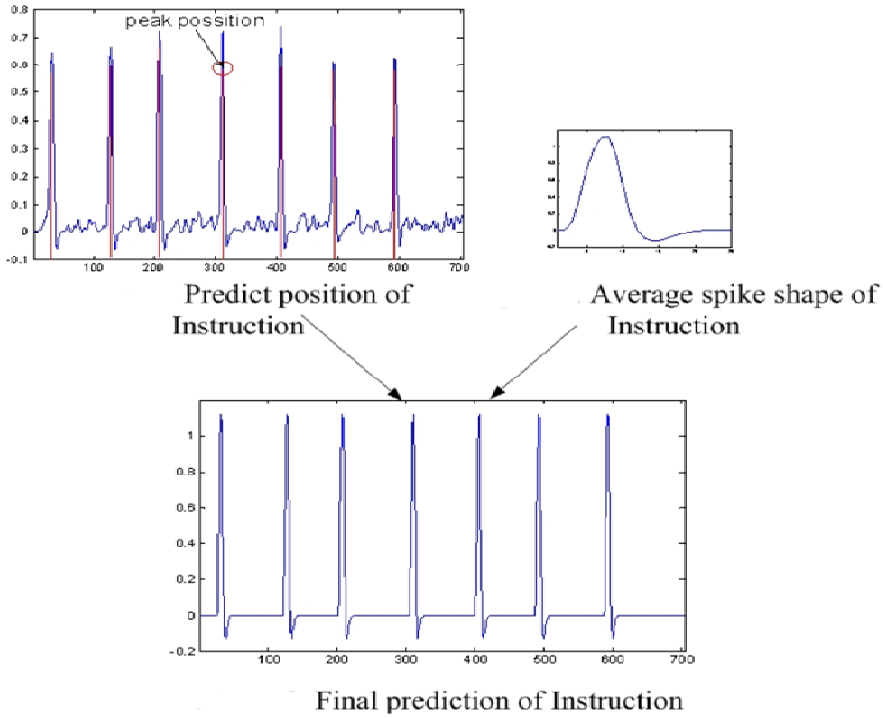


Figure 4.10 Model fitting for predicting "Instruction"

This figure shows the model fitting approach to boost the prediction of "Instruction". The top left graph shows the original prediction. The average shape of the response of "Instruction" was generated to fit the raw prediction. The bottom figure shows the fitted model of final prediction. This procedure removed the small noise between each "spike" and boosted the prediction from 0.8 to 0.99.

We also utilised the insights we found about the design of the VR game to achieve correlations in excess of 0.99 for "search fruit", "search weapons", and "search people". In fact, those three ratings were not predicted in the same way as other ratings, and the "search" ratings in the first and second run of the virtual reality games were not actually used for training the regression machine. To achieve such high accuracy, prior knowledge was utilised to arrive at a solution from predictions of more accurate ratings. From observation, a few repeating patterns were found in the design of the virtual reality (VR) game. These could be seen as a weakness in the design, and we exploited them as far as possible.

1. Each run of the game was cut into seven slots, each of which started at the

"instruction" and ended with the "instruction" - except for the last slot.

2. Each search request (fruit, weapons, and people) appeared after the instruction and occupied exactly four arbitrary slots at each run.
3. Each slot contained at least one request.
4. The requests to search the three categories were in the same order for all subjects.
5. The optional ratings of "hit (people, weapons, and fruit)" only appeared when the slot had the same category of search request, otherwise the rating of hit something would be zero for the whole slot. E.g. If during a particular time slot, the search requests were "search people" and "search weapons", then the rating of "hit fruit" would be zero for the entire slot.

From cross-validation, we found that training with the "search (people, weapons, and fruit)" rating returned very low accuracy, with correlations of only about 0.2. However, predicting "hit (people, weapons, and fruit)" could achieve correlations of 0.4~0.5, for at least one of the subjects. Hence, we used the prediction of "hit something" to predict "search something". For example, if we knew during one slot "hit people" is non-zero (after removing the noise) in at least one scan, there would definitely be a "search people" request in that slot. Mathematically, although the prediction of "hit something" would of course contain noise, the strength of the noise was believed to be lower than that of "true hits". Therefore, if we threshold the prediction and only kept high peaks, most noise would possibly be pruned out, but kept enough true "hit something" to infer which slots had the particular category of search request. Motivated by this observation and by previous findings, the procedure can be summarised by the following steps (figure 4.11)

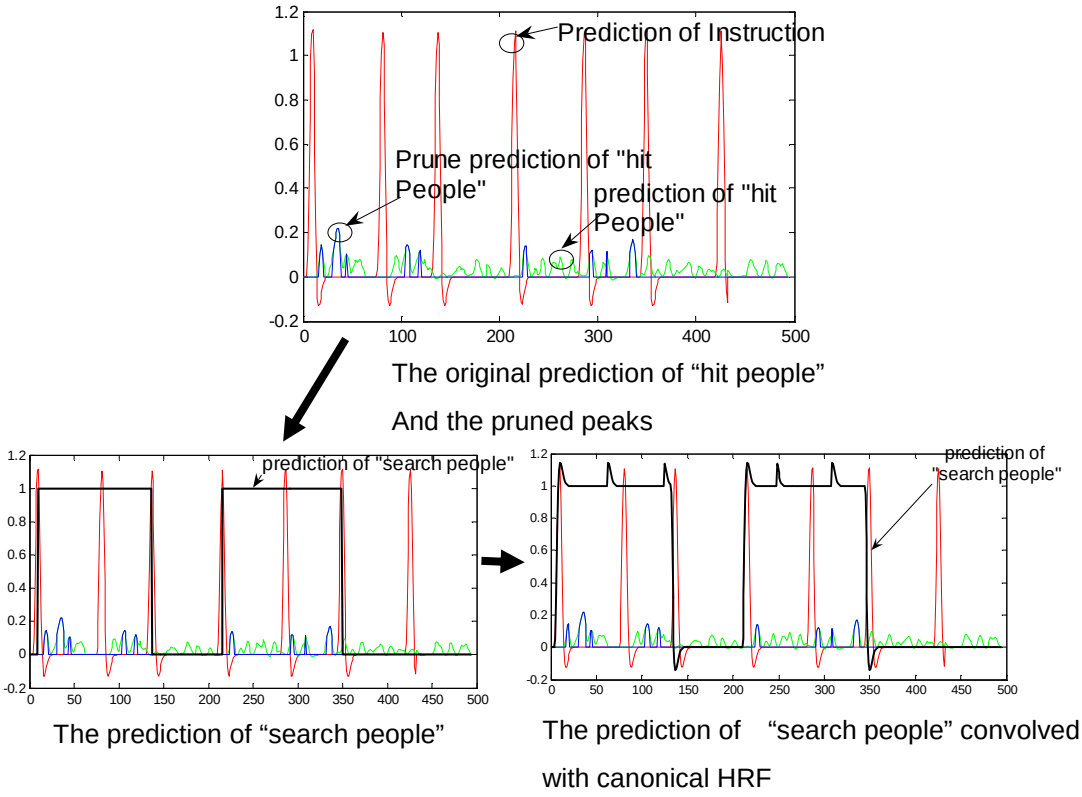


Figure 4.11 Illustration of how to predict “search people”

These figures show how to predict “Search people” using prediction of “hit people. The red spikes are the prediction of “instruction”. For simplification, the fixation (resting) period was removed. The “instruction” divided each run into 7 slots. The prediction of hit people is shown in green, and the threshold version of hit people is shown in blue. 4 out of the 7 slots were elected based on the strength and frequency of threshold hit people. In this case they are slot 1 2, 4, and 5, then the rating was set to 1 for time points in those 4 slots. The prediction was furbished by convolved with the HRF.

1. Predict “hit (people, weapons, and fruit)” for three subjects.
2. Prune most of the points and only keep some high value peaks (top 20%).
3. Count how many peaks are in each slot. For each “search something” request, we found the four most possible slots, which contain the highest counts of peaks. If peak distributions were different among the three subjects, a majority vote was used.
4. The rating of the “search something” was set to one during the four slots.
5. Finally, the predicted block was convolved with the canonical HRF.

Unlike most conventional fMRI studies, which provide controlled external stimuli, some of the ratings were self-paced, such as “hits” or “velocity”. It was believed that those ratings may have different HRF delays from the canonical HRF. The stringent way would have been to train with ratings convolved with different HRF parameter settings, but there are at least five parameters to adjust for generating the HRF using a double gamma functions. For the reason of generalisation and robustness, we simply applied forward or backward shifts by discrete numbers of time points (scans). The predicted rating was later inversely shifted. It was found, by cross validation, that shifting the original training target $\{\mathbf{t}_i\}_{i=1}^{N}$ by one scan (1TR) earlier would yield more accurate predictions. The shifted training target would be $\{\mathbf{t}_j\}_{j=1}^{N}$, where $\mathbf{t}_j = \mathbf{t}_{i+1}, \mathbf{t}_N = 0$. The following tables show the results of cross validation for “Hits”, “Velocity”, and “Faces”.

Hits	Subject 1		Subject2		Subject3	
	Predict VR1	Predict VR2	Predict VR1	Predict VR2	Predict VR1	Predict VR2
Original	0.5873	0.6861	**0.7427**	**0.8030**	0.6019	**0.7551**
Apply shift	**0.6094**	**0.7272**	0.735	0.8	**0.6096**	0.7341

Faces	Subject 1		Subject2		Subject3	
	Predict VR1	Predict VR2	Predict VR1	Predict VR2	Predict VR1	Predict VR2
Original	0.5538	0.5436	0.589	0.7521	0.8313	0.8706
Apply shift	**0.5549**	**0.553**	**0.7114**	**0.8155**	**0. 8328**	**0.8859**

Velocity	Subject 1		Subject2		Subject3	
	Predict VR1	Predict VR2	Predict VR1	Predict VR2	Predict VR1	Predict VR2
Original	0.7217	0.7207	0.7010	0.6347	0.664	0.7022
Apply shift	**0.7432**	**0.7312**	**0.7481**	**0.6464**	**0.7059**	**0.7508**

In these three feature ratings, only “Hits” did not show consistent improvement across all three subjects. “Velocity”, and “Faces”, both showed increasing accuracy for all subjects. This led us to ask why it might be that measured brain activity preceding an event would appear to be more predictive. It might be possible that the regions involved in those two ratings had an HRF that was considerably shorter for these regions than for other ratings. The alternative explanation is that brain activity preceding the event reflects what is subsequently recorded. The “Velocity” rating is related to the amplitude of joystick movement, so the involvement of processes underlying voluntary motor control would be expected. Motor preparation, or the readiness potential, has been known to precede onset of voluntary motor execution by over a second. This would conceivably correspond to the period of 1 TR.

Above all these additional improvements, we also used the standard “quadratic deconvolution” and smoothing (figure 4.5), which were identical to the post-processing performed in PBAIC 2006. In general, the standard procedure involved a linear kernel using a GM mask. The details and additional procedures for predicting each rating are summarised in the following table.

Rating	Arousal	Valence	Hits (items)	Search people	Search Weapons	Search Fruit
Additional Procedures	RBF kernel	RBF kernel	None	Information from predicting “hit people”	Information from predicting “hit weapons”	Information from predicting “hit fruit”

Instructions	Dog barks	(seeing) Faces	(seeing) Fruits Vegetables	(seeing) Weapons Tools	Interior or Exterior	Velocity
Model fitting with averaged template	Auditory cortex mask	Temporal shift	None	None	Visual cortex mask	Temporal shift

4.3.4 Our result in PBAIC 2007

PBAIC 2007 also allowed each team to submit the prediction three times. The results of the first two submissions would be returned to the team, and the best score out of the all submissions would be ranked with other teams. Our strategy was rather simple: we submitted the first submission with predictions by KRR, and the second submission with predictions by RVR. Both submissions included those additional procedures. Our final submission was based on selecting the best results from the first and second submissions (figure 4.12).

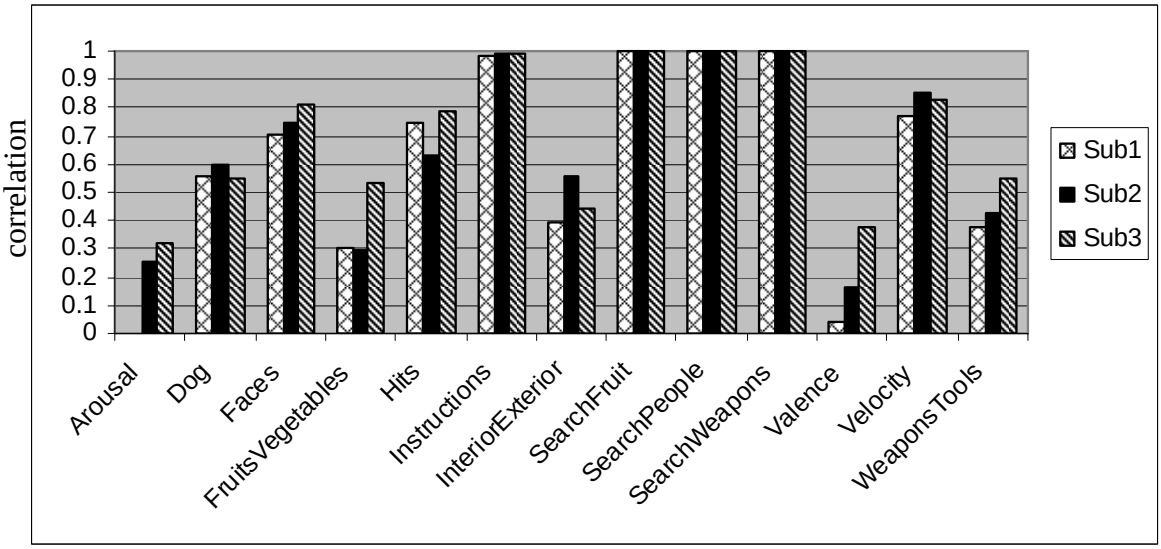


Figure 4.12 Results of our final submission

This figure shows prediction accuracy of our final (third) submission for all three subjects. We achieved nearly perfect prediction for "Instructions", "Search fruit", "Search people", and "Search weapons".

In general, subject one had the worst prediction accuracy (average Z score 0.980),

especially for emotional and subjective ratings such as “Arousal”, “Valence”. Subject three had the best overall prediction accuracy with average Z score 1.142. The variation of prediction accuracy for each rating across all subjects is quite consistent; i.e. subject one is often the worst. This implies that accuracy is influenced by subject-specific issues. This may relate to concentration, but was most likely due to motion in the scanner. By inspecting the movement parameters generated from the realignment procedure, subject one clearly showed more translation and rotation than subjects two and three. Our ability to predict particular ratings were clearly higher for objective ratings such as instructions, velocity and faces than they were for subjective ratings. We believe this is related to the reliability of the reported ratings (many of the subjective ratings were made at a separate occasion based on episodic recall of how they felt), and that this will improve if real-time measures such as skin conductance and heart rate or subjective ratings between each block were used instead. Among objective ratings, we were able to best predict those that involved attention or required a response on the part of the subject. Thus “instructions” required the subject to attend and comprehend, while “velocity” and “hits” required a motor response from the participant. These were followed by anthropomorphic objects such as faces.

As the 1st place winner in 2007 PBAIC competition, our final competition score was 0.785 which was substantially higher than that of other groups. Generally speaking, our team predicted all the objective ratings well within the top 5% of the maximum correlation for the entry, and we had the best prediction over the three subjects for “Hits”, “Search People”, “Search Weapons”, “Search Fruit”, “Faces”, “Fruits Vegetables”, and “Velocity”. (Figure 4.13) However, our method did not perform well for the subjective ratings, which were “Arousal” and “Valence”. It is probably because our team used the whole gray matter, and results from groups which did feature selection seemed to perform better for those two ratings. In addition, we

used RBF kernels to predict “Arousal” and “Valence”. Cross validation showed that linear kernels performed poorly for those two ratings. Linear methods are only able to model a single mode of difference, whereas nonlinear models can potentially model multiple modes of variability. This may indicate that these states may be represented in the brain by several alternative networks of activity, rather than a single consistent pattern of differential activity.

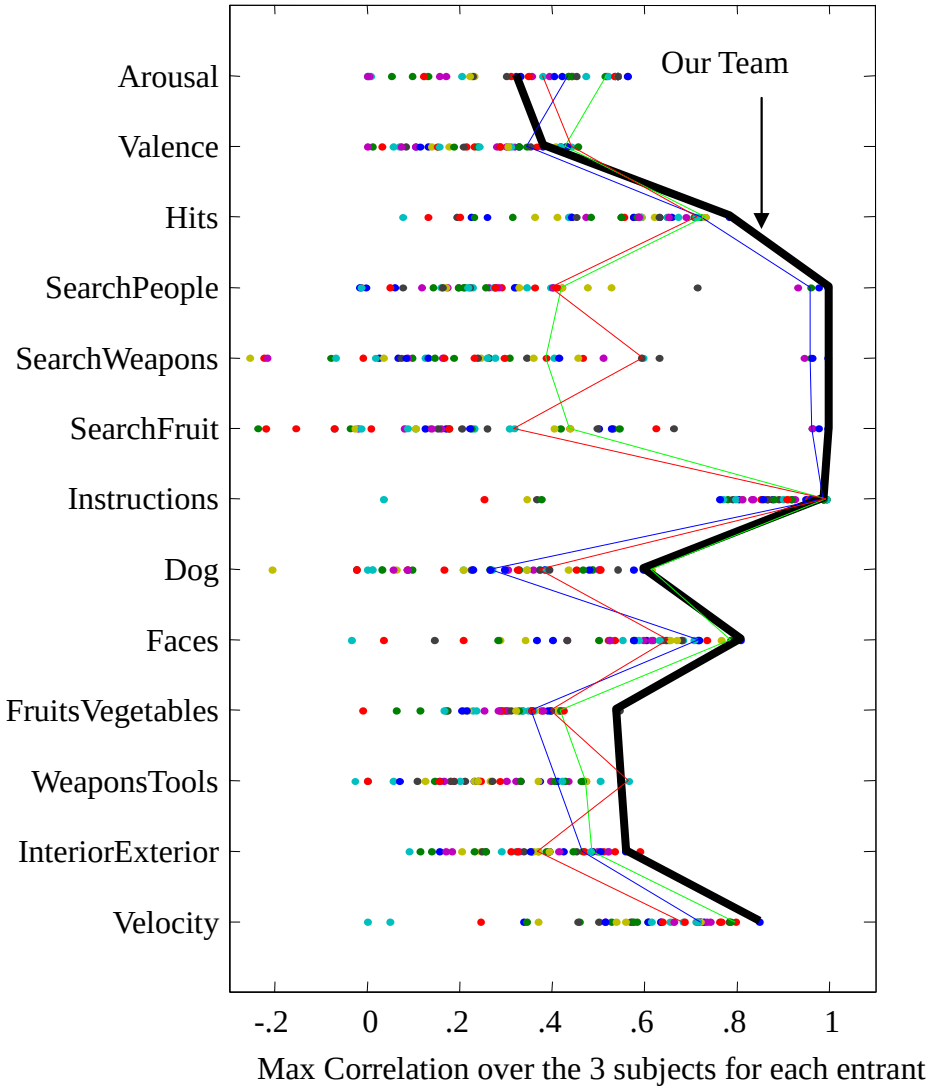


Figure 4.13 Our best results compared with other teams.

This figure shows the prediction accuracy of our final submission. We achieved nearly perfect prediction for “Instructions”, “Search fruit”, “Search people”, and “Search weapons”.

4.3.5 Overall discussion of PBAIC 2007

Relevance Vector Machine VS Kernel Ridge Regression

On average, kernel ridge regression (KRR) performed slightly better than relevance vector regression (RVR), but the results are mostly within 10% of each other. In the following table, we compared the results of KRR and RVR for predicting the third run of subject 3, using a linear kernel.

	Velocity	Hits	Weapons Tools	Fruits Vegetable	Faces
KRR	0.8277	**0.7835**	**0.5470**	**0.5366**	**0.8091**
RVR	**0.8309**	0.7552	0.4998	0.4955	0.7995

In addition, the sparseness of RVR (percentage of the training scans contributing to the prediction i.e. (number of relevance vectors)/(number of training samples)) is presented in the table below.

	Velocity	Hits	Weapons Tools	Fruits Vegetable	Faces
RVR Sparsity	21.3%	24.4%	22.8%	23.3%	18%

As we observed, KRR performed slightly better for most ratings. It is possible that sparse representations may not fully utilise all the information in the training set; hence pooling all the training scans would probably estimate the variance component more accurately. However, from the table above, RVR only required less than 25% of the training data to make predictions, with less than a 10% sacrifice in accuracy. For ratings which could be predicted well, such as “Velocity” and “Faces”, the differences between RVR and KRR are only about 1%. This sparsity may be due to consistent activation patterns in the brain during the same ratings; hence the regression machine only required a subset of the training data to represent it. It is also possibly due to

irrelevant training scans for particular ratings. Because the nature of this “experiment” is self paced, not all the scans contained relevant information for particular ratings. For general fMRI studies, sparse methods have the advantage of providing more interpretable results. By looking at the time point of scans with non-zero weights, researchers may gain some understanding about the temporal pattern of task activation.

Importance of pre-processing

We believe that one of the winning factors for our team was good spatial and temporal pre-processing models. Spatial smoothing and temporal detrending had been shown to change the results of SPM as well as the prediction accuracy (LaConte et al., 2003; LaConte et al., 2005; Strother, 2006; Tanabe et al., 2002) (figure 4.9). One major reason why temporal detrending is important is because scans from the all three games were combined together. In other words, all the scans were assumed to be collected in the same session with the same intra-sessional variance. If the low frequency components dominated the major variance components, i.e. the first few principle components, the signals due to brain activations would be reduced. In general though, detrending with eight DCT bases, with spatial smoothing (6mm FWHM Gaussian) gave the best results for most ratings. The improvement was most prominent for “Hits” and “Velocity” (figure 4.9). In figure 4.13, it shows that our prediction performed specifically better than other groups for those two ratings. We speculate that a lot of teams used the pre-processed dataset provided by the competition committee, which still contained large amount of low frequency noise. The noise eventually caused the inferior performances for those teams.

Comparing the regularisation parameter λ of KRR obtained by cross validation or GPR

Unlike RVR, where the hyper-parameters and parameters can be determined through maximisation of marginal likelihood, the regularisation parameter for KRR had to be determined empirically by cross validation. In figure 4.14, the correlations of training VR game 1 then testing on VR game 2 and the vice versa were evaluated with different regularisation parameter for four feature ratings. The graph showed that the correlation reached a plateau with the regularisation roughly between 10^2 and 10^5. Both predicting VR1 and VR2 had a consistent shape to the plateau. Alternatively, it is possible to find the parameters by maximising the marginal likelihood (4.23), with the Gaussian Processes Regression model having the covariance function $\mathbf{C} = \theta_1 \mathbf{I} + \theta_2 \mathbf{K}$. The regularisation of KRR can be computed by $\lambda = \theta_1 / \theta_2$. Intriguingly, in figure 4.14, it seemed the regularisation determined by maximising marginal likelihood was over-regularised, and the results were not very desirable. This may reveal the importance of well specified models to the application of Bayesian techniques. If a good model structure is not accurately known, then cross-validation may allow more accurate tuning of various hyper-parameters than the Bayesian evidence framework. In our case, the less accurate solution found by GP may due to several factors, firstly, no temporal autocorrelations were modelled, whereas the true noise for fMRI data is not independent and identically distributed (iid). Secondly, the objective function of marginal likelihood is based on the sum of the squares differences, which may have different characteristics from Pearson's correlation. Thirdly, a proper covariance matrix, **C**, should contain a constant term $\mathbf{C} = \theta_1 \mathbf{I} + \theta_2 \mathbf{K} + \theta_3$. From our experiment, including the constant term actually improved the correlation to around the same accuracy as the plateau in the cross-validation plot.

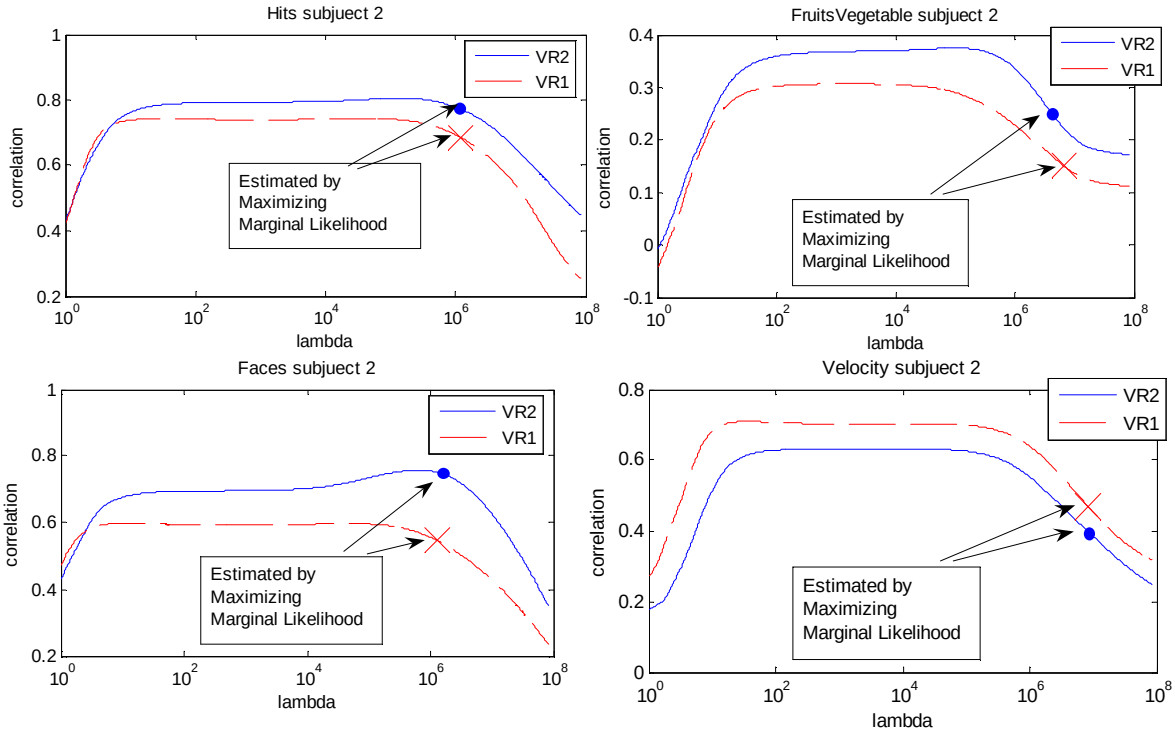


Figure 4.14 Determine regularization for KRR

This figure shows the cross validation results for subject two, using KRR to predict four ratings-"hits", "FruitsVegetable", "Faces", and "Velocity". The horizontal axis indicates different amount of regularisation for KRR. The plotted line of VR1 means the prediction of the first run by training the second run, and vice versa. The dot is the prediction of VR1 estimated via maximising of marginal likelihood with GPR, and the cross is the prediction of VR2. GPR based on maximising the marginal likelihood (evidence) seemed to slightly over-regularised the model, except for the rating "Faces".

In addition, we also generated the "weight map" for visualisation purpose. The weight map of "Velocity" is particularly interesting. Inspection of the weighting in voxel space (figure 4.15) shows that the motor areas around M1, the supplementary motor area and cerebellum had activity positively weighted with ratings. This is evidence supporting our assumption about motor preparation, or the readiness potential preceding onset of voluntary motor execution (Cunnington et al., 2002), and may explain why temporal shift of 1 TR could improve the prediction accuracy of "Velocity".

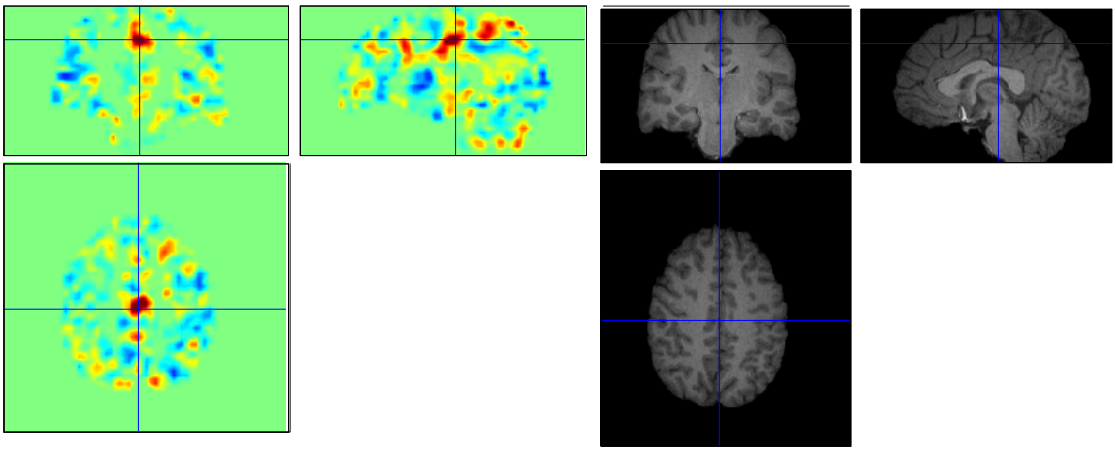


Figure 4.15 Weight map of "Velocity"

This figure shows a weight map of "Velocity" for subject 3. There are strongly positive weightings in the motor areas. This is evidence for our assumption about motor preparation preceding onset of voluntary motor execution, as "Velocity" was related to navigating around the VR game using a joystick.

4.4 Application: Regression Analysis for Clinical Scores of Alzheimer's Disease Using Multivariate Machine Learning Method

4.4.1 Introduction

In this study, we demonstrated that RVR can not only be applied to functional data, but it can also be applied to structural MRI data to predict clinical ratings. Voxel-based morphometry (VBM) studies have shown relationships between cognitive measures and structural differences in MRI (Baxter et al., 2006; Jack et al., 2008). However, VBM or any univariate method aims to localise regional atrophy, rather than characterise the pattern of difference due to atrophy. Physiological biomarkers have been shown to have good accuracy for detecting early Alzheimer's Disease (AD) (Ray et al., 2007). In practice, the combination of cognitive tests and

imaging may help physicians to decide the disease state of patients, since both cognitive and imaging changes are shown to be associated with the early stages of AD (Caselli et al., 2007; Twamley et al., 2006).

The main objective of this study was to examine the relationship between structural changes and clinical ratings, in the framework of pattern classification. Our hypothesis was that global GM patterns have a linear relationship with clinical scores. Using the probabilistic regression model (RVR), we can compute the conditional distribution of clinical ratings, given the structural images. We applied RVR to predict clinical ratings from structural MR, where the ratings were three commonly used cognitive measures: the Mini-Mental State Exam (MMSE) (Folstein et al., 1975), the Dementia Rating Scale (DRS) (Mattis, 1988) and the Auditory Verbal Learning Test (AVLT) (Rey, 1964). We compared the predicted clinical ratings with the true ratings using correlation (similar to the PBAIC scoring). Because all three clinical scores have different ranges and scales, it would be difficult to compare the predicative accuracies across different scores using mean square error (MSE). The advantage of using correlation is its scale invariance. On the other hand, correlation is also invariant to the bias of the prediction, which may be a less desirable feature.

4.4.2 Material and Methods

The dataset contained structural MRI scans of 73 patients with probable AD (clinically defined), ranging from mild to severe (MMSE from 10 to 30, mean 22.3) and 91 cognitively normal controls from the Mayo Rochester Alzheimer's Disease Research Centre (ADRC) and Mayo Alzheimer's Disease Patient Registry (ADPR) (Petersen et al., 1990). There were eight controls with MMSE less than 27. Because our study aimed at comparing clinical scores, rather than classifying AD or non-AD, we did not exclude controls with subclinical disease. In fact, both controls and AD patients were assumed to be from the same population in this study. All subjects had

MMSE, DRS, and AVLT scores recorded within three months of the MRI scans.

MR scans were collected from 14 different scanners over about 10 years. All scanners were the same model (General Electric Signa 1.5T), and the scans were acquired using T1-weighted imaging sequences (parameters: TR=17.7 to 27 ms, TE=6 to 10 ms, flip angle 25 degrees or 45 degrees, voxel size 0.86 mm x 0.86mm x1.6mm). A VBM analysis showed no significant interaction of scanner with the effect of disease (Stonnington et al., 2008).

We pre-processed the data using standard methods described in section 3.2.1. T1-weighted scans were firstly segmented into GM and WM tissue class images, which were imported for use with the DARTEL toolbox. This involved bringing them into the closest rigid body alignment with each other and resampling to isotropic 1.5mm voxels. These data were then DARTEL registered with the population template, as described in section 3.2.1. The nonlinearly aligned GM images were scaled by their corresponding Jacobian determinant maps, such that tissue volumes were conserved. No spatial smoothing was applied in this study.

We generated a linear kernel from whole GM, as well as, a kernel from the combination of two volume of interests (VOI: dimensions 12, 16, 12 mm in x,y,z directions respectively) centred around both left and right parahippocampus (equivalent to x,y,z =-17, -8, -18, and 16, -9 -18 in the population template space, it is approximately the same as in MNI space). The VOI was motivated by the findings that the earliest pathological changes of AD are found in the entorhinal cortex and hippocampus (Braak and Braak, 1991).

We applied standard leave-one-out cross validation to predict the clinical ratings. We left one subject out of the full set, and trained the remaining 163 with RVR. The trained RVR was then applied to predict the clinical scores of the subject left out. The training procedure was similar to that used for PBAIC, as we use the same kernel as

input, and three different clinical scores as target variables.

4.4.3 Results and discussion

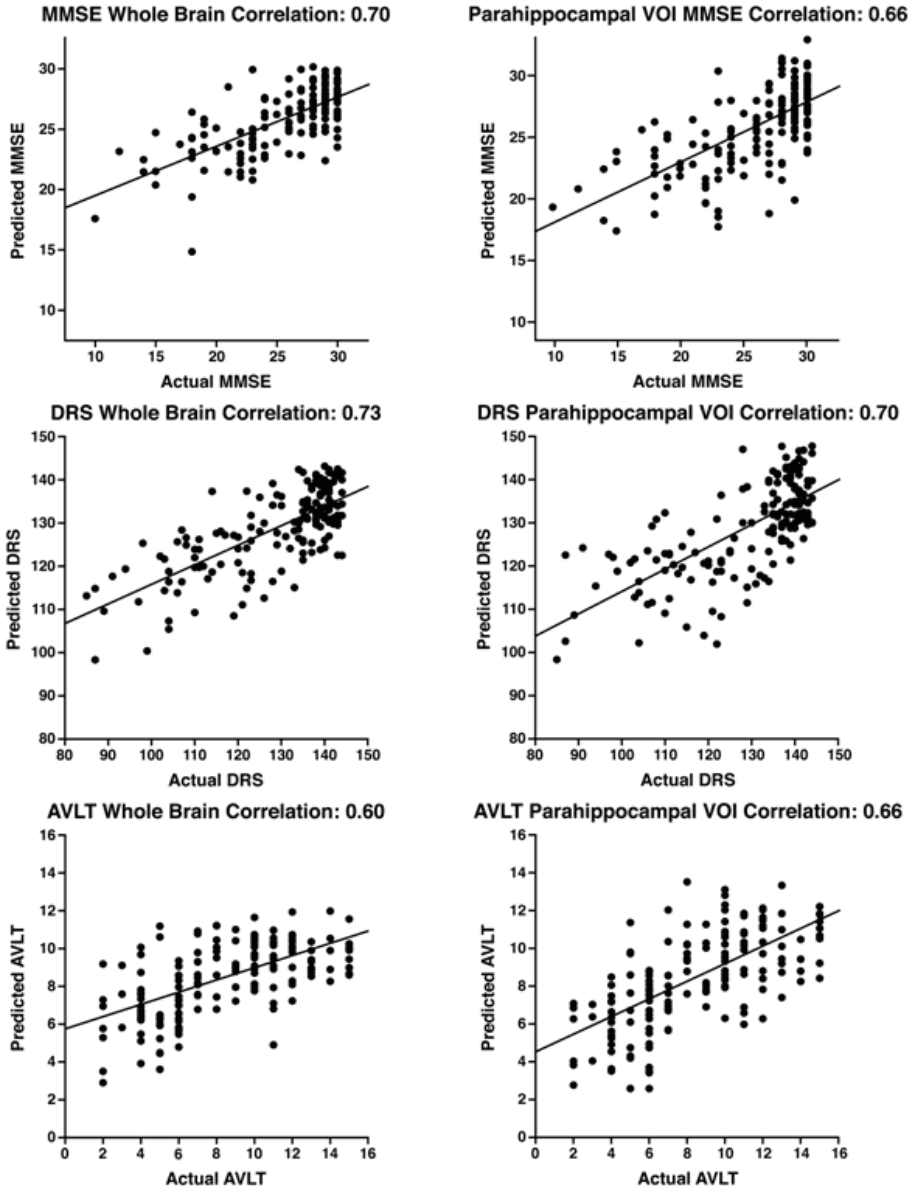


Figure 4.16 Predictions for three clinical ratings

These plots show the predicted scores versus actual scores for 3 different clinical ratings (MMSE, DRS, VALT) using whole brain grey matter (left column) and parahippocampal VOI (right column)

For whole brain GM, the correlations of predicted and real scores were as follows: MMSE: 0.70; DRS: 0.73; AVLT: 0.60. When we applied the

parahippocampal mask, the correlations were less good for MMSE (0.66) and DRS(0.7), but better for AVLT (0.66) (figure 4.16).

The result show strong linear relationships between all clinical scores and the structural MRI, with DRS showing the highest prediction accuracy. Both MMSE and DRS showed significantly better prediction results with whole brain data, because MMSE and DRS test multiple domains. It is also understandable why the accuracies dropped when we trained using only the parahippocampal VOI. Conversely, AVLT mostly tests the single domain of memory, which is associated with medial temporal lobe function. This can explain why the prediction accuracy of AVLT improved using only the parahippocampal VOI. From the weight maps (figure 4.17), we see that almost no regions, apart from the medial temporal lobe, were important for predicting AVLT. The DRS was found to be slightly more sensitive than MMSE to fit with disease progression (Galasko et al., 2000). Our results confirmed the findings, and we also agree that this is probably due to the ceiling effect of MMSE, as MMSE is capped at 30.

One practical use for such a machine learning approach would be to aid in clinical management. Bayesian machine learning methods can compute the predictive distribution, and not only a point-estimate. For tests showing good correlation with the MRI scans, such as MMSE or DRS, if an individual's scan predicts a significantly lower score than the actual score, this may imply that the patient may have greater resilience of cognitive reserve. In our dataset, one subject scored within the normal range in the three tests (MMSE 30, DRS, 135, AVLT 11), however the predicted scores (MMSE 24, DRS, 121, AVLT 7) were similar to the MCI patients. As has been shown (Mortimer et al., 2005), cognitive reserve may mask cognitive deficits, and prevent the detection of early AD using only clinical ratings. Therefore, a combination of both clinical measures and imaging information may be used to more accurately

predict the clinical state of any individual. On the contrary, if the actual score is significantly lower than the predicted score, there may be other factors affecting the performance of the tests, such as fatigue or depression. In addition, if longitudinal scans are collected and used optimally, the predicted rating should provide a more robust measure of disease progression.

To observe the confounding effect due to age, we also applied RVR with age removed using equation (3.14) described in section 3.2.1. This actually improved the prediction slightly, for whole brain GM, MMSE: 0.72, DRS: 0.76, AVLT: 0.63), and for the parahippocampal VOI, MMSE: 0.74, DRS 0.74, AVLT: 0.69.

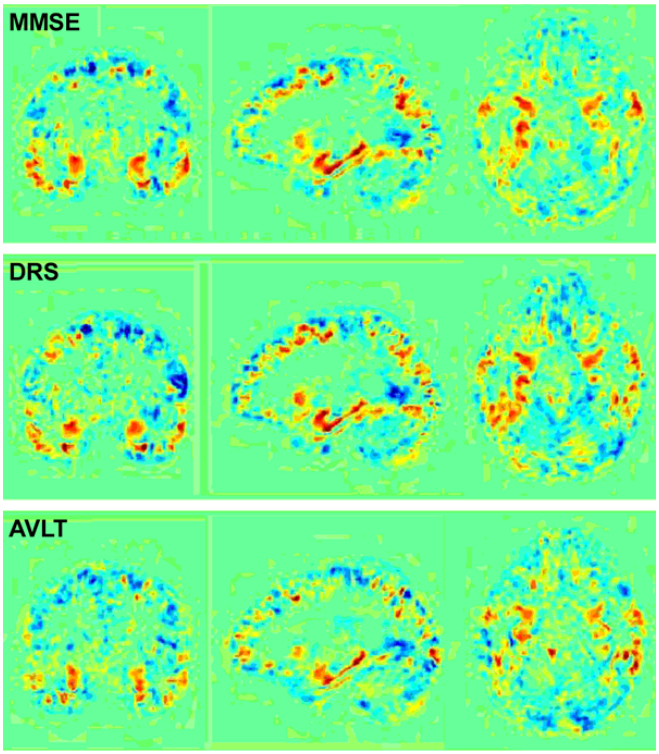


Figure 4.17 Weight maps for three clinical ratings

Weight maps for whole brain images of MMSE, DRS, AVLT, reflecting areas of the brain most vital in determining each predicted score. Because all three scores have different scales, we normalised the intensity range of the weight map for visualisation purpose. The red areas indicate where more grey matter adds to the predicted score, whereas blue areas indicate areas where more grey matter subtracts from the predicted score. Note that dementia does not necessarily cause increased volumes of grey matter in these blue areas, but these regions may have relatively less atrophy for dementia patients.

Chapter 5

Kernel Classification Methods and the Application in Functional and Structural MRI

Contents

In this Chapter, we introduce the methodological aspects of kernel classification, discussing Support Vector Classification (SVC), Relevance Vector Classification (RVC), Gaussian Processes Classification (GPC), one class classification, and a multiple class classification I proposed by combining a regression method with a decision function. Also, the projects employing those methods for both fMRI and structural MRI data will be presented in details. The fMRI related works are:

- "Decoding Neuronal Ensembles in the Human Hippocampus" (Hassabis et al., 2009), which was a collaborative work with Demis Hassabis.
- "Multi-class Classification of fMRI pattern by Kernel Regression Methods", which was chosen for oral presentation in the OHBM conference of 2008.

The projects with structural MRI are:

- "Automatic classification of MR scans in Alzheimer's disease" (Kloppel et al., 2008b; Kloppel et al., 2008c; Kloppel et al., 2008d), which was a collaborative work with Dr. Stefan Kloppel and Dr. Cynthia Stonnington, using data from Dr. Clifford Jack at the Mayo Clinic.
- "Automatic detection of preclinical neurodegeneration: presymptomatic Huntington disease" (Kloppel et al., 2009), which was a collaborative work with Dr. Stefan Kloppel.
- "Classification of major depressive disorder using SVM with structural MRI", where I collaborated with Dr. Cynthia Fu at the Institute of Psychiatry, Kings College, London.

The general framework for classification was introduced in section 2.4, whereby a training set contains input/output pairs, $S = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ..., (\mathbf{x}_N, t_N)\}$, and t is a discrete label. For binary classification, the labels are often denoted by $\{-1,1\}$ or $\{0,1\}$. The general model for a binary classification problem is

$t = (\sum_{d=1}^{D} w_d x_d + offset) > 0$, which is closely related to the linear regression problem, except the labels are separated by the decision boundary. The general model for a kernel binary classification is $t_* = (\sum_{i \in training} \beta_i K(\mathbf{x}_*, \mathbf{x}_i) + offset) > 0$. Both SVC and RVC are sparse kernel machines, which implies that some of the kernel weights, β, are zero. In other words, not all training samples contribute to the prediction of the testing samples. Similar to linear kernel regression models, when we apply the linear kernel, we can obtain the weights in the input feature space by $\mathbf{w} = \sum_{i=1}^{N} \beta_i \mathbf{x_i}$.

5.1 Introduction to Kernel Classification Algorithms

Generally speaking, RVC and GPC are probabilistic classifiers, which are closely linked to logistic regression, (section 2.4.6) i.e. they use a generalized linear model by applying a sigmoid link function. Therefore, both RVC and GPC can be extended into multiple classes by assuming a multinomial distribution. On the other hand, SVC was motivated by the statistical learning theory (Vapnik, 1998), is not a probabilistic model, and was originally formulated as a binary classifier. Usually when people refer to a Support Vector Machine (SVM), they often mean Support Vector Classification (SVC). In this section we will present the classification aproaches in the same order as the regression models of chapter 4, introducing the non-probabilistic SVC first, then RVC and GPC later.

5.1.1 Support Vector Classification

SVC is also known as the maximum margin classifier (Bishop, 2006b; Cristianini and Shawe-Taylor, 2000), and has gained in popularity in recent years because of its superior performance in practical applications, especially in the field of bioinformatics (Brown et al., 2000; Guyon et al., 2002; Noble, 2006). The philosophy behind the SVM is to estimate an optimal solution based on “structural risk

minimization" rather than "empirical risk" minimization. Motivated by statistical learning theory (Vapnik, 1995), the decision boundary is chosen so it achieves the maximum margin. The margin is defined as the distance between the decision boundary and the closest data points.

We begin our introduction of the simple binary SVC model, by assuming both classes are linearly separable. Let us define a training set $S = \{(\mathbf{x}_i, t_i)\} \in \Re^d \times \{-1, 1\}$. The objective of the training is to find the linear decision boundary, or a hyperplane in the feature space, that maximizes the margin. By a hyperplane we mean a set $H_{\mathbf{w},b} = \{\mathbf{x} \in \Re^d : \mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0\}$ parameterized by a vector $\mathbf{w} \in \Re^d$ and a scalar b. In other words, the hyperplane is in the subspace of $\Re^{d-1}$. The hyperplane is a line when the feature space is 2D and a flat surface when the feature space is 3D. When the two classes are totally separated, the following equation $t_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) > 0,\ i = 1, ..., N$ is satisfied. This is to say, the data points labelled with 1 will all be positive from $\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b$, and data points labelled with -1 will be all negative. The distance of a point, $\mathbf{x}_i$, to the hyperplane is $\frac{|\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b|}{||\mathbf{w}||}$, and the margin is defined by $\min_{i=1}^{N} \frac{|\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b|}{||\mathbf{w}||}$

Unlike the case of regression, the vector **w** is scale invariant in the context of classification; hence the solution of **w** and b is not unique in that any vector **w**, which is perpendicular to the hyperplane, is a valid solution. There are two ways to reformulate the parameterization to obtain a unique solution for each decision boundary. One way is to normalize **w** so that $||\mathbf{w}||=1$, the other way is to choose $||\mathbf{w}||$ such that the distance of the margin is defined by $\frac{1}{||\mathbf{w}||}$ i.e. $\min_{i=1}^{N} t_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)=1$.

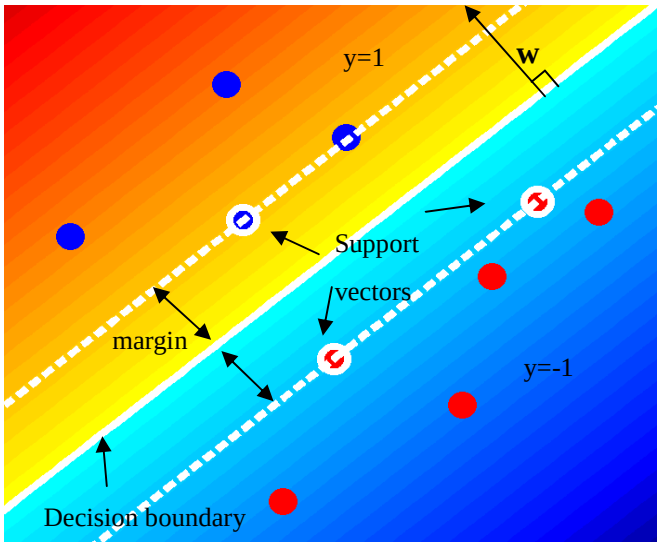


Figure 5.1 Hard-margin SVC

This figure is a 2D example of a linearly separable support vector classification. The blue and red spheres are the data points for different classes. The spheres enclosed by white circles are the support vectors, which are the data points closest to the decision boundary (hyperplane). In this formulation, there are no data points between the margins.

We will work on the second parameterization, where the optimization problem can be formulated as

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &\text{subject to} && t_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1, i = 1..., N \end{aligned} \tag{5.1}$$

This is often known as the primal form of the SVC optimization. To solve the constrained optimization problem, Lagrange multipliers, a_i, are introduced. The Lagrangian function is defined as

$$L(\mathbf{w}, b, a) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{i=1}^{N} a_i\{t_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) - 1\}, \qquad 0 \leq a_i \tag{5.2}$$

To solve the optimization, we set the derivative of the Lagrangian function with respect to $\mathbf{w}$ and b to zero, $\frac{dL}{d\mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} t_i a_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{N} t_i a_i \mathbf{x}_i$, $\frac{dL}{db} = \sum_{i=1}^{N} t_i a_i = 0$. Substituting these new conditions back to (5.2), leads to the dual form of the optimization, where $L(\mathbf{w}, b, a) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{i=1}^{N} a_i = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j t_i t_j \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j + \sum_{i=1}^{N} a_i$

$$\begin{aligned} &\text{Maximize} \quad -\frac{1}{2}\mathbf{a}^{\mathbf{T}}\mathbf{H}\mathbf{a}+\sum_{i=1}^{n}a_i \\ &\text{subject to} \quad \sum_{i=1}^{n}t_i a_i = 0 \\ &\qquad\qquad a_i \geq 0, \quad i=1,...,N \end{aligned} \tag{5.3}$$

where **H** is a *N* by *N* matrix . More generally we replace $\mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$ by the kernel $K(\mathbf{x}_i,\mathbf{x}_j)$ (see Chapter 3), and define $h_{i,j}=(t_i t_j K(\mathbf{x}_i,\mathbf{x}_j):i,j=1,...,N)$. Equation (5.3) is a typical constrained quadratic programming optimization problem. Although general purpose quadratic programming can be used(5.3), the most popular algorithm to solve this specific optimization is called sequential minimal optimization (SMO) (Platt, 1999a). After finding the optimal Lagrange multipliers, we can compute the weight in the feature space from the condition derived previously, $\mathbf{w}=\sum_{i=1}^{N}a_i t_i \mathbf{x}_i$. During the optimization, only some Lagrange multipliers have values greater than zero, and their corresponding data points are called support vectors (SVs). The decision boundary is defined only by those SVs, hence removing non-SVs from the training would yield identical solution from the original training. Theoretical investigations show that the proportion of SVs in the training set reflects an upper bound of the expected generalization error (Vapnik, 1998). The parameter *b* can be determined by finding the *b* that satisfies $t_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i+b)=1\Rightarrow b=t_i-\mathbf{w}^{\mathbf{T}}\mathbf{x}_i$ if $\mathbf{x}_i$ is a SV (notice $t\in\{-1,1\}$, so $\frac{1}{t}=t$), because the distance of the SV to the decision boundary is 1. More generally, if we are using the kernel trick and are unable to compute the weight vector **w** in the feature space, we can find *b* by satisfying

$$\begin{aligned} &t_i(\sum_{j=1}^{N}a_j t_j K(\mathbf{x}_j,\mathbf{x}_i)+b)=1 \\ &\Rightarrow b=t_i-\sum_{j=1}^{N}a_j t_j K(\mathbf{x}_j,\mathbf{x}_i) \end{aligned} \tag{5.4}$$

where $\mathbf{x}_i$ is any SV. In practice, *b* is estimated from all SVs and then averaged for

numerical stability. The decision function for predicting new input point is given by

$$y(\mathbf{x}_*) = \sum_{i=1}^{N} a_i t_i K(\mathbf{x}_i, \mathbf{x}_*) + b \tag{5.5}$$

This value measures the distance of the new data point to the decision boundary, in units of the margin width. Intuitively, the maximum margin formulation implicitly means only a subset contributes to the solution of the hyperplane. Trivial cases, which are far away from the opposite class, will not be used to calculate the decision boundary, and only "ambiguous" cases, those samples that are closer to the other class, will contribute to constructing the decision function (5.5). This may not be desirable if there are outliers that are close to the opposite class or mislabelled data points. To relax the formulation of the “hard margin” SVC, the “soft margin” SVC introduces space for some training errors. In the framework of the soft margin SVC, a free parameter *C* can control the trade off between training errors and the width of the margin. i.e. we can expand the margin by allowing some samples in the other side of the margin (a.k.a. margin errors) during the training (see figure 5.2). We introduce a slack variable to formulate this new problem.

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + C\textstyle\sum_{i=1}^{N}\xi_i \\ &\text{subject to} && t_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) \geq 1 - \xi_i \\ & && \xi_i \geq 0, i = 1..., N \end{aligned} \tag{5.6}$$

The larger the *C* is, the higher the penalty of the training errors, which has the inverse effect of the regularization, λ, in ridge regression. If we set *C* to a large enough value, this will be equivalent to the hard margin SVC (5.1). The corresponding Lagrangian is defined as

$$L(\mathbf{w}, b, a, \xi, r) = \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N} a_i\{t_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) - 1 + \xi_i\} - \sum_{i=1}^{N} r_i\xi_i, 0 \leq r_i \quad 0 \leq a_i \tag{5.7}$$

where *r* and *a* are the corresponding Lagrange multipliers. Differentiating with respect

to $\mathbf{w}$, ξ, and b, and setting the derivative to zero, leads to the dual form of the soft margin SVC

$$\begin{aligned} &\text{Maximize} && -\frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{H}\mathbf{a}+\sum_{i=1}^{n}a_i \\ &\text{subject to} && \sum_{i=1}^{n}t_i a_i = 0 \\ & && C \geq a_i \geq 0, \quad i = 1,...,N \end{aligned} \tag{5.8}$$

Interestingly, this is very similar to the hard-margin SVC, except now the Lagrange multiplier *a* is capped by the regularization parameter *C*. This is also the reason that a *C* larger than the maximum Lagrange multiplier would result the identical solution to that from a hard margin SVC (5.3). In the soft margin formulation, if the data points satisfy the condition $t_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) > 1 \Rightarrow a_i = 0$, then these are the non-SVs. If the points satisfy $t_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) = 1 \Rightarrow 0 < a_i < C$, then these are the SVs on the margin. If the points satisfy $t_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) < 1 \Rightarrow a_i = C$, then these are the SVs with positive slack variable ξ (see figure 5.2). It is important to notice that a positive slack variable does not necessary mean the training point is misclassified in the training. If $0 < \xi_i < 1$, then the point is between the decision boundary and the margin, but it is still on the correct side of the decision boundary. The point is misclassified only if the slack variable is greater than 1. Therefore, a soft margin SVC may still have no classification errors in the training.

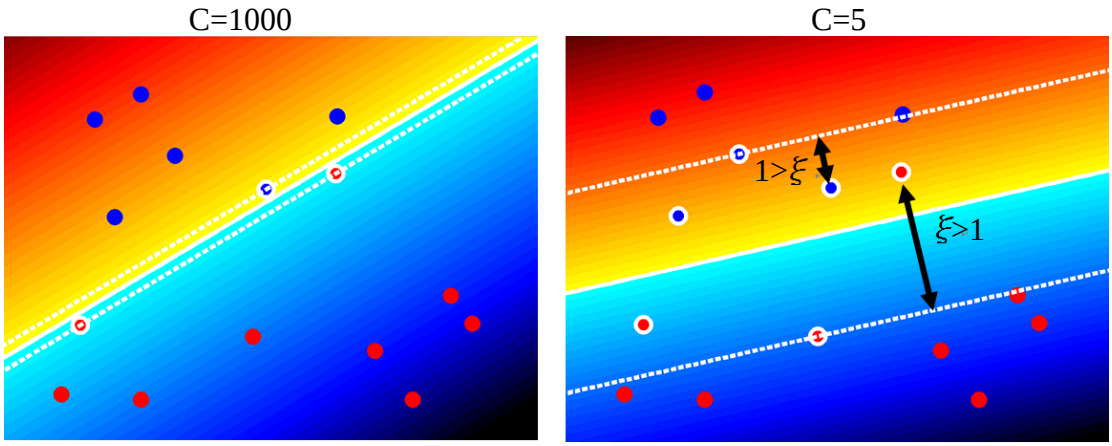


Figure 5.2 Soft-margin SVC

These figures show a 2D example of the soft margin support vector classification with different value of *C*. The blue and red spheres are the data points for different classes. The spheres enclosed by white circles are the support vectors. In this formulation, there are two types of support vectors (SVs): one is SVs on the margin and the other is the SVs on the opposite side of the margin boundary with positive slack variable ξ. If $0 < \xi_i < 1$, then this training point is still on the correct side of the decision boundary, but if $1 < \xi_i$, then this training point is on the wrong side of the decision boundary, hence results in a misclassification in the training. When the *C* is large enough (left figure), we obtain the hard margin SVC.

Parameter *C* is often chosen by minimizing the cross validation error, and it can be shown that the solutions of the optimization (5.8), i.e. $a_i, \mathbf{w}, b$ are piecewise continuous functions of *C*. In other words, identical decision boundaries may be obtained with different values of *C*. (figure 5.3)

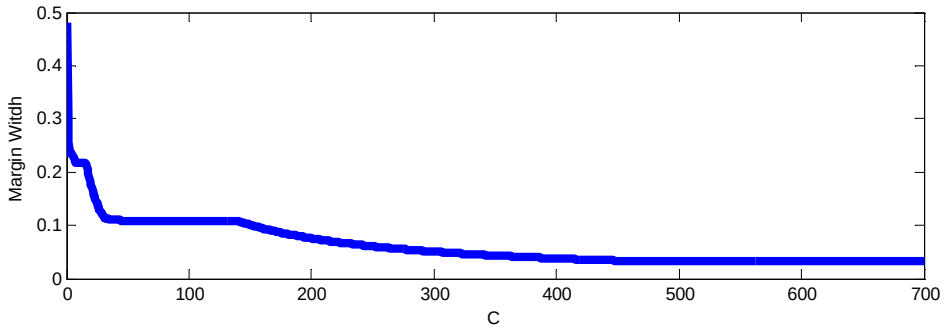


Figure 5.3 Margin width as a function of *C*

This figure shows the piecewise relationship between the margin and the parameter *C*. At some range of *C*, the margin and the hyperplane is unchanged, for instance when *C* is between 51 and 141. When *C* is greater than around 470, the solution of a hard margin SVC is obtained.

In practice, to reduce the computation in the procedure of finding an optimal *C* via cross-validation, we often find the maximum absolute value of the Lagrange multipliers in the hard margin SVC as the maximum *C*, and then set different *C* values in the validation based on the percentage of the maximum *C*. Since the parameter *C* has no intuitive meaning, an alternative formulation called $\nu - SVM$ or *nu-SVM* (Chen et al., 2005; Scholkopf et al., 2000) introduced a slight variation of problem (5.8) with the parameter $\nu \in (0,1]$. Intuitively, this parameter is realised as the lower bound on the fraction of SVs and the upper bound on the margin errors (data points that lie on the wrong side of the margin boundary and $\xi_i > 0$). The new formulation is given by

$$\begin{aligned} \text{Maximize} \quad & -\frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{H}\mathbf{a} \\ & \sum_{i=1}^{n} t_i a_i = 0 \\ \text{subject to} \quad & 1/N \geq a_i \geq 0 \\ & \sum_{i=1}^{n} a_n \geq \nu, \qquad i = 1,\ldots,N \end{aligned} \tag{5.9}$$

More details about the derivation of SVM and related theories can be fond in Dr. Pontil's lecture slides (http://www.cs.ucl.ac.uk/staff/M.Pontil/courses/index-SL05.htm) or the following text books and papers (Cristianini and Shawe-Taylor, 2000; Hastie et al., 2003; Pontil and Verri, 1998; Schlkopf and Smola, 2001). In our work, we often used the LIBSVM toolbox (Hsu et al., 2003) with a pre-computed kernel (http://www.csie.ntu.edu.tw/~cjlin/libsvm/) to perform SVC. We also have our own implementation of SVC using the quadratic programming function (quadprog) of the Matlab optimization toolbox. Identical solutions were achieved from LIBSVM and our implementation with hard margin SVC. However, there were disagreements in the solutions of soft margin SVC between two implementations. Our implementation using the quadratic programming function in Matlab did not seem to find the optimal

solutions when *C* was small.

Generally speaking, training SVC is very efficient, but the framework of SVM can not generate probabilistic outputs. One ad hoc approach to this is to impose a sigmoid function on the output of the SVC (Platt, 1999b), but this requires training the parameter of the sigmoid function from a separate testing set. Better approaches to obtain probabilistic outputs should come from the probabilistic models, which will be introduced in the later sections.

5.1.2 Relevance Vector Classification

Essentially, relevance vector classification (RVC) is logistic regression (section 2.4.6) with an ARD prior. The likelihood function is identical to (2.62), and since $p(\boldsymbol{\beta}\,|\,\mathbf{t},\boldsymbol{\alpha}) \propto p(\mathbf{t}\,|\,\boldsymbol{\beta})\,p(\mathbf{w}\,|\,\boldsymbol{\alpha})$, for fixed values of $\boldsymbol{\alpha}$, the 'most probable' value of $\boldsymbol{\beta}$ in the posterior distribution can be obtained by maximizing

$$\ln\{p(\mathbf{t}\,|\,\boldsymbol{\beta})\,p(\mathbf{w}\,|\,\boldsymbol{\alpha})\} = \sum_{n=1}^{N}\{t_n \ln f(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\varphi}_n) + (1-t_n)\ln(1-f(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\varphi}_n)\} - \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\mathbf{A}\boldsymbol{\beta} \qquad (5.10)$$

where the function *f*() is the logistic function defined in (2.49), $\boldsymbol{\varphi}$ is simplified notation for $\phi(\mathbf{x})$, $t_n \in \{0,1\}$ and $\mathbf{A} = diag(\alpha_0, \alpha_1, ..., \alpha_N)$. We take the same iterative Newton-Raphson method utilized by logistic regression, and compute the first and second derivative (Hessian matrix) of equation (5.10) with respect to the parameter $\boldsymbol{\beta}$

$$\nabla_{\boldsymbol{\beta}} \ln p(\boldsymbol{\beta}\,|\,\mathbf{t},\boldsymbol{\alpha}) = -\boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{f}-\mathbf{t}) - \mathbf{A}\boldsymbol{\beta} \qquad (5.11)$$

$$\nabla_{\boldsymbol{\beta}}\nabla_{\boldsymbol{\beta}} \ln p(\boldsymbol{\beta}\,|\,\mathbf{t},\boldsymbol{\alpha}) = -(\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi} + \mathbf{A}) \qquad (5.12)$$

where $f_n = f(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\varphi}_n)$, $\mathbf{f} = [f_1, ..., f_N]^{\mathrm{T}}$, and $\mathbf{R}$ is a diagonal matrix with elements $f_n(1-f_n)$, which is the gradient of the logistic function. Because of the form of the posterior distribution, integrating out the weights is analytically intractable, so we have to rely on approximation methods. For the Laplace approximation framework

(Friston et al., 2007b; MacKay, 2002) , which approximates a given distribution with a Gaussian distribution, it can be shown that the negative inverse of the covariance matrix of the Gaussian approximates the Hessian matrix of the distribution at its mode. Therefore, the approximate covariance matrix of the posterior distribution is given by $\mathbf{\Sigma}=(\mathbf{\Phi}^{\mathbf{T}}\mathbf{R}\mathbf{\Phi}+\mathbf{A})^{-1}$ and the maximum posterior weight $\boldsymbol{\mu}$ is computed by iterating the update function

$$\boldsymbol{\beta}_{new}=\boldsymbol{\beta}_{old}-(\mathbf{\Phi}^{\mathbf{T}}\mathbf{R}\mathbf{\Phi}+\mathbf{A})^{-1}(\mathbf{\Phi}^{\mathbf{T}}(\mathbf{f}-\mathbf{t})+\mathbf{A}\boldsymbol{\beta}) \tag{5.13}$$

The hyperparameters, $\boldsymbol{\alpha}$, are updated in the same way as in RVR (4.12), except there is no noise term for RVC. Similar to the regression model, some hyperparameters would also grow very large, effectively removing some of the basis functions. The implementation of RVC can be seen as a combination of regularized logistic regression with a standard RVR update of the hyperparameters. The implementation is a nested loop, such that that the inner loop performs the regularized logistic regression using (5.13) with fixed hyperparameters $\boldsymbol{\alpha}$, and the outer loop updates the hyperparameters using the current estimates of the posterior weights and posterior covariance matrix, using (4.12). We can take the Laplace approximation and model the approximate marginal likelihood $p(\mathbf{t}\,|\,\boldsymbol{\alpha})=N(\mathbf{t}\,|\,0,\mathbf{C})$ with covariance matrix $\mathbf{C}=(\mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^{\mathbf{T}}+\mathbf{R}^{-1})$. For a given testing point, the probability that this point belongs to class 1 is computed by $f(\sum_{i=0}^{N}\phi_i(\mathbf{x}_*)\beta_i)=f(\boldsymbol{\varphi}_*^{\mathbf{T}}\boldsymbol{\beta})$.

The main feature of RVC is that it provides a sparse probabilistic output. In practice, we found the performance of RVC to be slightly inferior to that of SVC. In terms of the computational efficiency, because the implementation of SVC (LIBSVM) is in C/C++ and RVC is in MATLAB, training SVC is often about one thousand times faster than training RVC. Even though SVC requires additional cross validation to determine the parameter *C*, it is still faster than RVC. This is why we rarely use RVC

when we were only interested in classification accuracies rather than probabilistic outputs.

5.1.3 Gaussian Process Classification

In section 4.1.3, we have introduced Gaussian processes (GP) for regression problems. To adapt the framework to one of of binary classification, we place a GP prior over the latent function g(**x**), and then squash it by a logistic or probit function (Rasmussen and Williams, 2006). The class probability is then given by $p(y=1|\mathbf{x}) = f(g(\mathbf{x}))$, where the function *f*() can be any squashing function, but we generally use a logistic function (2.49). For GPR, because both likelihood and prior are Gaussian, there is an analytic solution for making predictions (4.22). However, the non-Gaussian likelihood used for classification makes the integration analytically intractable. Therefore, approximation methods are used. In this thesis, we only consider the simple Laplace approximation (Williams and Barber, 1998), and ignore more sophisticated and computationally expensive methods such as expectation propagation (Minka, 2001). The Laplace approximation models the posterior distribution of the latent function as a Gaussian distribution having the mean at the maximum posterior estimate and the covariance matrix given by the negative inverse Hessian at the maximum estimate.

From Bayes' rule, the posterior of the latent variables is given by $p(\mathbf{g}|\mathbf{X},\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{g})p(\mathbf{g}|\mathbf{X})}{p(\mathbf{t}|\mathbf{X})}$, where $t_n \in \{0,1\}$ is the label of the class, **X** is the input features matrix, which can be built as the covariance function of the Gaussian processes (4.20), and **g** is the vector of latent variables. The goal is to maximize the posterior distribution with respect to **g**, as $p(\mathbf{t}|\mathbf{X})$ is independent of **g**, we can only consider the un-normalised posterior. The GP prior is given in (4.18) $p(\mathbf{g}|\mathbf{X}) = N(0,\mathbf{C})$ and the likelihood function is the standard binomial likelihood

used by logistic regression (2.61) $p(\mathbf{t}\,|\,\mathbf{g}) = \prod_{n=1}^{N} f(g_n)^{t_n}(1-f(g_n))^{1-t_n} = \prod_{n=1}^{N} e^{g_n t_n} f(-g_n)$.

Therefore the un-normalised log posterior function is given by

$$\begin{aligned}\Psi(\mathbf{g}) &= \ln p(\mathbf{g}\,|\,\mathbf{X}) + \ln p(\mathbf{t}\,|\,\mathbf{g}) \\ &= -\frac{1}{2}\{\mathbf{g}^{\mathbf{T}}\mathbf{C}^{-1}\mathbf{g} + \ln|\mathbf{C}|\} + \mathbf{t}^{\mathbf{T}}\mathbf{g} - \sum_{n=1}^{N}\ln(1+e^{g_n}) + \text{constant}\end{aligned} \tag{5.14}$$

We can optimize the above equation using the Newton-Raphson method. To do so, we differentiate equation (5.14) with respect to $\mathbf{g}$

$$\nabla\Psi(\mathbf{g}) = \mathbf{t} - \mathbf{f} - \mathbf{C}^{-1}\mathbf{g} \tag{5.15}$$

$$\nabla\nabla\Psi(\mathbf{g}) = -\mathbf{R} - \mathbf{C}^{-1} \tag{5.16}$$

where $f_n = f(g_n)$, $\mathbf{f} = [f_1, ..., f_N]^{\mathbf{T}}$, and $\mathbf{R}$ is a diagonal matrix with elements $f_n(1-f_n)$. The update function is then given by

$$\begin{aligned}\mathbf{g}_{new} &= \mathbf{g}_{old} + (\mathbf{R}+\mathbf{C}^{-1})^{-1}(\mathbf{t} - \mathbf{f} - \mathbf{C}^{-1}\mathbf{g}_{old}) \\ &= (\mathbf{R}+\mathbf{C}^{-1})^{-1}(\mathbf{R}\mathbf{g}_{old} + \mathbf{t} - \mathbf{f})\end{aligned} \tag{5.17}$$

Once we have found the mode $\mathbf{g}_{MAP}$ of the posterior distribution, we can compute the Hessian matrix, hence the Gaussian approximation of the posterior distribution is given by $p(\mathbf{g}\,|\,\mathbf{X},\mathbf{t}) = N(\mathbf{g}_{MAP}, (\mathbf{R}+\mathbf{C}^{-1})^{-1})$. To make the predictions, we first compute the latent variable of the new input by adopting the equation in GPR (4.22), $\mathrm{E}[g_* \,|\, \mathbf{X}, \mathbf{t}, \mathbf{x}_*] = \mathbf{k}^{\mathbf{T}}\mathbf{C}^{-1}\mathbf{g}_{MAP} = \mathbf{k}^{\mathbf{T}}(\mathbf{t}-\mathbf{f})$ (notice at $\mathbf{g}_{MAP}$, $\nabla\Psi(\mathbf{g}) = 0$, so $\mathbf{g}_{MAP} = \mathbf{C}(\mathbf{t}-\mathbf{f})$). The vector $\mathbf{k}$ has elements $\mathbf{k}_i = K(\mathbf{x}_i, \mathbf{x}_*), i = 1, ..., N$. If we use a linear kernel and are interested in generating the "weight map" for visualising the contribution of each input feature, we can computed the map by $\bar{\mathbf{w}} = \sum_{n=1}^{N}(t_n - f_n)\mathbf{x}_n$. The variance of the predicted latent variable is given by $\mathrm{var}[g_* \,|\, \mathbf{X}, \mathbf{t}, \mathbf{x}_*] = K(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}^{\mathbf{T}}(\mathbf{R}^{-1}+\mathbf{C})^{-1}\mathbf{k}$. Given the mean and variance of the predicted latent variable $\mathbf{g}_*$, we can compute the averaged prediction by $\bar{\pi}_* = \int f(g_*)\, p(g_* \,|\, \mathbf{X}, \mathbf{t}, \mathbf{x}_*) dg_*$. Because of the asymmetric

nature of the logistic function, the averaged prediction is always less than the *MAP* prediction $f(g_*)$, however, for binary classification, the predicted test labels given by selecting the class of highest probability obtained by averaged or *MAP* predictions are identical. i.e. if $0.5 < f(g_*)$, then $0.5 < \bar{\pi}_* < f(g_*)$, or if $f(g_*) < 0.5$, then $f(g_*) < \bar{\pi}_* < 0.5$.

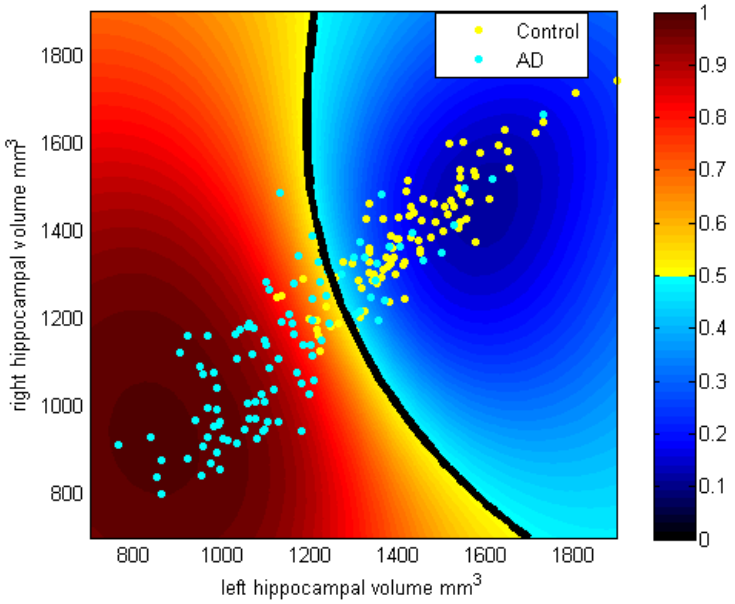


Figure 5.4 Gaussian Processes Classification with RBF kernel

This figure shows the decision boundary found from the AD data by GPC with an RBF kernel. The hyperparameters were learnt through optimising the marginal likelihood (5.18). The distribution in the figure shows the probability of the data point belongs to the AD group, $p(t = \mathrm{AD}|\mathbf{x})$, and the dark line indicates the decision boundary where the probability is 0.5.

The marginal likelihood function of GPC $p(\mathbf{t}|\mathbf{X},\theta) = \int p(\mathbf{t}|\mathbf{g})\, p(\mathbf{g}|\mathbf{X},\theta) d\mathbf{g}$ is analytically intractable, so we use the Laplace approximation of the posterior distribution, and an approximation of the log marginal likelihood

$$\begin{aligned} \ln p(\mathbf{t}|\mathbf{X},\theta) &= \Psi(\mathbf{g}_{MAP}) - \frac{1}{2}\ln|\mathbf{R}+\mathbf{C}^{-1}| + \frac{N}{2}\ln(2\pi) \\ &= -\frac{1}{2}\{\mathbf{g}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{g} + \ln|\mathbf{C}| + \ln|\mathbf{R}+\mathbf{C}^{-1}|\} + \mathbf{t}^{\mathrm{T}}\mathbf{g} - \sum_{n=1}^{N}\ln(1+e^{g_n}) + \text{constant} \end{aligned} \tag{5.18}$$

For the implementation, we used the GPML Matlab toolbox (http://www.gaussianprocess.org/gpml/code/matlab/doc/) with some modifications, so

we can use a pre-computed kernel as input. We also add some code to generate linear combinations of kernels. There is another in house implementation by Dr. Ashburner. In this implementation, Powell's line search method (Press et al., 1992) is used to optimize the marginal likelihood (5.18).

5.1.4 Multi-class Classification approaches

Generally speaking SVC, RVC, and GPC are all binary classifiers, although we can expand the likelihood function of those probabilistic models, such as RVC and GPC, from Bernoulli distribution to standard multinomial distribution, and then apply the softmax function

$$p(t = C_k \mid \mathbf{w}, \mathbf{x}) = \frac{\exp(\mathbf{w}_k^{\mathrm{T}}\mathbf{x})}{\sum_{i=1}^{K} \exp(\mathbf{w}_i^{\mathrm{T}}\mathbf{x})} \tag{5.19}$$

Often people take simpler approaches by combining a number of binary classifiers (Hassabis et al., 2009; Mourao-Miranda et al., 2006). For situations with K classes, there are two commonly used approaches; one is the "one versus the rest" classifier. This works by training K classifiers, each of them trained with one class versus the other K-1 classes. The classification for a testing point is determined by the classifier that achieves the highest classification scores i.e. furthest away from the decision boundary toward the particular class. Nevertheless, ambiguous cases may still occur when all the classifiers consider a testing point to be in "the other K-1 classes". The label of this point will be undetermined. Another approach is called the "one versus one" classifier, which works by introducing $K(K\text{-}1)/2$ or C_2^K classifiers. Each of the classifiers is trained with one class versus another class only. The assigning of a testing point is achieved by majority vote, in other words, the most frequent class to which the testing point is classified by all the classifiers. Ambiguous cases may also occur in this approach. For example if we have three classes 1, 2, and 3, then we will

have three classifiers (1vs2, 1vs3, 2vs3). The testing point may be classified into class 1, class 3, and class 2 from the three classifiers respectively.

To tackle the issue of ambiguous cases, we introduced a multi-classification method for fMRI block designed experiments. This method also utilizes the temporal information, without compressing into a reduced kernel by equation (3.18). Our approach breaks the classification into three stages: 1. Train *K* regression models; 2. Predict the temporal profiles for a testing block; 3. Match the predicted *K* profiles with the canonical profile (figure 5.5). This approach was originally inspired by the PBAIC, so we take a similar approach in the training phase, that is, we only change the target variable, but use the same input features. For example, consider an experiment with three conditions in the design. We could train three different regression machines with RVR or KRR, where each of the machines takes the same kernel generated from the fMRI volumes as input features, but the target variables are the corresponding regressors (HRF convolved) in the design matrix. In the predicting phase, temporal profiles of the test block (multiple fMRI volumes) are predicted from all three regression machines. To assign the class membership, we compare all the predicted profiles with the canonical profile which is the HRF convolved block. Covariance or correlation is chosen as the metric to measure similarities. Both measures ignore the constant offset, and covariance considers the magnitude of the prediction, but correlation ignores the information of magnitude. The class is assigned to the condition for which the machine achieves the highest similarity between the predicted profile and the canonical profile.

This method resolves the issue of ambiguous regions and showed higher accuracies for prediction than combinations of binary SVC at the individual level. Although we gave an example of block design experiment, there is no reason why this method should not be used for event related designs. A practical example will be

given in an application section of this chapter.

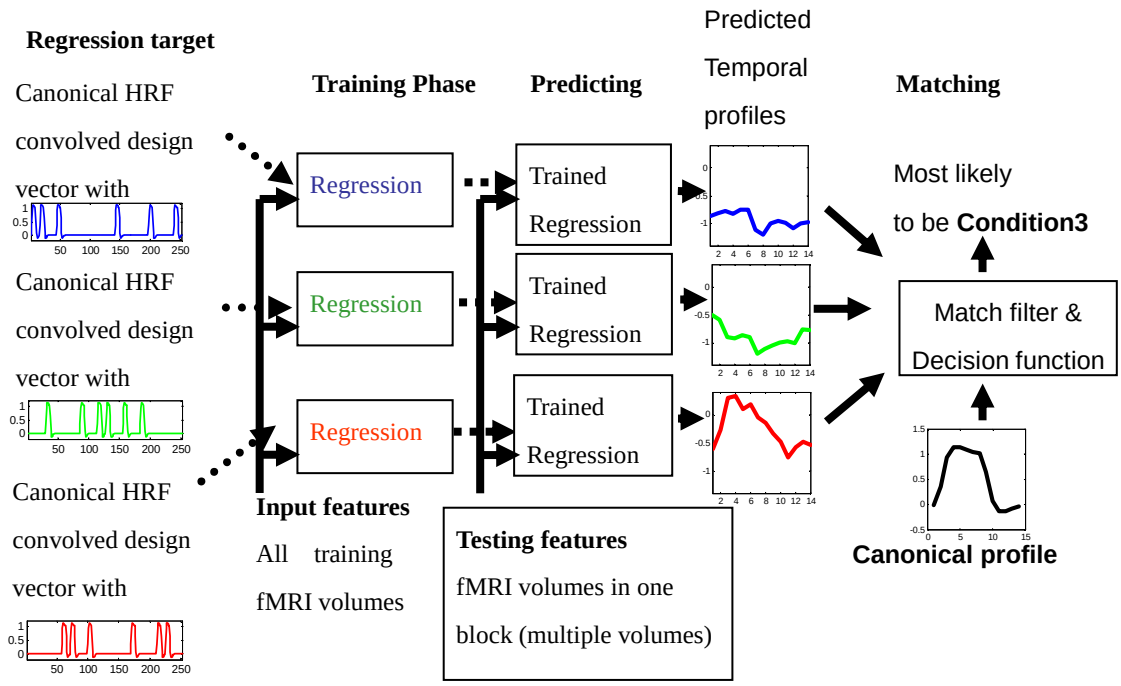


Figure 5.5 Multi-classification using regression machines

This figure shows the pipeline of the multi-classification method we proposed by predicting different conditions in a block design fMRI experiment

5.1.5 One-class Classification

One class classification is often referred to as novelty detection (Scholkopf et al., 2001; Shawe-Taylor and Cristianini, 2004; Tax, 2001). The one class classifier can learn the distribution of the 'normal' samples from the training data, and then perform the detection of abnormal or novel sample in a new set (Sato et al., 2008a). Often, when we have uneven sample sizes, for instance, a larger sample of control subjects than patients, it will be preferred to train a one-class classifier that can well describe the distribution of the controls. This classifier can still be used to classify patients by finding subjects who do not belong to the class of control subjects. We introduced a simple method in section 3.3.3, by calculating the distance from a data point to the centre of the mass of the set from the kernel (3.23). We can then define the radius, hence the boundary of the hypersphere enclosing the data set. There is a more flexible

approach, namely “the smallest hypersphere containing the training set” (Schlkopf and Smola, 2001; Shawe-Taylor and Cristianini, 2004). This method also works on kernel space, and therefore does not require explicit specification of the feature space. This allows us to explore the hypersphere in the feature mapping space. Unfortunately, there is no easy way to find the centre that minimises the radius of the hypersphere, so the solution can only be found using an optimization scheme. This leads to the following formulation

$$\begin{aligned} &\text{Minimize} && r^2 \\ &\text{subject to} && \|\phi(\mathbf{x}_i)-\mathbf{c}\|^2 \le r^2 \\ & && i=1,...,N \end{aligned} \tag{5.20}$$

Similar to SVM, we can use Lagrange multipliers to solve this optimization problem. The dual formulation is given by

$$\begin{aligned} &\text{Maximize} && W(\mathbf{a}) = -\sum_{j,i=1}^{N} a_i a_j K(\mathbf{x}_j,\mathbf{x}_i) + \sum_{i=1}^{N} a_i K(\mathbf{x}_i,\mathbf{x}_i) \\ &\text{subject to} && \sum_{i=1}^{N} a_i = 1 \\ & && a_i \ge 0, \quad i=1,...,N \end{aligned} \tag{5.21}$$

This is a standard quadratic programming optimization as used by SVM, and similarly, some of the Lagrange multipliers will have zero values. Only those data points laid on the circumference have positive Lagrange multipliers. They are analogous to the support vectors in SVM. The centre of the sphere can be calculated by $c = \sum_{i=1}^{N} a_i \phi(\mathbf{x}_i)$, and the radius of the sphere is $r = \sqrt{W(\mathbf{a}^*)}$. The function $f(\mathbf{x}_*) = K(\mathbf{x}_*,\mathbf{x}_*) - 2\sum_{i=1}^{N} a_i K(\mathbf{x}_i,\mathbf{x}_*) + \sum_{j,i=1}^{N} a_i a_j K(\mathbf{x}_j,\mathbf{x}_i) - r^2 \ge 0$ will assess whether the test point is inside the hypersphere. Positive output indicates that the test data is outside the hypersphere, hence is novel.

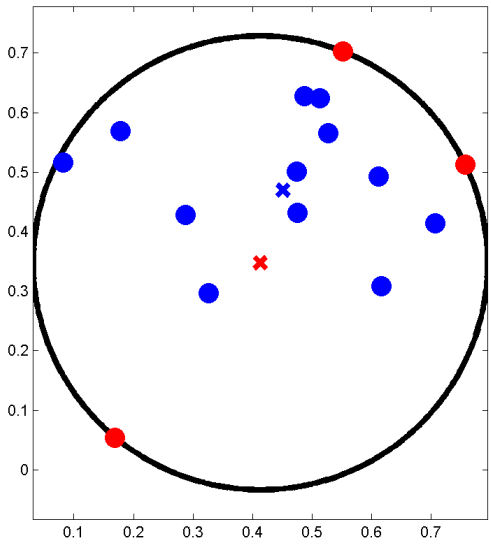


Figure 5.6 The smallest enclosing 2D circle

This figure shows 2D example of the smallest sphere that encloses all the data points. The red points are the points with positive Lagrange multipliers (support vectors). The distance of these points to the centre of the sphere is equal to the radius of the sphere, and the red cross indicates the centre of the sphere. The blue cross indicates the centre of mass. If the distribution of the points is asymmetric, the centre of mass and the centre of the smallest enclosing circle may be quite distant.

Furthermore, if data points in the mapped feature space have equal distance to the origin, in other words, the points are located on the boundary of the hypersphere centred at the origin of the feature space such as the RBF kernel and the normalized kernel. Since the diagonal elements of those kernels are all 1. We can simplify the objective function by maximizing only $-\sum_{j,i=1}^{N} a_i a_j K(\mathbf{x}_j, \mathbf{x}_i)$. Interestingly, this is equivalent to a binary SVC between all training points and the origin. Notice the LIBSVM implements this objective function rather than the objective function described in (5.21). Therefore, one class SVM in LIBSVM works only for the RBF kernel or the normalized kernel.

We can also take the approach in soft margin SVC by relaxing the boundary of the hypersphere to avoid over-fitting the data. This can achieve a “soft hypersphere"

that contains most of the data points. As in the case of the soft margin SVC, a free parameter *C* is introduced to penalize the training errors (points outside the sphere). The new objection function is given by

$$\begin{aligned} &\text{Minimize} && r^2 + C \| \xi \| \\ &\text{subject to} && \| \phi(\mathbf{x}_i) - \mathbf{c} \|^2 \le r^2 + \xi_i \\ & && \xi_i \ge 0, \quad i = 1, ..., N \end{aligned} \tag{5.22}$$

The dual form of this new objective function is give by

$$\begin{aligned} &\text{Maximize} && W(\mathbf{a}) = -\sum_{j,i=1}^{N} a_i a_j K(\mathbf{x}_j, \mathbf{x}_i) + \sum_{i=1}^{N} a_i K(\mathbf{x}_i, \mathbf{x}_i) \\ &\text{subject to} && \sum_{i=1}^{N} a_i = 1 \\ & && C \ge a_i \ge 0, \quad i = 1, ..., N \end{aligned} \tag{5.23}$$

The effect of *C*, which capped the Lagrange multipliers, is exactly the same as in soft margin SVC. The centre of the hypersphere is still given by $c = \sum_{i=1}^{N} a_i \phi(\mathbf{x}_i)$, although its computation is intractable from an RBF kernel. As with the soft margin SVC, points with the corresponding Lagrange multipliers equal to *C* would have positive slack variables, so these training points are outside the hypersphere. Points on the boundary of the hypersphere would have their Lagrange multipliers in the range $0 < a_i < C$. The radius of the sphere is $r = \sqrt{K(\mathbf{x}_l, \mathbf{x}_l) + \sum_{j,i=1}^{N} a_i a_j K(\mathbf{x}_j, \mathbf{x}_i) - 2\sum_{j=1\$}^{N} a_i K(\mathbf{x}_j, \mathbf{x}_j)}$, $0 < a_l < C$. To evaluate the testing data, the decision function is the same as for the standard hypersphere formulation without *C*. We can also take the approach of $\nu - SVM$, by converting the parameter *C* into a more intuitive variable ν, such that $C = 1/(\nu N)$. The ν parameter allows us to exert some control over the fraction of points that are excluded from the hypersphere.

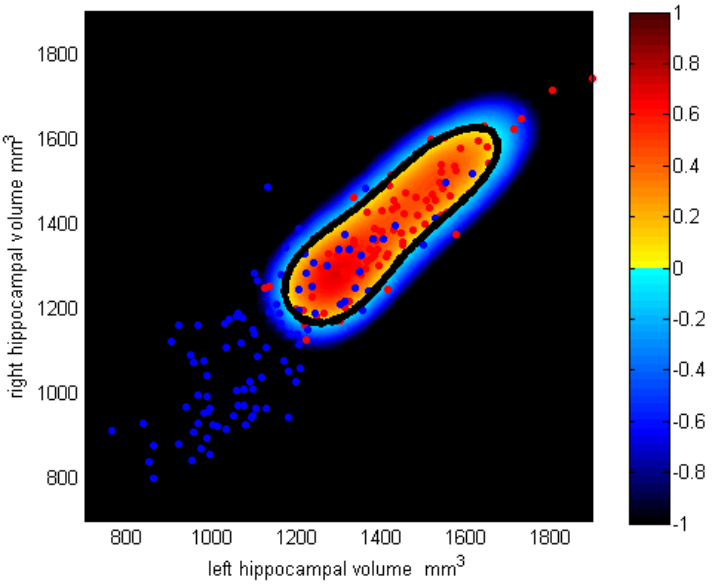


Figure 5.7 One Class SVM with RBF kernel

This figure shows a 2D example of the one class classifier applyied to the AD dataset using a RBF kernel ($\gamma = 5e^{-5}$). The one class classifier was trained on the normal subjects (red dots) with 20% training misclassification ($\nu = 0.2$). The dark contour indicates the boundary of the classifier. 76.77% of AD patients (blue dots) were classified as "novel" from the training class.

5.2 Application: Classification of MR Scans in Alzheimer's Disease

Recent advances in MRI segmentation (Ashburner and Friston, 2005), spatial normalization (Ashburner, 2007), and machine learning techniques (Bishop, 2006b) have led to the application of automatic classification to MRI for detection of a variety of disease states (Davatzikos et al., 2008; Fan et al., 2008a; Fan et al., 2005; Fan et al., 2007b; Golland et al., 2002). The objectives of our study were to assess how well SVC classified individuals, and to determine whether datasets from multiple scanners and different centres could be combined to obtain improved classification accuracy (Kloppel et al., 2008d).

5.2.1 Introduction

Alzheimer's disease (AD) is a neurodegenerative disorder, and a common cause

of dementia. Research advances have shown the biochemistry link to the molecular pathogenesis of plaques composed of amyloid beta, and tangles composed of hyperphosphorylated tau. Familial AD is very rare, but sporadic AD is common and has affected more than 15 million people worldwide. The exact cause of sporadic AD is unknown, and its early detection is important (Blennow et al., 2006). In practice, the diagnosis is mainly based on clinical history and neuropsychological examination, but the diagnosis rate of AD is less than one-half in the primary care setting (Solomon and Murphy, 2005). When more detailed criteria are used, the diagnostic accuracy is improved, but still has around 80% sensitivity (Petersen et al., 2001). Only recently, people have realized that MRI can improve the diagnostic accuracy of AD. Studies have shown that the use of MRI to measure temporal lobe atrophy can assist diagnostic accuracy (Barnes et al., 2004), evidence shows that hippocampal volume is a sensitive marker for pathological AD stage (Jack et al., 2002). However, a lot of studies still rely on manual tracing of hippocampi, which is laborious and time consuming. Besides, single measurements of hippocampal volume are unlikely to be more sensitive than multivariate measures. Averages of multiple voxels into a single volume measure may be easy for human interpretation, but this simplification results in some information loss. On the other hand, statistical learning methods are well suited to finding patterns in high dimensional space, especially as the computational cost of kernel methods are bounded by the number of training samples rather than the number of input features. We applied SVC in this work to examine different sets of MRI from AD patients and elderly control subjects. One advantage of our approach is that all the procedures are fully automatic; therefore the result is not biased by subjective errors from manual tracing.

5.2.2 Materials and methods

We have three sets of data from different sources. In the first set (group 1), AD

patients were largely from a community based setting in Rochester, Minnesota, USA. All AD diagnoses were confirmed with neuropathology. There were 20 patients and 20 controls. Controls all had MMSE greater or equal to 27.

The second set consisted of neuropathologically confirmed AD patients and controls from the Dementia Research Centre, University College London. There were 14 patients and 14 controls. The AD patients in this group (group 2) tended to be younger than AD patients in group 1. Cognitively normal controls were confirmed by clinical exam or pathology.

The third set (group 3) consisted of 99 clinically confirmed AD patients and 90 age and sex matched control subjects. Subjects were from a community and referral based sample in Rochester, Minnesota, USA. Since the patients were only clinically confirmed (no post mortem examination), some of the patients may not actually have had AD. From previous studies, we speculated that only about 85% of the patients in the third set actually had AD. In fact, group 3 is a subset of the subjects mentioned in section 4.4 of chapter 4.

For groups 1 and 3, MR scans were collected over a period of about 10 years, from 13 different scanners. However, all scanners were the same platform, General Electric Signal 1.5T scanners. The scanning protocols used were also similar, and the parameters for the T1-weighted images were: TR=23 to 27 ms, TE=6 to 10ms, flip angle 25 degrees or 45 degrees, voxel size 0.86mm x 0.86mm x 1.6mm or 0.94mm x 0.94mm x1.6mm.

For group 2, the scans were acquired from three different 1.5T scanners. Image parameters were TR=35 or 15, TR=5 or 5.4 or 7, flip angle 35 degrees or 15 degrees.

The image pre-processing procedures are described in section 3.2.1. Images were firstly segmented by SPM into GM and WM, and then imported into a rigidly aligned space. The GM and WM were iteratively registered to the population mean by the

DARTEL toolbox. Finally, the linear kernel is computed from the normalized and Jacobian scaled GM. For classification, we applied standard hard-margin SVC in this work. Methodological details can be found in 5.1.1. Leave one out cross validation (section 2.5.1) was applied to test the generalization performance.

5.2.3 Results and discussion

Classification between confirmed AD patients and controls in group 1 yielded the accuracy of 95% with whole brain GM from the leave one out cross validation. The corresponding sensitivity was 95% (i.e. probability of correctly identifying AD patients) and specificity was 95% (i.e. probability of correctly identifying control subjects). One 89 year old AD patient with a MMSE of 29 and one 86 years old control were misclassified. Classification of group 2 achieved 92.9% classification accuracy with the same procedures used for group 1. The sensitivity was 100% and specificity was 85.7%. The two oldest controls were misclassified. We then combined both groups 1 and 2, and obtained a cross validation accuracy of 95.6%, which was higher than any of the groups alone. The sensitivity was 97.1% and the specificity was 94.1%. Finally, we trained the SVC with data in group 1, and then used group 2 as the test set. The accuracy was 96.4% (sensitivity 100%, specificity 92.9%). Conversely, we trained group 2 and tested on group 1. The accuracy was 87.5% (sensitivity 95%, specificity 80%). Because subjects in group 1 were generally older than subjects in group 2, we suspected that the poor specificity was due to misclassification of older subjects in group 2. This actually raises an important issue of supervised learning methods, which is that the statistical machine is constrained by the information available in the training set. When the training samples are relatively scarce, as in our case, differences of the distribution between the test set and training set can impair the accuracy of classification. In our case, because aging patterns have some similarities between atrophy patterns in AD, the classifier may not learn sufficient information to

discriminate between aging patterns and AD patterns from a younger group. However, when we trained from a set of relatively older subjects, the classifier would be able to characterise the pattern between AD and aging much more clearly.

Group 3 contains probable AD and mild AD patients. The classification accuracy using whole brain GM was 83.2% (sensitivity 81.8%, specificity 85.6). A further improvement to 88.9% (sensitivity 85.9%, specificity 93.3%) was obtained when volumes of interest (VOI) centred around both left and right parahippocampi were used. This VOI is the same one described in 4.4.2. Because many AD patients in both groups 1 and 2 were in the later stage of AD, we would like to test whether training using data from group 1 plus group 2 can predict patients in group 3, which consisted of many mild AD patients. The result was surprisingly biased, with an accuracy of 80% (sensitivity 63.6% specificity 97.8%). It seemed that training SVC with severe cases of AD drove the hyperplane toward the direction of the AD group. One way to correct this situation is by biasing the decision boundary closer towards the control subjects. By sacrificing the specificity, we can improve the sensitivity. We found that by biasing by -0.5 in the direction perpendicular to the decision boundary, the accuracy improved to 89% and the sensitivity increased to 87.9% with specificity dropping to 91.2%.

Conversely, when we trained using group 3 data and predicted groups 1 and 2, an accuracy of 94.1% (sensitivity 94.1%, specificity 94.1%) was achieved. This implied that patterns of mild AD are consistent with patterns of severe AD. Because patients with mild AD should be more similar to normal controls, training with difficult cases should yield good results when predicting less difficult cases.

Finally, we combined all three groups together, and the accuracy was 87.2% (sensitivity 85.7%, specificity 88.8%). The following table summaries our results.

Group	% correctly classified	Sensitivity (%)	Specificity (%)
Group 1	95.0	95.0	95.0
Group 2	92.9	100	85.7
Group 3	83.2	81.8	85.6
Dataset 1 for training, set 2 for testing	96.4	100	92.9
Dataset 2 for training, set 1 for testing	87.5	95.0	80.0
Group 1 + 2	95.6	97.1	94.1
Group 3 with VOI	88.9	85.9	93.3
Dataset 1 +2 for training, set 3 for testing	80	63.6	97.8
Dataset 1 +2 for training, set 3 for testing (after correcting the bias)	89	87.9	91.2
Dataset 3 for training, set 1+2 for testing (after correcting the bias)	94.1	94.1	94.1
Group 1 + 2+ 3	87.2	85.7	88.8

Because we used linear kernels in this application, it is feasible to reproduce the linear weights in the input feature space, or the “weight map”. The weight map allows visualisation of those regions that contribute more to the discrimination between AD and controls. In order to produce a less biased weight map, we utilise the “bootstrap methods” (Efron, 1979; Efron and Tibshirani, 1993; Zoubir and Boashash, 1998). We resampled the whole dataset 200 times. Each time, around 70% of the samples were selected. The SVC was trained 200 times, and the 200 corresponding weight maps were averaged to produce the mean map. Because the weight maps vary across different training subsets, to produce a map that illustrates most consistent voxels discriminating AD from controls, we divide the mean weight map by its voxel-wise standard deviation. This then constructed the z-score map. The assumption was that if there is no information in the training images to distinguish between AD and controls, the mean weight map should be 0 across all voxels. From the aspect of visualization, the z-score map would suppress voxels having high variance of weights from different

weight map. In other words, the z-score map can indicate regions which are consistently “informative” across subjects.

From the weight map, voxels around the parahippocampal gyrus and parietal cortex showed strong contribution to classify between AD and controls. Because we set the label of AD patient as 1, and controls as -1, negative values in the weight map indicates relatively higher grey matter volume increasing the likelihood of classifying into normal. In other words, degeneration in the parahippocampal gyrus and parietal cortex would lead to be classified as AD patients.

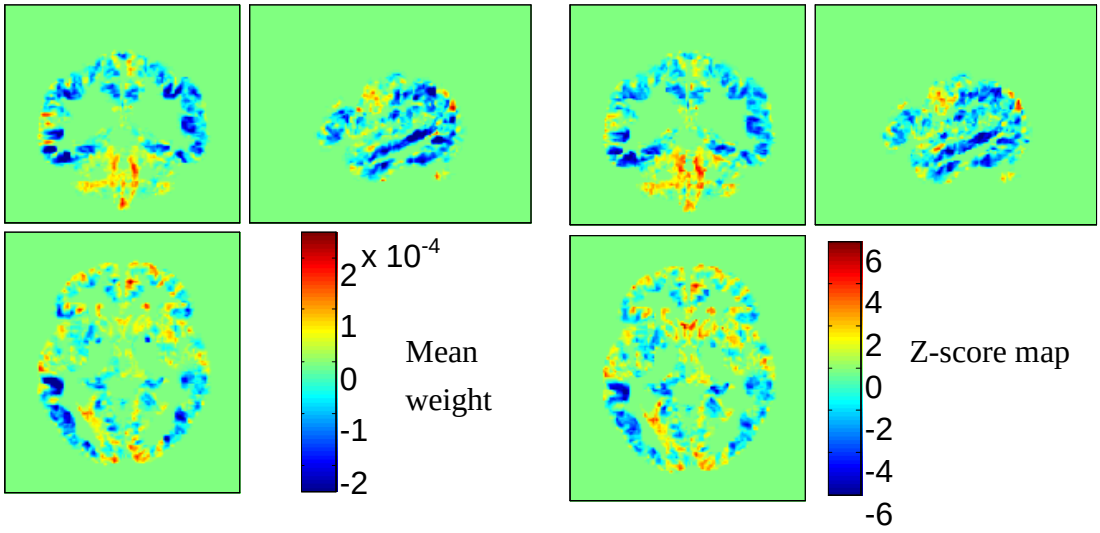


Figure 5.8 Weight maps for AD classification

These two figures show the relevance of voxels for classifying patients from both groups 1 and 2 (pathologically confirmed subjects). The left figure is the mean weight map generated by averaging 200 SVC solutions with bootstrap sampling. The right figure shows the corresponding z-sores map. Both maps are visually very similar. The blue areas indicate where relatively higher grey matter volume increases the likelihood of classifying as normal. The red areas indicate the opposite effect.

Our results clearly indicate the feasibility of apply machine learning techniques to aid the clinical diagnosis of AD. The procedure presented here promises to classify disease specific atrophy from that of normal aging in a standard T1 weighted structural MRI scan. Generally speaking, our results have been comparable with or

better than other classification methods published based on MR images (Barnes et al., 2004; Csernansky et al., 2004; Jack et al., 2002; Wahlund et al., 2005). Most of these studies restricted analysis to temporal lobe structures. Although, we also used temporal VOIs for group 3 to improve the classification performance, this improvement could be because the clinical determination of AD was partially based on examining temporal lobe structures. To be fair, it is actually difficult to judge whether our approach was superior to other methods, as every group worked on different datasets. However, we can still compare our automatic methods with human experts.

5.2.4 Direct Comparison between radiologists and our computerised method

The aim here is to verify the performance of the automatic classification system, including the automatic segmentation and spatial normalization processing. One way is to compare the prediction accuracy with those made by clinical radiologists. This project was mainly led by Dr. Stefan Kloppel, and he kindly allowed me to put the material of this study, for which I was one of the co-authors, in my thesis (Kloppel et al., 2008b) to make the chapter more complete. The binary diagnosis was made by six radiologists, with different levels of experience, using scans from groups 1 and 2 (pathologically confirmed cases).

To allow a fair comparison, radiologists were provided information about the age range of patients and controls, and were also told that the both diagnostic categories were age matched and equal in number. This means the radiologists were not told the age for each scan, but the age range in the group. The radiologists were asked to perform binary classification with an additional level of diagnostic confidence (low, intermediate or high). The radiologists rated group 1 first, and just before rating group 2, we disclosed the diagnosis of a third of patients and controls to the radiologists

from group 2 to mimic the training in SVC. Disclosed cases were randomly selected and removed from the test set. There was no time limit for the radiologists to perform their diagnoses.

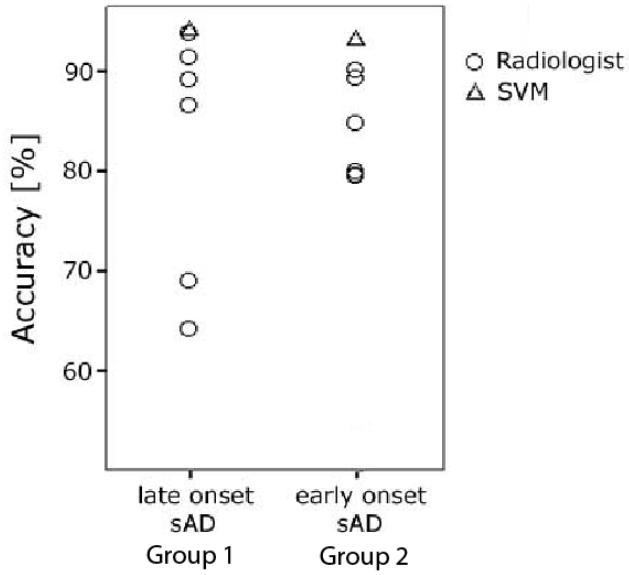


Figure 5.9 Classification performance of radiologists and SVM

This figure shows the classification accuracy of radiologists and SVM for both groups 1 and 2. The results of radiologists are shown as circles, and those of SVM are shown as triangles.

One radiologist performed as accurately as SVC in the task of classifying AD and controls from group 1, but SVC outperformed the other five radiologists. Radiologists' diagnostic accuracy was highest when they expressed high diagnostic confidence. Correlations between the diagnostic accuracies of the radiologists with the percentage of brain scans in their daily workload showed that their diagnostic accuracy improved with their level of experience. All radiologists achieved relatively high accuracies on the third dataset (figure 5.9). This suggests that training by disclosing part of the data may have helped the less experienced radiologists, as they improved the most.

Given the good diagnostic accuracy achieved by SVC relative to the radiologists, it substantially extended the possibility of the use of computers in clinical decision making (Ashburner et al., 2003). Although, experienced radiologists working under optimal conditions are very accurate, the automatic system could improve diagnoses in places where trained neuroradiologists or cognitive neurologists are scarce. It is also important to realize that the performance achieved in our classification system was hugely attributed to the image pre-processing algorithms. The pre-processing can be understood as the procedure for feature extraction. If the extracted features were meaningful, the task of classification would be relatively invariant to the statistical learning tools. Nevertheless, the automatic classification system introduced here warrants similar applications to large image sets, such as those being collected for the Alzheimer's Disease Neuroimaging Initiative (ADNI) (Mueller et al., 2005). When a large training set is available, multiple sub-training groups can be established based on the gender, age, and ethnicity. This should reduce the inter-subject variability due to other factors. Once the pattern of AD has been characterised, screening the patients could just take a few minutes (including segmentation and registration), as there is no need to re-train the classifier. Considering the growing possibilities of cloud computing (Buyya et al., 2008) and high speed internet, computation does not necessary need to be done on site, but could be dispatched to computing centres around the world.

5.3 Application: Automatic Detection of Presymptomatic Huntington Disease Using Structural MRI

The benefit of treatments for neurodegenerative diseases is much higher in the very early and preclinical stages of degeneration. In this project, we applied the fully

automatic classification method, which is nearly the same one described in section 5.2, for detecting subtle degenerative change in preclinical Huntington Disease (HD).

5.3.1 Introduction

The availability of a definitive genetic test for HD provides a perfect standard for evaluating the performance of classification systems from gene mutation carriers who do not have symptoms. Group studies of familial HD have shown substantial neurodegeneration before the onset of clinical symptoms (Thieben et al., 2002). Preclinical degeneration observed from structural MRI scans may indicate important timing information for treatment to slow down the process of degeneration. Automatic and efficient methods would be required for screening large numbers of subjects for early detection. In addition, Presymptomatic HD is an important model for the study of the earliest stages of neurodegeneration and atrophy. This autosomal dominant disorder has complete penetrance, and results from expanded CAG trinucleotide repeats in the Huntington gene, which can be detected from the blood (Penney Jr et al., 1997).

5.3.2 Materials and Methods

A total of 96 presymptomatic Huntington disease gene mutation carrier (PSC) and 95 control subjects enrolled in the PREDICT-HD study (Paulsen et al., 2006) were included. PREDICT-HD is an international multicentre study that focuses on discovering biological and refined clinical predictors of disease progression in PSCs. The PSCs in the dataset have at least 39 CAG repeats in the HD gene, whereas controls have fewer than 30 CAG repeats. Subjects were also screened for unstable illness, alcohol or drug abuse, a history of special educational needs, and a history of other CNS diseases. The scans acquired were also checked for artifacts with a semi-automatic quality control procedure at the time of acquisition.

In this study, we separate PSCs into various groups by their estimated time to

clinical manifestation, based on age and CAG repeat length (tables available at http://www.cmmt.ubc.ca/sites/default/files/pdf_hayden_supplementary_tables.pdf) (Langbehn et al., 2004). This robust model was based on 3,000 gene carriers. We used the algorithm to estimate the probability of developing clear signs of HD in the next 5 years, and then divided PSCs into three equal sized groups according to their probability of clinical manifestation within 5 years:

1. less than 10% (far group)
2. 10% to 33% (mid group)
3. more than 33% (near group)

Controls were selected to match the age in each group. A control subject may also be repeatedly used in different groups. Each group contains 32 HD and 32 controls.

The T1-weighted MRI scans were acquired using the three dimensional volumetric spoiled gradient echo series on 1.5T scanners (TE=3ms, TR=18ms, flip angle 20 degrees, field of view 240mm, 124 slices at 1.5mm thickness, matrix size 256x192). We applied standard pre-processing procedures. Briefly speaking, images were segmented into GM and WM, and then imported into a rigidly aligned space. The GM and WM were iteratively registered to the population mean by the DARTEL toolbox. Finally, the linear kernel is computed from the spatially normalized and Jacobian scaled GM (for details, see section 3.2.1).

Unlike the AD dataset, where the controls and the AD patients were quite distinct, there was a lot of overlap between the PSCs and controls, so we applied soft-margin SVC with parameter C in this project. When we evaluated the generalization performance, in order to prevent overly optimistic estimation, we performed a three way split cross validation (section 2.5.1). This means we split the data into training, validating, and testing sets. We used the validating set to optimise C, and use the test

set to verify the classification accuracy with the optimal *C* value.

From previous voxel based morphometry (VBM) studies, we knew that the atrophy in HD is localised in the striatum (Kassubek et al., 2004; Thieben et al., 2002). This prior knowledge should increase the predictive power of the classifier, so we performed additional feature selections procedures. We performed a VBM comparison between normal controls and PSCs from an external dataset, to generate the statistical map. This new dataset consisted of 42 PSCs and control subjects. The scans were acquired from a 1.5T Siemens Sonata scanner (T1-wegithed MDEFT sequence, TR=20.66ms, TE=8.42ms, inversion time=640ms, flip angle 25 degrees, 176 slices at 1mm thickness, sagittal, phase encoding in anterior/posterior, field of view 224x256 mm^2). All images were spatially normalized into the same space as the original dataset. We applied three different levels of threshold on the t-map to obtain the mask for feature selection. The thresholds are as followed 1) uncorrected $p<0.001$, 2) FEW corrected $p<0.05$, 3) single voxel of the global maxima.

5.3.3 Results and Discussion

Classification accuracy depended greatly on estimated time to disease manifestation. Subjects with at least 33% chance of developing unequivocal signs of HD in 5 years were correctly classified at 68.7% accuracy (sensitivity 62.5%, specificity 75%). The prediction accuracy improved to 82.8% (sensitivity 78.1%, specificity 87.5%) when a mask of uncorrected $p<0.01$was used for selecting features. For the mid group, the best classification accuracy of 79.7% (sensitivity 78.13, specificity 81.3) was achieved when the single voxel of the global maxima was used. The far group obtained the best prediction at 60% (sensitivity 62.5%, specificity 59.38) when a mask of uncorrected $p<0.01$ was applied (figure 5.10). Results are presented in the following table

	full group	**near group**	**mid group**	**far group**
	accuracy sensitivity specificity [%]	**accuracy sensitivity specificity [%]**	**accuracy sensitivity specificity [%]**	**accuracy sensitivity specificity [%]**
whole brain grey matter	56.3 62.5 56.3	68.7 62.5 75.0	57.8 59.4 56.3	59.4 62.5 56.25
T-map at p<0.001	60.9 62.5 59.4	82.8 78.1 87.5	64.1 84.4 43.8	60.0 62.5 59.38
T-map at p<0.05 FWE corrected	45.3 50.0 41.3	81.3 84.4 78.1	71.9 68.8 75.0	53.1 50.0 56.3
Single voxel with maximum T value	65.0 70.4 59.6	78.13 78.13 78.13	79.7 78.13 81.3	50.0 100 0

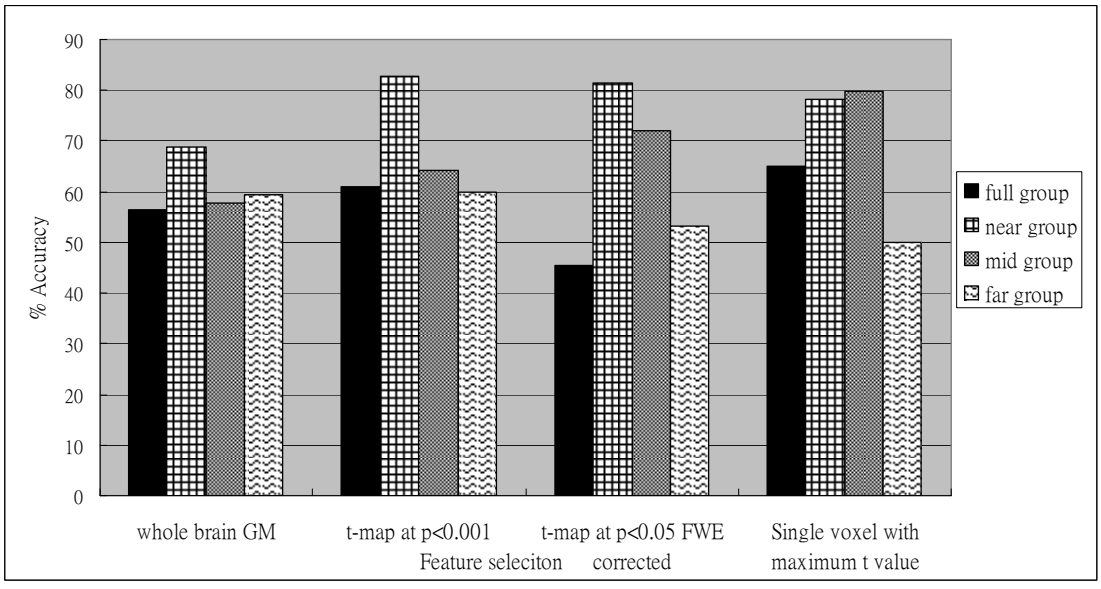


Figure 5.10 Classification performances for preclinical Huntington Disease

This figure shows the classification accuracy of PSCs using soft-margin SVC with different input features in each subgroup.

Classification performance was satisfactory considering that PSCs do not have clinical symptoms. Our study also provided evidence supporting the gene prediction model. The probability of developing unequivocal signs of HD within a period of time is strongly correlated with prediction performance. In general, feature selection improves the prediction accuracy for both near and mid groups, however, performance was around chance level for the far group, regardless of feature selection criteria. Subjects in this group were estimated 20 years or more from developing signs of disease. It was very likely some people the group did not manifest atrophy, hence those people would have similar patterns as normal controls. When we combined all three groups, the classification performance had similar accuracy to the far group. This was possibly due to many PSCs in the far group having similar pattern of normal controls thus affecting the classifier during training. In one of our previous studies, we applied SVM on diffusion weighted imaging (DWI) data, and 82% classification was achieved with whole brain data (Kloppel et al., 2008a). Although the cohort was not the same, the subjects in the DWI study were estimated to be an average of 19 years from clinical manifestation. This strongly implies that DWI may provide more salient information related to structural changes for PSCs than grey matter atrophy. Nevertheless, in the framework of kernel methods, we can combine features by means of generating kernels from weighted linear combination of kernels. If both DWI and T1-MRI of the same subjects are available, the appropriately weighted combination of both modalities should allow even better performance.

5.3.4 Automatic feature selection using Gaussian processes

When training samples are scarce, feature selection techniques are shown to improve the performance of classification. A popular method called recursive feature elimination (RFE) was developed for SVM. RFE works by eliminating feature which

contributes the least to the change of margin width. i.e. $\underset{i}{\operatorname{argmin}}(\left| \|\mathbf{w}\|^2 - \|\mathbf{w}_{(i)}\|^2 \right|)$, where $\|\mathbf{w}\|^2$ is the margin width of the full features, and $\|\mathbf{w}_{(i)}\|^2$ is the margin width when the i-th feature is removed. If we use a linear kernel, $\underset{i}{\operatorname{argmin}}(\left| \|\mathbf{w}\|^2 - \|\mathbf{w}^{(i)}\|^2 \right|)$ this will be equivalent to $\underset{i}{\operatorname{argmin}}(|\mathbf{w}_i|)$ (Fan et al., 2005; Guyon and Elisseeff, 2003; Guyon et al., 2002; Rakotomamonjy, 2003). The original SVM-RFE removes features one by one. However, when there are too many features this is computationally unfeasible and many tend to remove multiple features at once, which can give sub-optimal results. However, non-Bayesian methods suffer from the need to use cross validation in order to determine the optimal number of features. If we have only limited samples, splitting the data into training, validation and testing sets will reduce the number training samples. Besides, cross validation for feature selection normally requires a lot of computation due to the high dimensional nature of the imaging data. In this small additional project, we tested the ability of automatic feature selection using marginal likelihood maximisation with Gaussian Process Classification (section 5.1.3). We used the external dataset, which generated the statistical map in the main project, with additional scans in the work. The scans were acquired from a 1.5T Siemens Sonata scanner (T1-weighted MDEFT sequence). The dataset consisted of 40 PSCs and 40 controls. The average years of onset, calculated from the CAG repeat in the dataset, is about 15 years.

Standard pre-preprocessing procedures were applied, but with additional Gaussian smoothing (6mm). We partitioned the whole normalized and Jacobian scaled grey matter into five regions: the left and right striatum, the left and right hippocampus, and everything left. In principle, we could have automatically parcellated the brain using standard templates (e.g. AAL template). The covariance

matrix takes the form

$$\mathbf{C} = \theta_1 \mathbf{K}_{lh} + \theta_2 \mathbf{K}_{rh} + \theta_3 \mathbf{K}_{ls} + \theta_4 \mathbf{K}_{rs} + \theta_5 \mathbf{K}_{else} + \theta_6 \tag{5.24}$$

We then run the GPC to optimise the hyperparameters by maximising the marginal likelihood (5.18). Because optimising the weights for each kernel is equivalent to weighting the importance of each region. Automatic feature selection was achieved in the training process, so we did not need to run the time consuming three-way split cross validation. Standard leave one out cross validation was applied to test the generalisation performance. When we used the whole brain as features, i.e. $1 = \theta_1 = \theta_2 = \theta_3 = \theta_4 = \theta_5$, the classification accuracy from GPC is 52.5%. (sensitivity 50%, specificity 55%). When automatic feature selection was used, the accuracy increases to 67.5% (sensitivity 65%, specificity 70%).

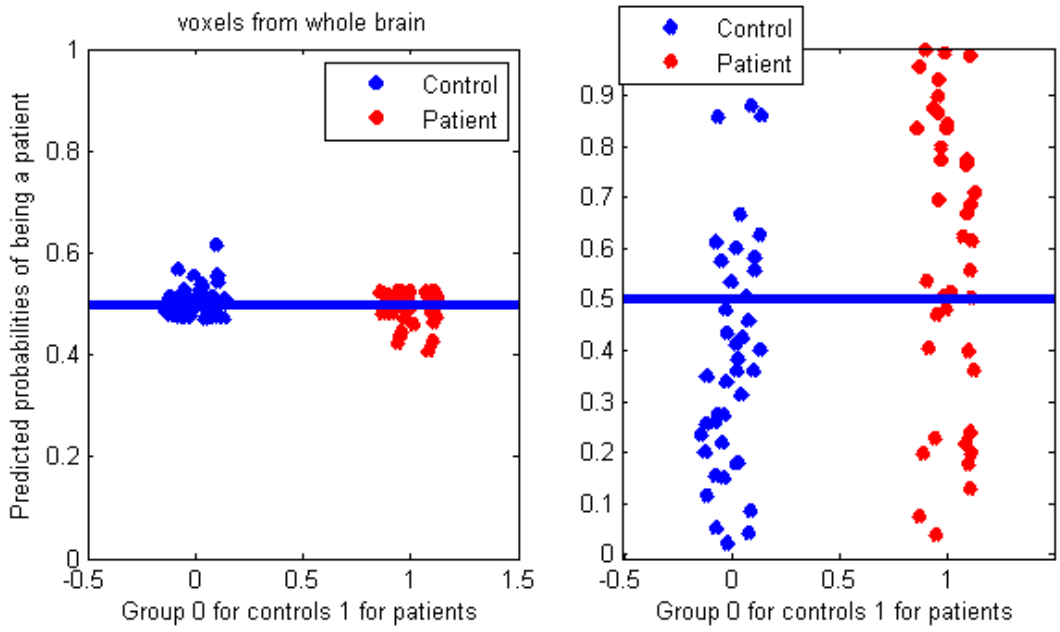


Figure 5.11 Classification performances of Gaussian Process Classification

The left figure shows the predicted probabilities of both PSCs and controls using whole GM. The accuracy was about chance level (52.5%). The right figure shows the prediction, when automatic feature selection was applied using maximising marginal likelihood. The accuracy increased to 67.5%.

To examine which regions were most important for classification, we trained the

GPC with the full training set. We then normalize the hyperparameters except the constant term to calculate the percentage of contribution from each region (figure 5.12). Interestingly, the kernel generated from left striatum contributed 99% of the covariance matrix. This result is in agreement with previous VBM studies (Kassubek et al., 2004; Thieben et al., 2002), which showed higher t values in the left striatum.

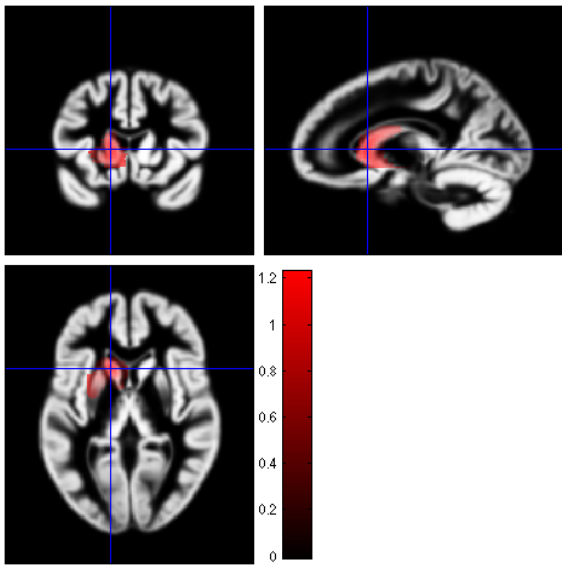


Figure 5.12 Contribution of each region to the construction of covariance matrix
The figure shows the contribution of five regions to the covariance matrix (5.24). From the figure we can see that the left striatum dominated over other regions.

5.4 Application: Multi-class Classification of fMRI Patterns by Kernel Regression Methods

This application is based on the method mentioned in 5.1.4. We compared our multi-class classification system against a multi-class classification system built from a combination of binary classifiers, at both single subject level and at group level.

5.4.1 Introduction

There has been increasing interest in the application of pattern classification, especially with support vector machines (SVM), to fMRI analysis. In these

approaches, fMRI volumes are treated as the input features and the patterns reflect the strength of BOLD signal. However, there are strong temporal correlations in fMRI time series, especially as a result of the delay and smoothing due to the hemodynamic response (HRF). For block designed experiments, investigators have typically either applied a shift to account for the hemodynamic delay, or they have averaged the volumes in the block (Cox and Savoy, 2003a; Mourao-Miranda et al., 2005). Both strategies ignore the temporal correlation due to hemodynamic convolution. An alternative method, which preserves the HRF information, is to fit a GLM to obtain parameter maps (sometimes called the "beta maps") for each block or event (Eger et al., 2008). However, all these methods involving temporal compression greatly reduce the size of the training set, impairing the training process and exacerbating the relative sparsity of events common in many fMRI designs. Here, we propose a novel approach, which treats the fMRI pattern as a regression problem, and predict the fMRI pattern with a regression machine rather than a classification machine. We adapt a match filter as the final decision function to compare the predicted time series with the canonical time series pattern, and then select the best matched pattern as the predicted class.

5.4.2 Materials and methods

The dataset used for this work was also used in previously published papers (Hardoon et al., 2007; Mourao-Miranda et al., 2007; Mourao-Miranda et al., 2006). Functional MRI scans from 16 male right handed healthy US college students (age 20–25), without any history of neurological or psychiatric illness, were acquired. After the study was explained to them, all subjects gave written informed consent to participate in the study. The study was performed in accordance with the local Ethics Committee of the University of North Carolina. The data were collected at the Magnetic Resonance Imaging Research Center at the University of North Carolina on

a 3T Siemens Allegra Head-only MRI system. The fMRI runs were acquired using a T2* sequence with 43 axial slices (slice thickness, 3 mm; gap between slices, 0 mm; TR = 3 s; TE = 30 ms; Flip angle = 80 dgrees; FOV = 192 × 192 mm; matrix, 64 × 64; voxel dimensions, 3 × 3 × 3 mm). In each run, 254 functional volumes were acquired.

The experimental stimuli were in a standard block design. It was a passive experiment with visual stimuli, so subjects were not required to react to the stimuli. The visual stimuli were categorized into three different active conditions: viewing unpleasant (dermatological diseases), neutral (people) and pleasant images (girls in bikini). Each active condition was followed by a resting condition (fixation) with equal duration. In each run, there were 6 blocks of the active condition (each consisting of seven images volumes) alternating with resting (fixation) of seven image volumes. Six blocks of each of the three stimuli were presented in random order.

We applied the pre-processing procedures described in 3.2.2. Briefly speaking, the fMRI volumes were realigned and resliced. Grey matter (GM) masks were generated by segmenting the fMRI volume for each subject. Similar to PBAIC competition 2007 (section 4.3), we masked out non GM voxels, which were less likely to contain BOLD signals, to increase the signal to noise ratio. For group level prediction, we further normalised each subject into the population template by the DARTEL toolbox. A linear kernel was generated for each subject from GM masked fMRI series in the native space. Also one linear kernel for all subjects was computed from all the GM masked spatially normalized fMRI series. Linear detrending was applied using the residual forming matrix (equation 3.14).

To test the prediction accuracy, we applied the multi-class method using a regression machine mentioned in section 5.1.4. We used both kernel ridge regression (KRR) and relevance vector regression (RVR) for training the regression machine. Covariance was chosen as the metric to measure the similarity between the predicted

profile and the canonical profile. Because we used different pre-processing from previously published papers (Mourao-Miranda et al., 2007; Mourao-Miranda et al., 2006), we also retested the performance of support vector classification (SVC) by combining three “one versus one” classifiers, which were the same procedures used for previously published results. We applied the averaging, beta-map, and spatial-temporal techniques to compress the kernel. Details on efficient ways to compress the kernel are described in chapter 3.

5.4.3 Results and discussion

Leave one block out cross validation was performed to estimate the prediction performance for each subject. The results were then averaged across all subjects. The accuracy of predicting experimental stimuli from fMRI volumes in each single subject was very high when we applied our approach. 100% classification accuracy was obtained for six subjects and an average of 94% accuracy was achieved across 16 subjects. KRR and RVR resulted the same accuracy, but the computation time of RVR was about 500 times more than for KRR. We achieved similar results for SVC compared with previously published work. The best classification accuracy for SVC was 83% using averaging blocks. The results are presented in the following table.

Single Subject

Prediction Accuracy %	Multiclass with RVR	Multiclass with KRR	SVC (block average)	SVC (beta-map)	SVC (spatial temporal)
Average Accuracy	93.8	93.8	83.3	83	64.2
Unpleasant	94.8	93.8	79.2	79.2	65.6
Neutral	91.7	92.7	84.4	84.4	59.3
Pleasant	94.8	94.8	86.5	85.4	67.7

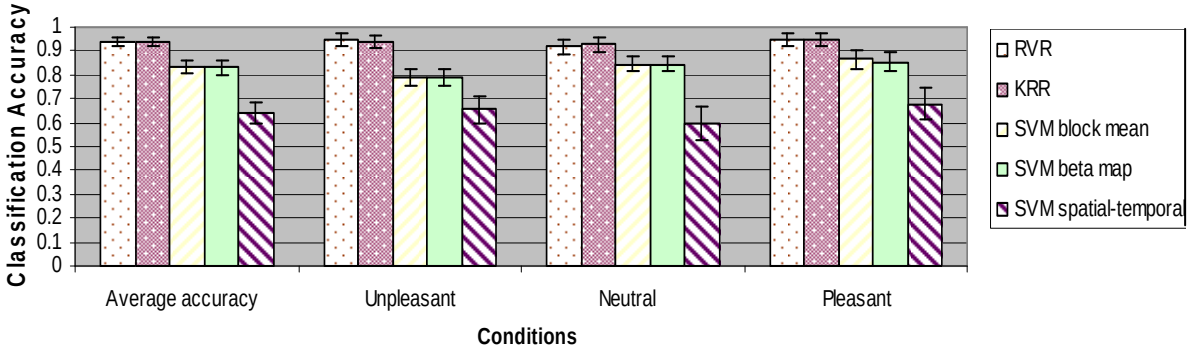


Figure 5.13 Multi-class classification performance for single subject

The figure shows the classification accuracy of the multi-class regression machine and SVC for single subject.

And the confusion matrix for multi-class with RVR and SVC (temporally compressed by block average) is given by

Predicted %		Actual		
Multiclass with RVR		Unpleasant	Neutral	Pleasant
Predicted	Unpleasant	93.75	3.13	2.1
	Neutral	3.13	91.67	3.13
	Pleasant	3.13	5.2	94.79

Predicted %		Actual		
SVC (block average)		Unpleasant	Neutral	Pleasant
Predicted	Unpleasant	79.17	5.21	7.29
	Neutral	6.25	84.38	6.25
	Pleasant	13.54	8.33	86.46

Notice that the columns of unpleasant and neutral in SVC (block average) do not sum to 1. This was due to ambiguous cases, which were unclassifiable. i.e. the condition was predicted as all three conditions from three “one versus one” SVC classifiers. Our method of multi-class with RVR did not show strong prediction bias, but SVC seemed to have a strong bias toward mistaking the condition of unpleasant as pleasant. This could be explained from our clustering analysis shown in figure 3.13. The analysis showed that the temporally averaged fMRI pattern under unpleasant stimuli were more similar to the pattern under pleasant stimuli than neutral stimuli. In general, our multi-class regression based method outperformed conventional multiclass SVC approaches regardless of the type of temporal compression. This implies that additional information for discriminating among different conditions is encoded in the temporal profile of the HRF. The superior performance could also be due to sufficient training samples, which allowed more accurate estimation of the distribution. In this work, each regression machine was trained on a 252 by 252 kernel (18 blocks times 14 volumes per block), whereas SVC was trained on an 18 by 18 kernel after temporal compression. Some information must be lost in the process of compression. It is not surprising that spatial temporal compression had the worst predicting accuracy (figure 5.13), because if we recall the way that spatial temporal compression works in figure 3.9. Spatial temporal compressed the original kernel by summing only the diagonal components of each sub block. Relative to block average or beta-map, spatial temporal ignores more information in the original kernel. Both beta-map compression and block averaging had nearly the same prediction accuracies, which implies that these approaches did not preserve much temporal information after the compression.

For the group level prediction, leave one subject out cross validation was performed. This involved training from the fMRI volumes of 15 subjects, and then making predictions about the subject left out. The prediction accuracies were then

averaged across all subjects. The multi-class classifier using regression machine performed worse than SVC with temporal compression. The result of 83.3% accuracy was achieved from multi-class using RVR. In contrast, SVC with block averaging achieved 95.1% classification accuracy. Even SVC with spatial temporal compression obtained 94.1% classification accuracy. The results are presented in the following table.

Multiple Subjects

Prediction Accuracy %	Multiclass with RVR	Multiclass with KRR	SVC (block average)	SVC (beta-map)	SVC (spatial temporal)
Average Accuracy	83.3	80.1	95.1	94.8	94.1
Unpleasant	83.3	79.2	94.8	93.8	93.8
Neutral	81.3	78.1	94.8	94.8	93.8
Pleasant	85.4	84.4	95.8	95.8	94.8

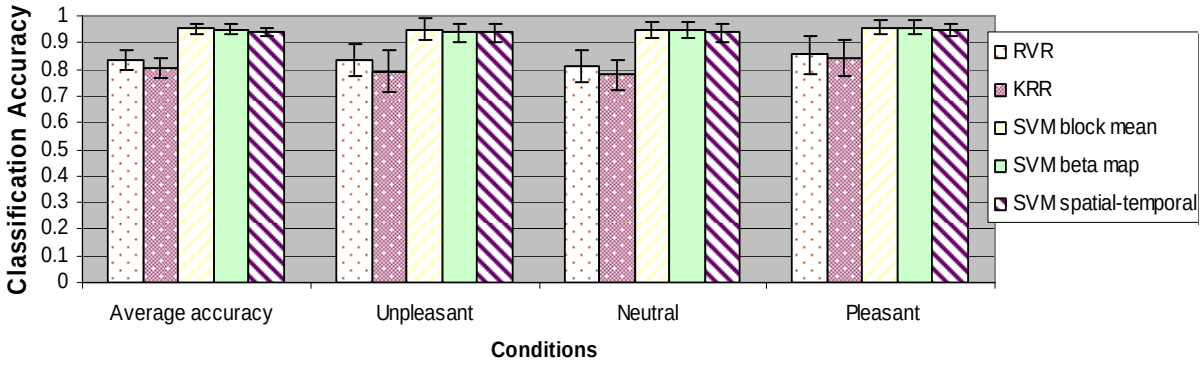


Figure 5.14 Multi-class classification performances for multiple subjects

The figure shows the classification accuracy of the multi-class regression machine and SVC for multiple subjects.

The confusion matrix for multi-class with RVR and SVC (block average) is given by

Predicted %		Actual		
Multiclass with RVR		Unpleasant	Neutral	Pleasant
Predicted	Unpleasant	83.33	14.58	10.42
	Neutral	8.33	81.25	4.17
	Pleasant	8.33	4.17	85.42

Predicted %		Actual		
SVC (block average)		Unpleasant	Neutral	Pleasant
Predicted	Unpleasant	95.83	2.08	4.17
	Neutral	4.17	94.79	1.04
	Pleasant	0	3.13	94.79

There were no ambiguous cases in SVC for multiple subjects, so the columns in the confusion matrix summed to one. The prediction for SVC seemed to be less biased, unlike predicting for single subjects, where there was the tendency to mis-identify the condition of unpleasant as pleasant. However, multi-class classification using RVR seemed to be biased towards predicting the conditions of neutral or pleasant as unpleasant.

We suspect that the reason why the multi-class regression machine did not work well in multiple subjects, as opposed to in single subjects, was due to the inter-subject variability of the HRF. The approach of the multi-class regression machine is more sensitive to variation of HRF than is SVC with temporal compression. If we could characterise the HRF for each individual, the prediction performance of the regression

machine should be at least compatible with SVC. One possible explanation of why SVC improved in the case of multiple subjects was because of sufficient training samples. For investigators interested in real time fMRI online prediction, our results suggest that if there are sufficient fMRI volumes available to train the prediction system offline, SVC will be the recommended classifier. However, if the prediction system is going to be trained online i.e. trained on the first half of the experiment, our multi-class regression machine should be a better choice.

In addition, we also trained the multi-class regression machine with fMRI volumes corresponding to the active condition only. The standard multi-class regression machine trained all fMRI volumes, the only difference between each regression machines was the input target variable, which was the corresponding regressor in the design matrix (figure 5.5). For example, if we trained a regressing machine to predict condition 2, the corresponding elements of the target vector for other conditions would be set to zero. In the new approach, we did not train fMRI volumes of other conditions, but only the fMRI scans of the corresponding condition. The prediction accuracy of single subject for KRR was 84% (unpleasant 87.5%, neutral 76%, pleasant 88.5%), but the accuracy for RVR was much lower, only 57.6% (unpleasant 29%, neutral 70.9%, pleasant 72.9%). For multiple subjects, KRR achieved 81.3% correct classification (pleasant 82.3%, neutral 81.3%, unpleasant 80.2%), and RVR achieved 62.9% (pleasant 65.5%, neutral 82.3%, unpleasant 40.6%). In this approach, sparse methods did not work well. This could be caused by an inability to capture subtle differences among conditions using sparse methods, because predicted profiles from different conditions by RVR all appeared very similar.

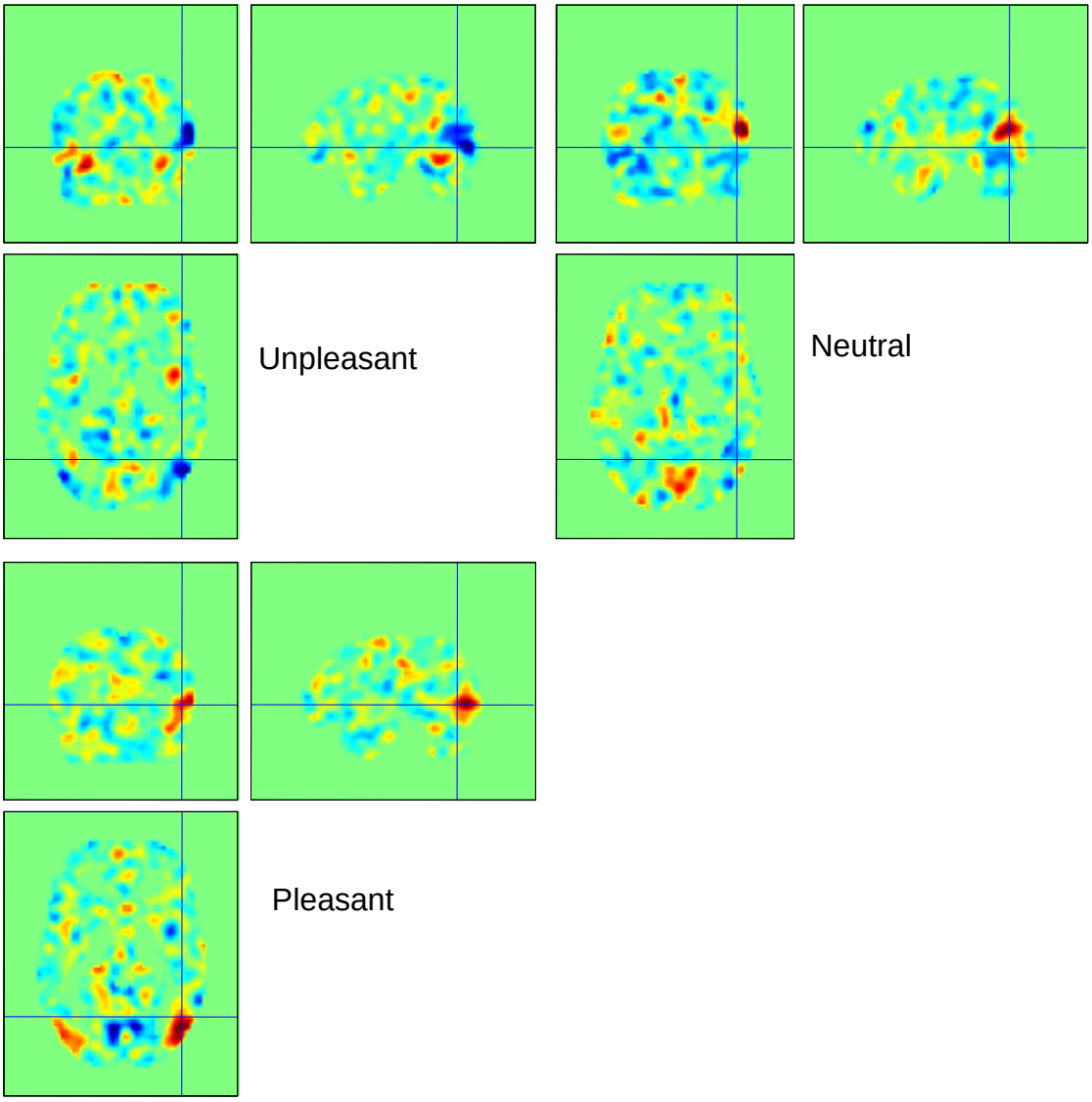


Figure 5.15 Weight maps for all conditions from training all 16 subjects with RVR

These figures show the weight map that predicts the temporal profile. These maps were generated by training three RVR machines with the corresponding target variables. Red indicates positive values and blue indicates negative values.

For visualisation purpose, we also computed the corresponding weight map for all three conditions by training all the subjects with RVR. Notice in the weight map, there were large areas of negative weightings in the visual cortex for unpleasant stimuli. This does not necessarily mean those areas experienced deactivation during stimulation, but that those areas may have had less activity compared with neutral or pleasant stimuli.

5.5 Decoding Neuronal Ensembles in the Human Hippocampus

This was a joint work with Demis Hassabis, whose interests are in hippocampal functions. The hippocampus appears to be important in spatial navigation (Maguire et al., 2000). There is debate about how hippocampal neurons code such information. Nevertheless, Hassabis believed the neuronal coding related to navigation has a representation in the population of neurons, and is detectable from BOLD signals. Because the change of signals may be subtle, we applied multivariate pattern classification using high spatial resolution fMRI. Specifically we applied a "searchlight" based method (Haynes et al., 2007; Kriegeskorte et al., 2006) to explore regions in the temporal lobe that contain information that discriminates positions of the subject in a virtual environment.

5.5.1 Introduction

Studies had shown that the brain encodes information about the environment using a large population of neurons (Buzaki, 2004). Previously, recording of single, or small numbers of, neurons have demonstrated the memory-related response of hippocampal place cells, which fire invariantly when an animal is at a particular spatial location (Moser et al., 2008). It is difficult to examine what information such place cells represent at a neuronal population level, as recording thousands of hippocampal neurons simultaneously is not currently possible. Recently, examining neuronal codes through high resolution fMRI has been achieved (Haynes et al., 2007). The hypothesis was that although the fine grain of the neuronal representations are below the spatial resolution of fMRI, the ensemble activity of such distributed patterns could be used to predict the perceptual state or intention of an individual. We believed that if decoding from focal hippocampal fMRI signals was successful, this

would have significant implications for understanding how information may be represented within neuronal populations in the human hippocampus. To test our hypothesis, we used an interactive virtual reality (VR) environment with first person view to test spatial navigation of the subjects in the fMRI scanner.

5.5.2 Materials and methods

Prior to scanning, subjects were trained to familiarise with the VR environment. The navigation task allowed participants to move in the VR environment using a 4-button control pad. There were two rooms, namely a blue room and a green room. These two rooms had the same dimensions and were designed to minimise the impact of irrelevant sensory inputs. There were four target positions (A, B, C, D) in each room. Participants were asked to navigate as quickly and accurately as possible among these four positions (figure5.16).

Each room was visited 20 times during the scanning session giving 40 “environment blocks” in total. Within each room, every target position was visited 14 times in total. The cue for visiting the next destination would appear after the subject had arrived at the current target position and looked down at the floor. There would be a 13 second resting period after visiting 2~4 target position in the room. Then the subject would be placed randomly in the initial position of one of the rooms. To maintain the concentration of the subject, eight catch trials were included that involved an incidental visual task. During the count down, the numbers were normally displayed in white text, but occasionally one would flash red. Subjects were asked to press the trigger button when spotting a red number. The catch trials spread randomly, but they would always appear at the end of a block. Details and exact timing can be found in the published work (Hassabis et al., 2009).

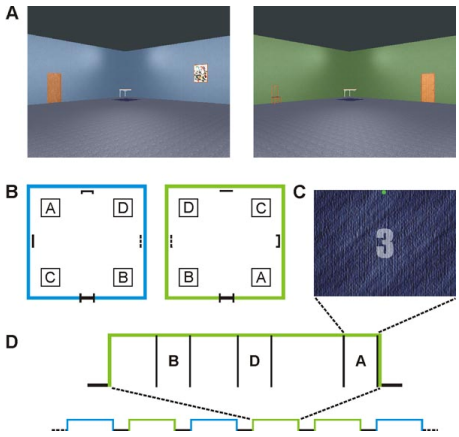


Figure 5.16 Virtual reality environment for the navigation task

(A)The top two figures show the blue and green rooms. (B) This figure shows the layout of the 4 different positions in both blue and green rooms. (C) When the subject reached the destination, they would press the button to look at the floor, and a 5 second countdown was given (the '3' in this figure indicates the countdown), the count down was followed by the text label for the next destination. (one of the 4 positions). (D) After completing 2 to 4 navigation trials in the room (based on the time they spent), there would be a 13s fixation period. Then the subject would be replaced in one of the rooms.

Actually, we had different experimental design for the first batch of fMRI experiments. Initially, we had four sessions in the experiment, and each session contained 6 environment blocks. The sessions were interrupted by turning off scanning, which made the temporal detrending perform poorly. The exact reason was unknown, but interrupting the process of acquiring fMRI volumes did change the properties of the low frequency drift. As a result of this, when we displayed the first few principle components of the data, we found scans in the four sessions came from different distribution (even after detrending). Unlike univariate analysis, each session can be modelled by different regressors. The multivariate classification system assumes all training samples come from the same distribution. When the training samples came from four slightly different distributions, the prediction accuracy would deteriorate. We made the decision to discard all the scans, and re-scanned four

subjects out of the original ten using the new design. In the new design, the time of each environment block (room switching) was reduced to decrease the effect of low frequency noise and increase the predictability. We also scanned the subjects in one long session without interrupting the scanning.

The scanning was done in a 3T Siemens Allegra scanner (in-plane resolution=1.5x1.5mm^2, matrix=128x128, field of view= 192x192mm^2, 35 slices acquired in an interleaved order; slice thickness=1.5mm without gap, TR=3.57s, TE=30ms, flip angle=90 degrees). Every subject completed the navigation with different speed, so the scanning time for each subject was not fixed.

We pre-processed the fMRI data by realigning and re-sampling the scans with spatial smoothing using 3mm FWHM Gaussian kernel. Linear detrending was applied to the pre-processed images directly. Because we used the searchlight method, applying the detrending to the images would be more efficient than applying it to kernels generated from each searchlight region. Next, we convolved the image data with the canonical HRF to increase the signal to noise ratio, which effectively acted as a low-pass temporal filter. To compensate for the delay induced by the inherited hemodynamic delay in BOLD and the additional HRF smoothing, all onset times were shifted forwards in time by three volumes yielding an approximation to the 12s delay (HRF peaks at 6s) given a TR of 3.57s. The first volume and the last four volumes of each environmental block were discarded to reduce the effects of appearing suddenly in a room, and to exclude catch trials. Three classification tasks were carried out to (1) indentify which of two target positions in the same environment that the participant was standing (A vs. B and C vs. D); (2) classify all four target positions in the same environment (A vs. B vs. C vs. D); (3) classify which of the two rooms the participant was in (green vs. blue). For task 2, we combined six “one versus one” binary SVCs.

To identify regions in the temporal lobe that contained information for discriminating the position of the participant, we applied the searchlight approach (Kriegeskorte et al., 2006). This approach utilised the multivariate information in local voxels. Thus, for one voxel v_i, we selected the voxels in a sphere centred at v_i with a radius of three voxels. This yielded total 123 voxels in the spherical volume of interest (VOI). We then generated the linear kernel from each VOI for classification. No temporal compression was applied because there were only two volumes per corresponding target position. Soft-margin SVC was applied to train the pattern with C fixed at 1 (empirically determined). To estimate the predicting accuracy, we used a leave one block out cross validation. We had 40 environment blocks, each containing about 12 volumes. Also, there were 56 position blocks, each containing two volumes. During the training, only volumes in the relevant blocks were used i.e. when training A vs. B in green room, we only used volumes acquired while the subject was standing at those two green room positions. Each volume within a test block was individually classified in the cross-validation, and the classification accuracy was calculated by the percentage of correctly classified volumes, as opposed to correctly classified blocks. For visualisation, we generated a “prediction map”, where the accuracy in each voxel represented the accuracy of the searchlight sphere centred at that particular voxel.

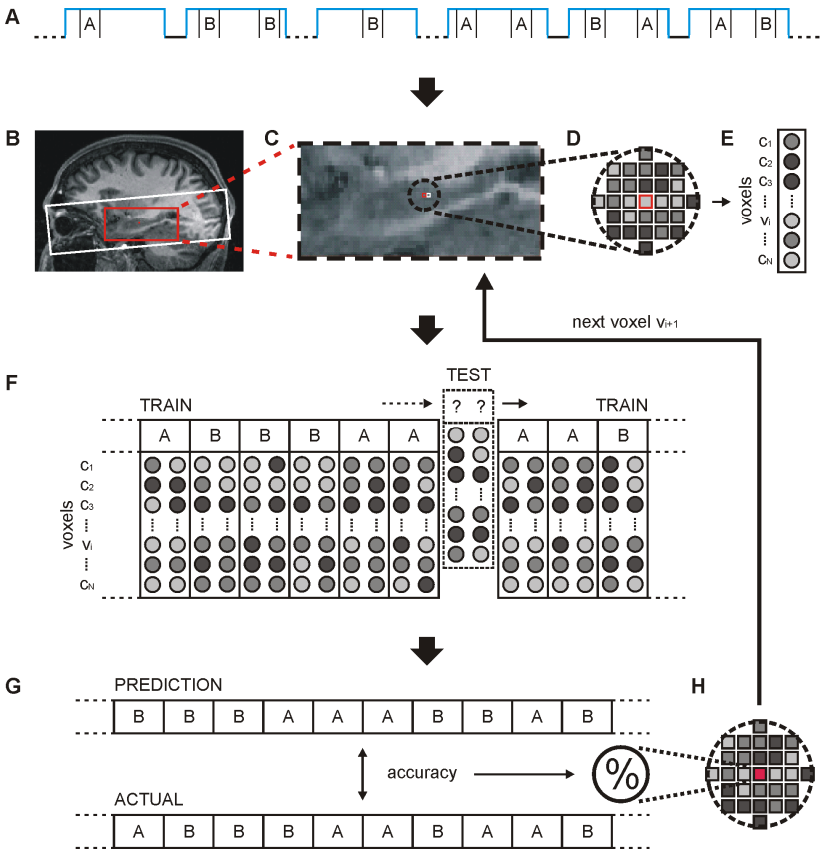


Figure 5.17 Illustration of searchlight pattern classification

(A)The top figure shows the fMRI volumes that corresponded with the participant standing at position A and B in the blue room. (B~E) This shows how we selected the searchlight sphere. (F,E) illustration of the "leave one block out" cross validation. The prediction accuracy was then stored at the central voxels of the searchlight

To account for multiple comparisons issue, we applied permutation tests (Nichols and Holmes, 2002) to simulate the null distribution and estimate the significance threshold. We repeated the classification and cross-validation procedure 100 times with a different random permutation of training labels for each classification tasks for each subject. The threshold was then estimated from the 95% quantile of the accuracies generated from the permutation trials, equating to a family-wise confidence level of $p<0.05$. This threshold was computed for each voxel in the prediction map, and prediction accuracies that did not reach significance were masked out from the map.

5.5.3 Results and discussion

The first classification task was to indentify which of two target positions in the same environment the participant was standing at (A vs. B and C vs. D). The prediction map showed that voxels in the body-posterior hippocampus bilaterally were crucial for classifying position. The findings were highly consistent across participants

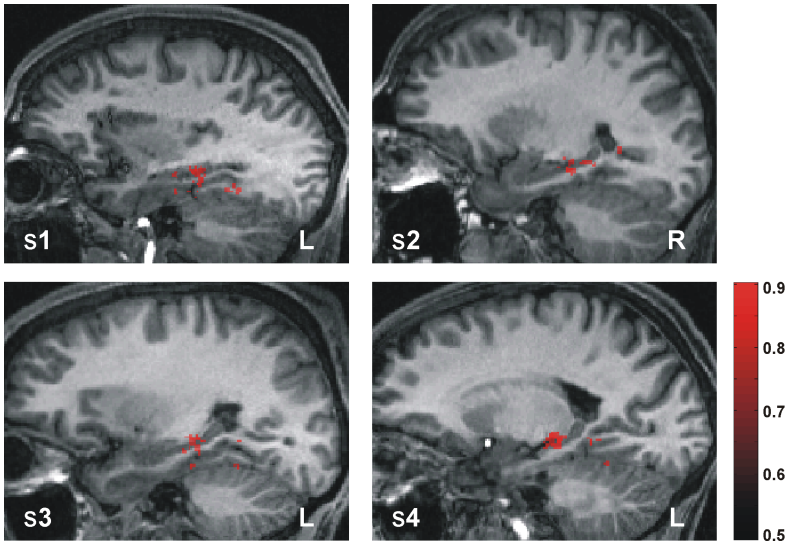


Figure 5.18 Prediction map of classifying two target positions

The prediction map was overlaid on the anatomical image to identify regions that contained information for discriminating two target positions (A vs. B and C vs. D)

The second classification task was to indentify which of all four target positions in the same environment the participant was standing at. The prediction map revealed a focal cluster of voxels in the body-posterior of the hippocampus bilaterally. These findings were also very consistent across subjects.

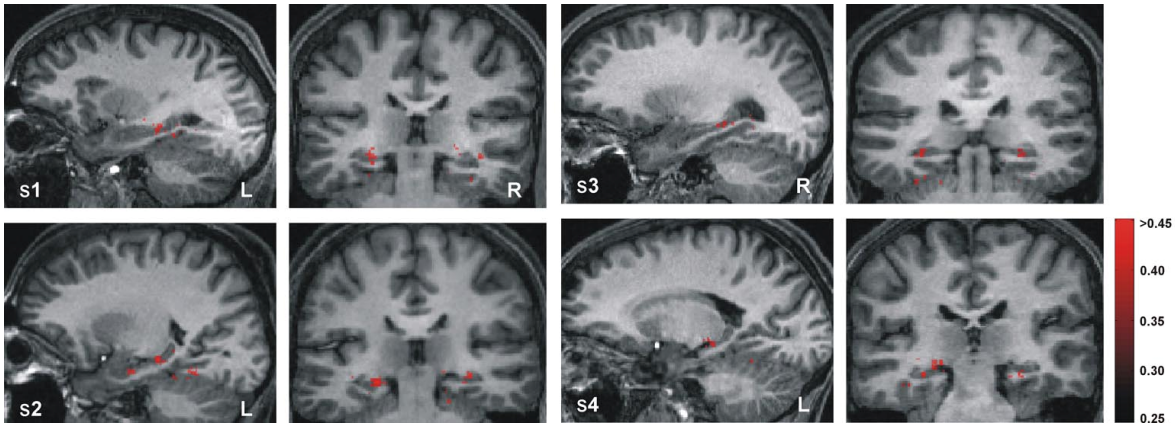


Figure 5.19 Prediction map of classifying four target positions

The prediction map was overlaid on the anatomical image to identify regions that contained information for discriminating among four target positions (A vs. B vs. C vs. D)

The third classification task was to idenify which of the two rooms the participant was in (green vs. blue). The prediction map showed that voxels in the posterior parahippocampal gyrus bilaterally were important for discriminating between rooms.

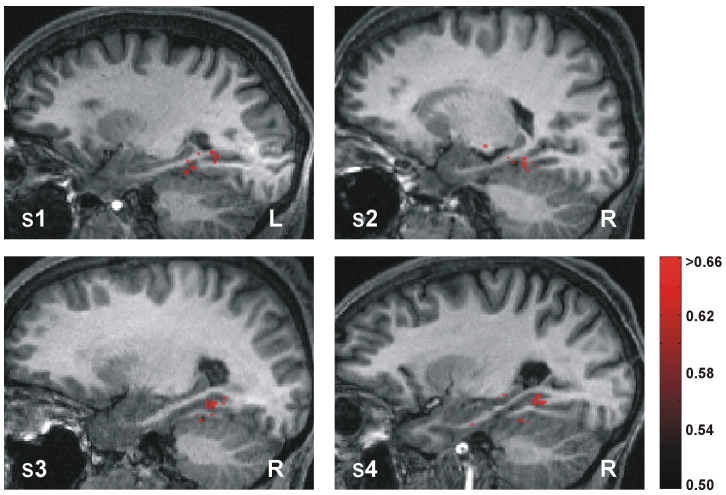


Figure 5.20 Prediction map from classifying environments

The prediction map was overlaid on the anatomical image to identify regions that contained information for discriminating between two environments (green room vs. blue room)

To summarise the findings, we found there was a significantly higher proportion of voxels in the hippocampus than in the parahippocampal gyrus containing information for identifying position. For environment classification, there was a

higher proportion of voxels active in the parahippocampal gyrus than the hippocampus for all participants. Our results showed the possibility of decoding spatial information from the pattern of fMRI signals across spatially distributed voxels in the human hippocampus. This implies that the hippocampal neurons may represent spatial locations by large populations of neurons. The permutation test and the consistency of prediction maps across subjects suggested that it was unlikely that the patterns were just random. Although, the mechanism that caused the fMRI pattern is unknown at the neuronal level, the finding suggested a different view to prevailing theories, that there may be an underlying functional organization to the hippocampal neural code.

5.6 Prognostic and Diagnostic Potential of the Structural Neuroanatomy of Depression

This work was a collaboration with Cynthia Fu at the Institute of Psychiatry at Kings College London, for which I did all the analysis and coding. There are no neurobiological diagnostic markers for psychiatric disorders currently. Usually, the diagnosis of depression is based on self reported symptoms without using evidence from neurobiological markers. In this work, we tried to examine the potential of discriminating between MDD patients and normal controls using anatomical MRI.

5.6.1 Introduction

Major depression disorder (MDD) is characterised by persistent low mood or anhedonia, and by behavioural and cognitive distractions that disturb daily functioning. Neuroimaging studies have shown structural and functional patterns associated with MDD (Drevets, 2000; Lyoo et al., 2004). A previous study had also

shown the possibility of distinguishing between depressed and healthy individuals from the fMRI pattern during the task of viewing sad facial expressions. An accuracy of 86% was achieved (Fu et al., 2008). However, the use of fMRI is still not as prevalent as structural MRI as a diagnostic tool. Therefore, we tried to test the diagnostic performance for depression using anatomical scans. Unlike Alzheimer's disease and Huntington's disease, which have known pathology of degeneration in localised areas of the brain, MDD is not a neurodegenerative disease. Nevertheless, subtle shape variation and degeneration may still induce patterns in the structural MRI that could be utilised for diagnosis. Generally, global cerebral volume is slightly lower in patients with depression than in healthy subjects, and the pattern of difference seems to be distributed. There were reports of reduced volume in hippocampus (Campbell et al., 2004), anterior cingulate (Caetano et al., 2006), and middle frontal cortices (Bremner et al., 2002) for depressed patients.

5.6.2 Materials and methods

In the analysis, there were 37 right-handed patients (28 women, mean age 41.9 years) meeting Diagnostic and Statistical Manual of Mental Disorder-IV (DSM-IV) (Spitzer, 1994) criteria for major depression by Structured Clinical Interview for DSM-IV. All patients had been free of psychotropic medication for a minimum of four weeks. There were 37 right-handed healthy control cases (CC) matched for age, gender, and IQ with no history of psychiatric disorder, neurological disorder, or head injury resulting in a loss of consciousness. Their Hamilton Rating Scale for Depression (HRSD) (Hamilton, 1960) was less than or equal to 7. Eighteen depressed patients had participated in a treatment study with the antidepressant medication fluoxetine (20mg daily) and twelve patients were treated with cognitive behavioural therapy (CBT). Nine patients achieved remission from the antidepressant medication, and six patients achieved remission from CBT.

Anatomical MRI data were acquired on a 1.5T IGE LX system (General Electric). The acquisition protocol involved collecting 120 slices using a dual echo, fast spin echo sequence (T2 weighted and proton density weighted), coronal orientation, in-plane resolution 0.8mm, slice thickness 3mm, TR=4s, TE=15ms and 105ms, echo train length=8.

The image pre-processing procedures are described in section 3.2.1. Briefly, images were first segmented by SPM into GM and WM, and then imported into a rigidly aligned space. The GM and WM were iteratively registered to the population mean by the DARTEL toolbox. Jacobian scaled spatially normalised GM images were generated, which were smoothed using a 6mm FWHM Gaussian kernel. Finally, the linear kernel is computed from the pre-processed data.

Voxel-Based Morphometry (VBM) was used to localise any volumetric differences between the controls and patients. No voxels survived a family-wise error (FWE) correction or even a false discovery rate (FDR) correction. When the threshold was set to uncorrected $p<0.001$, the SPM map showed small clusters of a few voxels in size, which were scattered across the brain. This suggested that the anatomical pattern of difference between MDD and controls might need to be characterised at a finer spatial scale than afforded conventionally. To increase the classification accuracy, feature selection was applied. Feature selection, the process of selecting a subset of features that may be most useful for prediction (Guyon and Elisseeff, 2003), is important for high dimension data when only a few sparsely distributed features are informative. We implemented a simple univariate t-map filtering. First, the t-value and degrees of freedom was estimated for each voxel in the training set using equations for unequal sample sizes and unequal variance. The t-map was then converted into a p-map. Different thresholds of uncorrected p-values were chosen, such that voxels higher than the threshold were masked out. We then applied

hard-margin SVC to discriminate between patients and controls. Leave one out cross validation was used to estimate the classification accuracy. It is important to realise that we only used the training set for computing the p-map, so the test set remained completely independent. However, reporting the best cross validation accuracy out of all possible selected feature sets is biased towards being too optimistic (a multiple comparisons issue). Three way cross validation should really be used to estimate the generalisation performance of SVC with feature selection. Because of the intensive computations required for three way cross validation, we chose another approach to estimate the p-value of the predicted accuracy. We applied permutation tests (Nichols and Holmes, 2002) to simulate the null distribution and estimated the significance value of the predicted accuracy using the threshold that achieved the best leave one out accuracy. We fixed the threshold for feature selection and repeated the cross-validation procedure 300 times with a different random permutation of the training labels. The significance level was then estimated from one minus the percent quantile of the accuracies generated from the permutation trials. For example, if the leave one out accuracy using correct labels achieved 67.6% accuracy, which ranked the 292nd in the permutation trails (ascending order), then the p value was estimated as 1-292/300=0.0267.

A one class classifier was also applied to test the overlap between classes. To make the comparison less biased, each group was resampled to a subgroup of 30 subjects. We then tested the percentage overlap between one class and another. i.e. train the one class classifier on patients, and test the percentage of controls who were classified as patients, and vice versa. This procedure was repeated 30 times, and the results were averaged.

5.6.3 Results and discussion

The prediction performance was very poor (45% accurate) when we used the

whole brain grey matter. With feature selection, the best prediction accuracy was 67.6% (sensitivity 64.9%, specificity 70.3%) when we fixed the threshold of the p-map obtained from the training set at $p<0.005$. Using the permutation test, we estimated the corresponding p-value to be 0.0267.

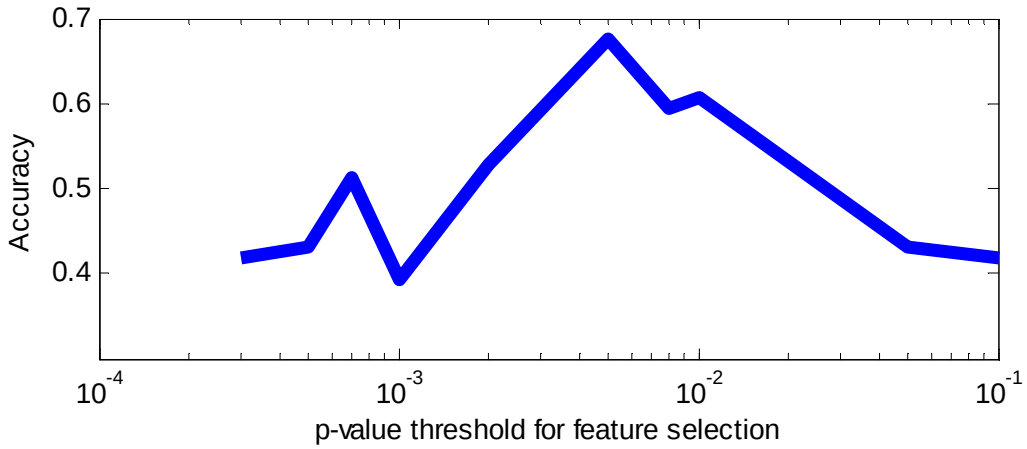


Figure 5.21 Classifying MDD and control with different thresholds

This plot shows the leave one out cross validation accuracy for classifying depressed patients and normal controls by varying the thresholds in the feature selection. The p-map was calculated from the training set, and only voxels lower than the threshold were selected.

Because each cross validation trial used different features, we could not generate the weight map from SVC easily. To localise regions that were most important for classification, a frequency map was computed. The value in each voxel of the frequency map indicates the rate that that voxel was selected as a feature in the cross validation. From the frequency map based on the best prediction accuracy (threshold $p<0.005$), regions in right subgenual anterior cingulate, medial frontal gyrus, superior temporal cortex, precuneus, hippocampus, thalamus, left inferior parietal cortex, occipital cortex, and cerebellum, all contributed to discrimination between patients and controls.

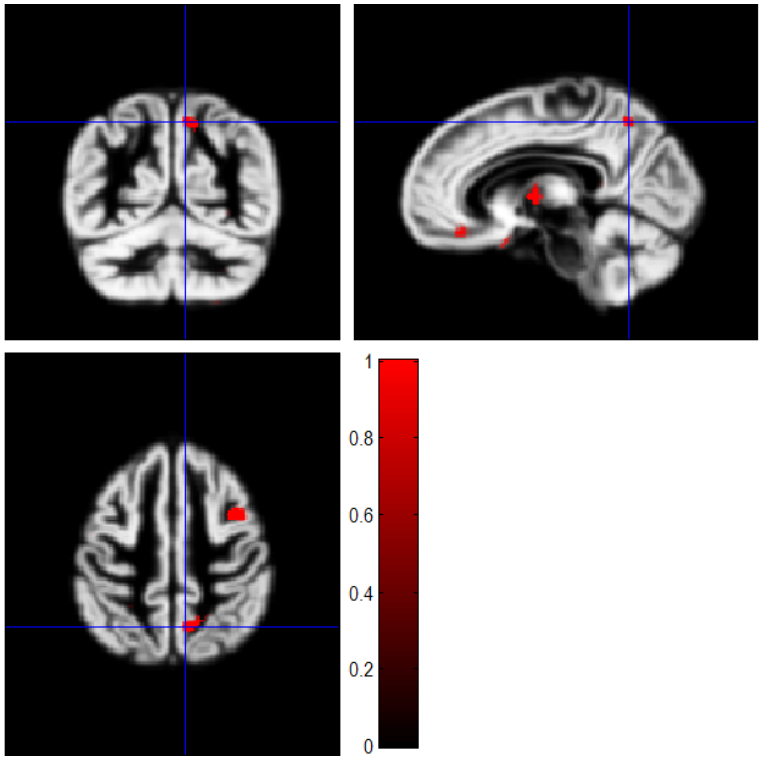


Figure 5.22 Frequency map for separating MDD from controls (threshold p<0.005).
This figure illustrates regions that were most often selected in the cross validation trials with the threshold p<0.005.

We suspected the reason that the classification result was not as good as for classifying AD or HD, and also the poor VBM result, was due to class overlapping. This suggests that some of the MDD patients had similar anatomical brain patterns to the controls, so the classifier would not be trained properly. To test this hypothesis, we applied the one-class classifier to both MDD, and CC separately. When the MDD group was trained, we found the average proportion of CC classified as MDD is 73%. When we changed the training set to the control group, interestingly, there were 92% of MDD classified as CC. The result may suggest that the anatomical pattern in the CC group had broader variations than in the MDD group. In other words, the structural pattern of the MDD group is likely to be a subset of the CC group. This could also be due to outliers in the control group. Because the assessment of depression is very subjective, some subjects in the control group might be depressed

for a period of time slightly prior to the recruitment, but not depressed during the assessment of HRSD. There was a large proportion of overlap between these two groups. However, we also noticed that about 25% of healthy controls had distinctive patterns compared with the MDD group.

We also tried to classify between subjects who achieved clinical remission to treatment with the antidepressant medication fluoxetine and the subjects who received the antidepressant with residual symptoms. The anatomical MRI scans were acquired before the treatment. Surprisingly, 88.9% classification accuracy was achieved (88.9% of patients in clinical remission (sensitivity) and 88.9% patients with residual symptoms (specificity)). The significance level estimated using the permutation test was 0.01 with the fixed threshold set to $p<0.005$.

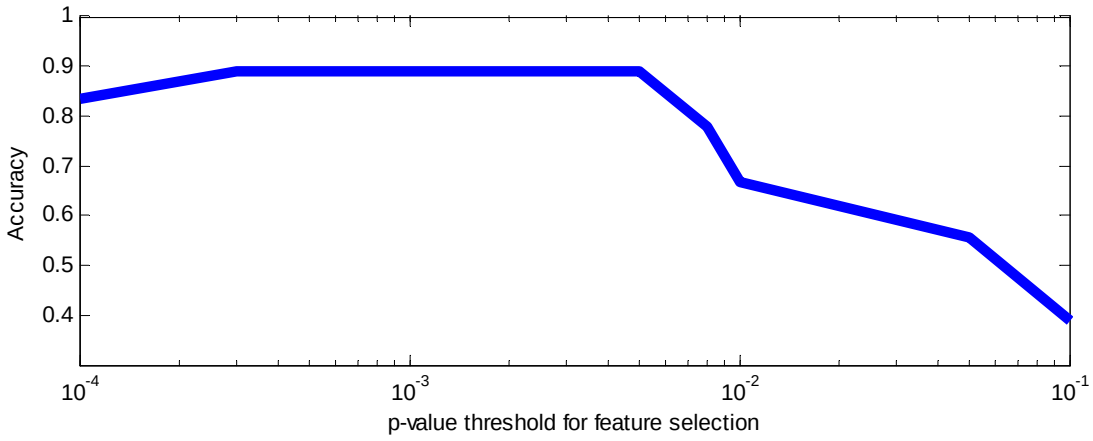


Figure 5.23 Classifying patients who achieved remission or not with different thresholds

This plot shows the leave one out cross validation accuracy for classifying patients who achieved remission with the antidepressant and those who did not improve after taking the antidepressant.

Regions in right rostral anterior cingulate cortex, left posterior cingulate cortex, left middle frontal gyrus, right occipital cortex, orbitofrontal cortices bilaterally, right superior frontal cortex, and left hippocampus contributed highly to the classification.

The anatomical information did not show any ability to predict who would achieve clinical remission from CBT.

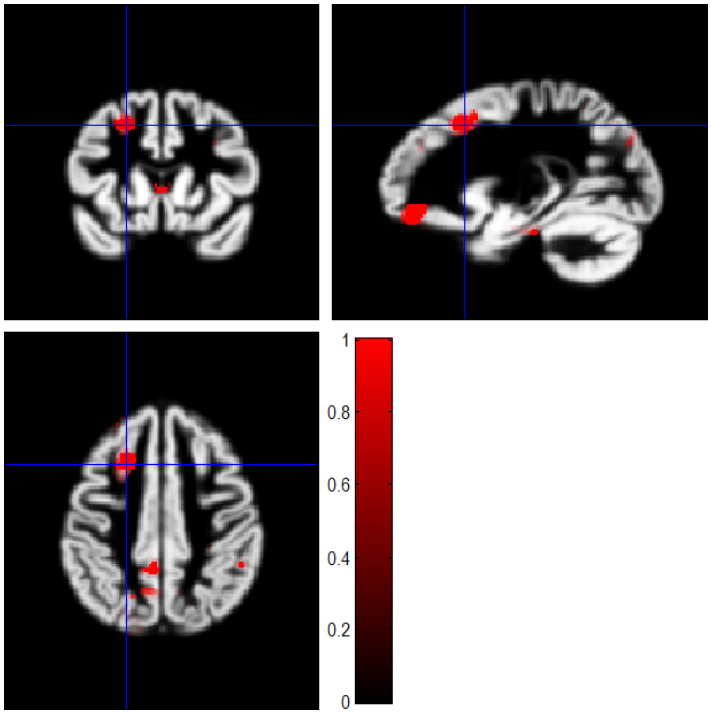


Figure 5.24 Frequency map from classifying patients achieving remission or not (threshold p<0.005).

This figure illustrates regions which were most often selected in the cross validation trials with the threshold p<0.005 between patients who achieved remission and patients who had residual symptoms.

Because there is no clear neurobiological mechanism that can describe the cause of depression, it was expected that the structural MRI would be inferior to fMRI for the clinical diagnosis of depression. From the analysis, we found a large proportion of depressed patients exhibited similar anatomy to normal subjects. However, for acutely depressed patients who received medication, the effect of the treatment was found to be significantly correlated with brain structure. The high predictability of clinical remission suggests an initial step towards the development of personalized medicine. If similar predictabilities could be obtained from other antidepressants, the most effective medications could be prescribed to the patient based on the prediction of

treatment effects. If we use probabilistic classifiers such as Gaussian processes, a combined utility function may provide the basis for making decisions that optimally balance the trade-off between side effects and treatment benefit (and financial cost).

Chapter 6

Discussion

Contents

The main focus of this thesis has been to introduce prediction based analyses that utilise state of the art machine learning methods. This thesis has described a number of pattern recognition algorithms that can be used for both functional MRI decoding and anatomical MRI prediction. These include various classification and regression algorithms. Most algorithms were not originally invented for this thesis, and many of implementations of the algorithms can be found freely on the internet, or could be fairly easily implemented in MATLAB by following the description in the text. The most frequently used methods in this thesis were Support Vector Classification (SVC), Relevance Vector Regression (RVR), and Kernel Ridge Regression (KRR).

Practical applications were also presented to demonstrate the performance of those algorithms. A total of eight applications were presented in chapters 4 and 5. There were two applications on fMRI regression prediction, one application on anatomical MRI regression prediction, two applications on fMRI classification, and three applications on anatomical MRI classification.

6.1 Original Contributions of This Thesis

A few original contributions have been made within this thesis. Some original ideas came from my supervisors. Many of the contributions simply involve combining different parts of pre-existing methods in a new way.

In chapter 3, a residual forming matrix is introduced to remove confounding factors or low frequency drifts from the kernel directly. This operation is more efficient than removing covariates voxel by voxel. Although a similar operation was introduced in a paper I contributed to (Friston et al., 2008), it was not formulated for general kernel methods, such as SVM or RVM. In this thesis, the operation is further extended to temporal compression for fMRI data. Matrix operations on the kernel generated from all the fMRI volumes can yield an equivalent kernel computed from

the “beta-map” or “averaged block”. Although Spatial-temporal compression can not be expressed as simple matrix operations, a generic algorithm (figure 3.9) is provided to compute all forms of temporal compression from the original kernel.

A number of novel ideas can be found in the fMRI competition 2007 (PBAIC 2007) in chapter 4. These include applying temporal shifts to train the regression machine, model fitting with average template (predicting instruction), masking functional regions, and utilising information from other conditions i.e. use hit weapons, hit people, and hit fruit to predict search weapons, search people, and search fruit. These novel approaches subsequently resulted our beating more than 40 other teams to win in the competition, especially as the two teams who achieved joint second place (Team Princeton and Team Maastricht) applied similar algorithms as ours (ridge regression and RVR respectively). The other novel idea in chapter 4 is the prediction of clinical ratings using structural MRI, and clinical ratings are compared based on their prediction accuracy. This idea was recognized by the abstract reviewers of OHBM 2007, as it was considered suitable for an aural presentation.

Chapter 5 introduced two main original ideas. The first is the use of marginal likelihood maximization for automatic feature selection. This avoids the time consuming three-way cross validation. The second idea is the multi-class classifier using regression methods. This approach uses temporal information and was shown to significantly outperform conventional classifiers for single subject prediction. This idea was also recognized by the OHBM 2008 reviewers, and was also selected for an oral presentation.

6.2 General Conclusions

Kernel methods were shown to be powerful for predicting linear and non-linear patterns in brain MRI data. In chapter 3, the definition of the kernel was introduced,

and efficient operations were demonstrated for constructing various kernels. Some common non-linear kernels can be computed directly from the linear kernel. Also, linear operations can efficiently remove confounds from the kernel or temporally compress it. Conventionally, researchers tend to use eigen-decomposition or singular value decomposition to reduce the dimensionality of the input features, but this is redundant for kernel algorithms as solutions can be sought in the space of the input kernel, where the computation is bounded by the number of samples rather than the number of input features. This characteristic is favourable for high dimensional imaging data. However, prior knowledge is still required to define the similarity measure. Extracting relevant features to predict the target variable, which can be labels (classification) or continuous variable (regression), from the raw image data depends on one's understanding of what information is encoded. For example, kernels generated from Jacobian determinants may encode different information to kernels generated from "velocity fields" that encode brain shape. In principle, it would also be possible to combine the advantages of both discriminative models and generative models using Fisher kernels (Jaakkola and Haussler, 1999).

Kernel regression methods predicted BOLD signals accurately for some experimental conditions. Both PBAIC 2006 and PBAIC 2007 allowed a comparison among a diverse range of approaches for making predictions from brain imaging data. As in any model comparison problem, it allowed the most accurate approach to be selected from a range of candidates. Our approaches proved to be superior by showing competitive results. In general, objective ratings had higher predictability than subjective ratings, such as valence and arousal. It was also shown empirically that pre-processing would have a higher impact than the choice of algorithm or model, assuming optimal parameters of the models were found. For example, insufficient detrending in PBAIC 2007 led to relatively poor performance (figure 4.9). RVR was

applied to most of the applications, because it does not require free parameters, and the optimisation of RVR is faster than GPR in our implementations. No systematic comparisons between algorithms were performed in this thesis, because the “no free lunch theorem” (Duda et al., 2000), says that there is no algorithm that is superior to others across all problems and contexts. The optimal algorithm may be different for each dataset. However, from our empirical results, it seems that RVR generally performs well for both anatomical and functional MRI datasets. Perhaps this is because MRI datasets satisfy the Gaussian assumption well.

Unlike kernel regression methods, kernel classification algorithms have greater variability among their model structures and assumptions. For instance, Fisher’s linear discriminant assumes equal covariance for classes; logistic regression, GPC, and RVC assume a Bernoulli likelihood model; the philosophy behind SVC is based on structural risk minimization. The maximum margin approach in SVC effectively prevents overfitting of the training data. Because the posterior and marginal distributions of GPC and RVC are analytically intractable, Laplace approximations are used, which may reduce the accuracy of the estimation. In practice, SVC generally performed more accurate binary classifications than most of the Bayesian methods, but Bayesian methods provide probabilistic measures which can be more easily integrated into a decision theoretic framework. Most importantly, the applications in chapter 5 demonstrated the feasibility of an automatic diagnostic system. The classification system showed comparable performance with clinical radiologists in the task of discriminating between AD patients and normal controls. For neurodegenerative diseases affecting large regions in the brain such as Alzheimer’s disease, whole brain features can achieve accurate classification. However, when the regions with anatomical change are relatively small, selecting salient features increases the performance of classification when only small training datasets are

available.

For investigators doing prediction based fMRI analysis, it is recommended to have long sessions or short sessions without turning off the scanning. From our experience, turning off scanning disturbed the classification performance.

6.3 Directions for Future Research

Overall, decoding patterns in both fMRI and structural MRI were achieved successfully in this thesis. These achievements can lead to two main directions of research.

6.3.1 Clinical decision support system

Medical examinations are becoming increasingly complicated, and more measurements can be acquired from the patient. Physicians relying on only few markers may risk the chance of misdiagnosis. Pooling all available information should allow the most accurate diagnoses to be achieved, but there is a cognitive limit for normal humans to retain and utilise all the details in a useful way. To reduce the cost of medical systems without sacrificing the quality of diagnosis, it seems inevitable that computer aided diagnoses or clinical decision support systems will become increasingly used. Modern computers are already powerful enough to perform very complicated calculations in real time. Because Moore's law still holds today, multi-core computers, parallel computing, and terabyte storages system are likely to become prevalent in the future. Computation power can be expected to take over some aspects of the physician's analytic ability (maybe not for physician's experiences and intuitions). Developing models and algorithms to integrate measurements from different examinations will be crucial. For example, physiological biomarkers can achieve 90% classification of AD from blood samples (Ray et al., 2007). Combining such measurements with MRI data could push the diagnostic

accuracy even higher. If individual's genetic information is added, 99% accuracy may be achieved one day. Anatomical MRI may also predict the effect of treatment as shown in section 5.6. The achievement of personalised medicine may only become feasible by adopting computer aided clinical systems. One of the main obstacles comes from the lack of large datasets for training such systems, which in turn arises from the reluctance of investigators in the neuroimaging field to share their primary data.

6.3.2 Prediction based fMRI analysis

Classification from fMRI patterns has been shown to work successfully by many people. The current approaches mainly use direct signal changes from the baseline, but patterns of functional connectivity may be more robust to noise. With sufficiently high classification accuracies, practical applications may be performed. For example, the fMRI patterns of arousal can be found for a particular subject by training with specifically designed experiments (calibrating phase). Then different advertisements are shown to the same subject. Based on the previously found patterns, it may be possible to measure the level of arousal for different advertisements. This type of method could also be used to look for interaction between cognitive functions. For example, training could be based on listening to pleasant and unpleasant music, and then the classifier is applied to fMRI scans of viewing pleasant and unpleasant images. The classification performance could provide a similarity measure between different cognitive processes, for instance by assessing the similarity between the feeling of listening to pleasant music to that from looking at pleasant images.

Pattern classifiers can also be applied to real-time fMRI experiments. For example, it may be possible to train a classifier that can predict the move of the subject playing "rock-paper-scissors" in the scanner. However, the hemodynamic delay may complicate this type of experiment. Generally, pattern classifiers will be

useful for experiments involving real-time feedback.

Future developments could potentially lead to directions of research that raise important ethical questions. For example, insurance companies may wish to know what kinds of health predictions could be made from data about their clients. Similarly, there may also be legal implications if the procedures allowed very accurate decoding of mental states. Currently, our ability to make predictions from neuroimaging data has only limited accuracy, but the techniques are likely to become much more accurate in future. Many of the potential ethical implications are not yet known. To quote Niels Bohr:

"Prediction is very difficult, especially about the future."

Appendix A: Basic proves

Unbiased variance

Sample variance $\sigma_n^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 = \mathrm{E}(x^2)\text{-}\overline{x}^2$, where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

$$
\begin{aligned}
&\mathrm{E}(\sigma_n^2) \\
&=\mathrm{E}(\frac{1}{\mathrm{n}}\sum(x-\overline{x})^2) \\
&=\frac{1}{\mathrm{n}}\sum(\mathrm{E}(x^2)-\mathrm{E}(2x\overline{x})+\mathrm{E}(\overline{x}^2)) \\
&=\frac{1}{\mathrm{n}}(n\mathrm{E}(x^2)-2n\mathrm{E}(\overline{x}^2)+n\mathrm{E}(\overline{x}^2)) \\
&=(\sigma_n^2+\overline{x}^2-\mathrm{E}(\overline{x}^2)) \\
&=(\sigma_n^2+\overline{x}^2-\mathrm{var}(\overline{x})-\mathrm{E}(\overline{x})^2) \qquad \text{note:}\ \overline{x}^2-\mathrm{E}(\overline{x})^2=0 \\
&=(\sigma_n^2-\frac{1}{n^2}\mathrm{var}(\sum x)) \qquad \text{note:}\ \mathrm{var}(nx)=n^2\,\mathrm{var}(x) \\
&=(\sigma_n^2-\frac{1}{n^2}n\sigma_n^2) \\
&=\sigma_n^2(1-\frac{1}{n}) \\
&=\sigma_n^2\frac{n-1}{n}
\end{aligned}
$$

Appendix B: Demo codes

Least squares logistic regression

```
function w=logisticRegression_LS(X,y,lam);
for iter=1:100,
    f1   = X*w;
    f    = exp(f1)./(1+exp(f1));
    df   = X'*(2*(f-y).*f.*(1-f));
    % Approx 2nd derivative
    %d2f  = (B'*diag(2*(f.*(f-1)).^2)*B);
    % True 2nd deriv
    d2f  =
(X'*diag(2*f.*(1-f).*(2*f-3*f.^2+y.*(2*f-1)))*X);
    % Regularization
    d2f  = d2f + lam*eye(size(d2f));
    df   = df  + lam*w;
    %update
    old_w=w;
    w    = w   - d2f\df;
    dw=w-old_w;
    if dw'*dw/numel(w)<1e-6
        disp(iter);
        break;
    end
end;
```

Binary logistic regression

```
function w = logistic_binary(X, y,lam)

[n, m] = size(X);
w = zeros(m,1);
for i=1:60
   z=X*w;
   f=1./(1+exp(-z));
   deriv= f.*(1-f);
   R=spdiags(deriv,0,n,n);
   w=w-(X’*R*X+eye(m)*lam)\(X'*(f-y)+ones(m,1)*lam);
end
```

References

1992. Evidence-based medicine. A new approach to teaching the practice of medicine. JAMA 268, 2420-2425.

Aguirre, G.K., Zarahn, E., D'Esposito, M., 1998. The variability of human, BOLD hemodynamic responses. Neuroimage 8, 360-369.

Akaike, H., 1974. A new look at the statistical model identification. IEEE Transactions on Automatic Control 19, 716-723.

Ashburner, J., 2007. A fast diffeomorphic image registration algorithm. Neuroimage 38, 95-113.

Ashburner, J., Csernansky, J.G., Davatzikos, C., Fox, N.C., Frisoni, G.B., Thompson, P.M., 2003. Computer-assisted imaging to assess brain structure in healthy and diseased brains. Lancet Neurol 2, 79-88.

Ashburner, J., Friston, K.J., 1999. Nonlinear spatial normalization using basis functions. Hum Brain Mapp 7, 254-266.

Ashburner, J., Friston, K.J., 2000. Voxel-based morphometry--the methods. Neuroimage 11, 805-821.

Ashburner, J., Friston, K.J., 2001. Why voxel-based morphometry should be used. Neuroimage 14, 1238-1243.

Ashburner, J., Friston, K.J., 2005. Unified segmentation. Neuroimage 26, 839-851.

Ashburner, J., Friston, K.J., 2008. Computing average shaped tissue probability templates. Neuroimage, 333-341.

Ashburner, J., Hutton, C., Frackowiak, R., Johnsrude, I., Price, C., Friston, K., 1998. Identifying global anatomical differences: deformation-based morphometry. Hum Brain Mapp 6, 348-357.

Attwell, D., Iadecola, C., 2002. The neural basis of functional brain imaging signals.

Trends Neurosci 25, 621-625.

Barnes, J., Scahill, R.I., Boyes, R.G., Frost, C., Lewis, E.B., Rossor, C.L., Rossor, M.N., Fox, N.C., 2004. Differentiating AD from aging using semiautomated measurement of hippocampal atrophy rates. Neuroimage 23, 574-581.

Baxter, L.C., Sparks, D.L., Johnson, S.C., Lenoski, B., Lopez, J.E., Connor, D.J., Sabbagh, M.N., 2006. Relationship of cognitive measures and gray and white matter in Alzheimer's disease. J Alzheimers Dis 9, 253-260.

Bishop, C.B., 2006a. Pattern recognition and machine learning.

Bishop, C.B., 2006b. Pattern recognition and machine learning. Springer.

Bishop, C.M.a.L., J., 2007. Generative or Discriminative? getting the best of both worlds. Bayesian Statistics 8, 3–23.

Blennow, K., De Leon, M., Zetterberg, H., 2006. Alzheimer's disease. Lancet(British edition) 368, 387-403.

Braak, H., Braak, E., 1991. Neuropathological stageing of Alzheimer-related changes. Acta Neuropathol 82, 239-259.

Brand, M., 2002. Incremental singular value decomposition of uncertain data with missing values. Proceedings of the 2002 European Conference on Computer Vision, 707--720.

Bremner, J.D., Vythilingam, M., Vermetten, E., Nazeer, A., Adil, J., Khan, S., Staib, L.H., Charney, D.S., 2002. Reduced volume of orbitofrontal cortex in major depression. Biological Psychiatry 51, 273-279.

Brickman, A.M., Habeck, C., Zarahn, E., Flynn, J., Stern, Y., 2007. Structural MRI covariance patterns associated with normal aging and neuropsychological functioning. Neurobiol Aging 28, 284-295.

Brown, M.P., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares, M., Jr., Haussler, D., 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc Natl Acad Sci U S A 97, 262-267.

Buchanan, M., 2007. The Social Atom: Why the Rich Get Richer, Cheaters Get Caught, and Your Neighbor Usually Looks Like You. Cyan Books and Marshall Cavendish.

Buyya, R., Yeo, C.S., Venugopal, S., Ltd, M.P., Melbourne, A., 2008. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities.

Buzaki, G., 2004. Large-scale recording of neuronal ensembles. Nature Neuroscience 7, 446-451.

Caetano, S.C., Kaur, S., Brambilla, P., Nicoletti, M., Hatch, J.P., Sassi, R.B., Mallinger, A.G., Keshavan, M.S., Kupfer, D.J., Frank, E., 2006. Smaller cingulate volumes in unipolar depressed patients. Biological Psychiatry 59, 702-706.

Campbell, S., Marriott, M., Nahmias, C., MacQueen, G.M., 2004. Lower hippocampal volume in patients suffering from depression: a meta-analysis. Am Psychiatric Assoc, pp. 598-607.

Carlson, T.A., Schrater, P., He, S., 2003. Patterns of activity in the categorical representations of objects. Journal of Cognitive Neuroscience 15, 704-717.

Carlton Chu, Ni, Y., Tan, G., Saunders, C.J., Ashburner, J., 2009. Kernel Regression for fMRI pattern prediction. Neuroimage (Special issue), In review.

Caselli, R.J., Reiman, E.M., Locke, D.E., Hutton, M.L., Hentz, J.G., Hoffman-Snyder, C., Woodruff, B.K., Alexander, G.E., Osborne, D., 2007. Cognitive domain decline in healthy apolipoprotein E epsilon4 homozygotes before the diagnosis of mild cognitive impairment. Arch Neurol 64, 1306-1311.

Chau, W., McIntosh, A.R., 2005. The Talairach coordinate of a point in the MNI space: how to interpret it. Neuroimage 25, 408-416.

Chen, L.-F., Liao, H.-Y.M., Ko, M.-T., Lin, J.-C., Yu, G.-J., 2000. A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition 33, 1713-1726.

Chen, P.H., Lin, C.J., Scholkopf, B., 2005. A tutorial on v-support vector machines.

Applied Stochastic Models in Business and Industry 21, 111-136.

Cheng, C.W., Su, E.C., Hwang, J.K., Sung, T.Y., Hsu, W.L., 2008. Predicting RNA-binding sites of proteins using support vector machines and evolutionary information. BMC Bioinformatics 9 Suppl 12, S6.

Cox, D.D., Savoy, R.L., 2003a. Functional magnetic resonance imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. Neuroimage 19, 261-270.

Cox, D.D., Savoy, R.L., 2003b. Functional magnetic resonance imaging (fMRI) rain reading? detecting and classifying distributed patterns of fMRI activity in human visual cortex. Neuroimage 19, 261-270.

Cox, R.T., 1946 Probability, Frequency and Reasonable Expectation. American Journal of Physics 14, 1-13

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

Csernansky, J.G., Hamstra, J., Wang, L., McKeel, D., Price, J.L., Gado, M., Morris, J.C., 2004. Correlations Between Antemortem Hippocampal Volume and Postmortem Neuropathology in AD Subjects. Alzheimer disease and associated disorders 18, 190-195.

Cunnington, R., Windischberger, C., Deecke, L., Moser, E., 2002. The preparation and execution of self-initiated and externally-triggered movement: a study of event-related fMRI. Neuroimage 15, 373-385.

Davatzikos, C., Fan, Y., Wu, X., Shen, D., Resnick, S.M., 2008. Detection of prodromal Alzheimer's disease via pattern classification of magnetic resonance imaging. Neurobiology of Aging 29, 514-523.

Davatzikos, C., Genc, A., Xu, D., Resnick, S.M., 2001. Voxel-based morphometry using the RAVENS maps: methods and validation using simulated longitudinal atrophy. Neuroimage 14, 1361-1369.

Demirci, O., Clark, V.P., Magnotta, V.A., Andreasen, N.C., Lauriello, J., Kiehl, K.A.,

Pearlson, G.D., Calhoun, V.D., 2008. A Review of Challenges in the Use of fMRI for Disease Classification/Characterization and A Projection Pursuit Application from A Multi-site fMRI Schizophrenia Study. Brain Imaging and Behavior 2, 207-226.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39, 1-38.

Domingos, P., 1998. Occam's two razors: The sharp and the blunt. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, 37-43.

Drevets, W.C., 2000. Neuroimaging studies of mood disorders. Biological Psychiatry 48, 813-829.

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Pattern Classification, 2 ed. Wiley-Interscience.

Efron, B., 1979. Bootstrap methods: another look at the jackknife. The Annals of Statistics, 1-26.

Efron, B., Tibshirani, R., 1993. An introduction to the bootstrap. Chapman & Hall/CRC.

Eger, E., Ashburner, J., Haynes, J.D., Dolan, R.J., Rees, G., 2008. fMRI activity patterns in human LOC carry information about object exemplars within category. J Cogn Neurosci 20, 356-370.

Evans, A.C., Collins, D.L., Mills, S.R., Brown, E.D., Kelly, R.L., Peters, T.M., 1993 3D statistical neuroanatomical models from 305 MRI volumes. Nuclear Science Symposium and Medical Imaging Conference, San Francisco,CA, USA, pp. 1813-1817.

Fan, Y., Batmanghelich, N., Clark, C.M., Davatzikos, C., 2008a. Spatial patterns of brain atrophy in MCI patients, identified via high-dimensional pattern classification, predict subsequent cognitive decline. Neuroimage 39, 1731-1743.

Fan, Y., Rao, H., Hurt, H., Giannetta, J., Korczykowski, M., Shera, D., Avants, B.B., Gee, J.C., Wang, J., Shen, D., 2007a. Multivariate examination of brain abnormality

using both structural and functional MRI. Neuroimage 36, 1189-1199.

Fan, Y., Resnick, S.M., Wu, X., Davatzikos, C., 2008b. Structural and functional biomarkers of prodromal Alzheimer's disease: a high-dimensional pattern classification study. Neuroimage 41, 277-285.

Fan, Y., Shen, D., Davatzikos, C., 2005. Classification of structural images via high-dimensional image warping, robust feature extraction, and SVM. Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv 8, 1-8.

Fan, Y., Shen, D., Gur, R.C., Gur, R.E., Davatzikos, C., 2007b. COMPARE: classification of morphological patterns using adaptive regional elements. IEEE Transactions on Medical Imaging 26, 93-105.

Folstein, M.F., Folstein, S.E., McHugh, P.R., 1975. "Mini-mental state". A practical method for grading the cognitive state of patients for the clinician. J Psychiatr Res 12, 189-198.

Forster, M.R., 2002. Predictive Accuracy as an Achievable Goal of Science. Philosophy of Science 69, S124-S134.

Friman, O., Borga, M., Lundberg, P., Knutsson, H., 2004. Detection and detrending in fMRI data analysis. Neuroimage 22, 645-655.

Friston, K., Chu, C., Mourao-Miranda, J., Hulme, O., Rees, G., Penny, W., Ashburner, J., 2008. Bayesian decoding of brain images. Neuroimage 39, 181-205.

Friston, K., Mattout, J., Trujillo-Barreto, N., Ashburner, J., Penny, W., 2007a. Variational free energy and the Laplace approximation. Neuroimage 34, 220-234.

Friston, K., Mattout, J., Trujillo-Barreto, N., Ashburner, J., Penny, W., 2007b. Variational free energy and the Laplace approximation. Neuroimage 34, 220-234.

Friston, K.J., Ashburner, J., Kiebel, J.S., Nichols, T.E., W., W.D., 2007c. Statistical parametric mapping, the analysis of functional brain images. Academic press.

Friston, K.J., Holmes, A.P., Worsley, K.J., Poline, J.P., Frith, C.D., Frackowiak, R.S.J.,

1995. Statistical Parametric Maps in Functional Imaging: A General Linear Approach. Hum Brain Mapp 2, 189-210.

Friston, K.J., Penny, W., Phillips, C., Kiebel, S., Hinton, G., Ashburner, J., 2002. Classical and Bayesian inference in neuroimaging: theory. Neuroimage 16, 465-483.

Fu, C.H.Y., Mourao-Miranda, J., Costafreda, S.G., Khanna, A., Marquand, A.F., Williams, S.C.R., Brammer, M.J., 2008. Pattern classification of sad facial processing: Toward the development of neurobiological markers in depression. Biological Psychiatry 63, 656-662.

Fuller, R.W., Bromer, W.W., Snoddy, H.D., Baker, J.C., 1975. Regulation of enzyme activity by glucagon: increased hormonal activity of iodinated glucagon. Adv Enzyme Regul 13, 201-215.

Galasko, D.R., Gould, R.L., Abramson, I.S., Salmon, D.P., 2000. Measuring cognitive change in a cohort of patients with Alzheimer's disease. Stat Med 19, 1421-1432.

Ghahramani, Z., Sahani, M., 2005. Unsupervised Learning 2005 Course Web Page.http://www.gatsby.ucl.ac.uk/~zoubin/course05/

Gigerenzer, G., 2002. Adaptive Thinking: Rationality in the Real World. Oxford University Press.

Golland, P., Fischl, B., Spiridon, M., Kanwisher, N., Buckner, R.L., Shenton, M.E., Kikinis, R., Dale, A., Grimson, W.E.L., 2002. Discriminative analysis for image-based studies. Lecture Notes in Computer Science, 508-515.

Good, C.D., Ashburner, J., Frackowiak, R.S., 2001a. Computational neuroanatomy: new perspectives for neuroradiology. Rev Neurol (Paris) 157, 797-806.

Good, C.D., Johnsrude, I.S., Ashburner, J., Henson, R.N., Friston, K.J., Frackowiak, R.S., 2001b. A voxel-based morphometric study of ageing in 465 normal adult human brains. Neuroimage 14, 21-36.

Gopikrishnan, P., Meyer, M., Amaral, L.A.N., Stanley, H.E., 1998. Inverse cubic law for the distribution of stock price variations. The European Physical Journal B 3, 139-140.

Grootoonk, S., Hutton, C., Ashburner, J., Howseman, A.M., Josephs, O., Rees, G., Friston, K.J., Turner, R., 2000. Characterization and correction of interpolation effects in the realignment of fMRI time series. Neuroimage 11, 49-57.

Guyon, I., Elisseeff, A.e., 2003. An Introduction to Variable and Feature Selection. Journal of Machine Learning Research 3, 1157-1182.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine learning 46, 389-422.

Hamilton, M., 1960. A rating scale for depression. British Medical Journal 23, 56.

Hardoon, D.R., Mourao-Miranda, J., Brammer, M., Shawe-Taylor, J., 2007. Unsupervised analysis of fMRI data using kernel canonical correlation. Neuroimage 37, 1250-1259.

Harville, D.A., 1977. Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems. Journal of the American Statistical Association 72, 320-338.

Hassabis, D., Chu, C., Rees, G., Weiskopf, N., Molyneux, P.D., Maguire, E.A., 2009. Decoding Neuronal Ensembles in the Human Hippocampus. Curr Biol.

Hastie, T., Tibshirani, R., Friedman, J.H., 2003. The Elements of Statistical Learning. Springer.

Haxby, J.V., Gobbini, M.I., Furey, M.L., Ishai, A., Schouten, J.L., Pietrini, P., 2001. Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science 293, 2425-2430.

Haynes, J.D., Rees, G., 2005. Predicting the stream of consciousness from activity in human visual cortex. Curr Biol 15, 1301-1307.

Haynes, J.D., Rees, G., 2006. Decoding mental states from brain activity in humans. Nat Rev Neurosci 7, 523-534.

Haynes, J.D., Sakai, K., Rees, G., Gilbert, S., Frith, C., Passingham, R.E., 2007. Reading hidden intentions in the human brain. Curr Biol 17, 323-328.

Henderson, C.R., 1953. Estimation of Variance and Covariance Components. Biometrics 9, 226-252.

Henson, R., 2004. Chapter 10: Analysis of fMRI Timeseries. Human Brain Function 2nd Edition.

Hirata, Y., Matsuda, H., Nemoto, K., Ohnishi, T., Hirao, K., Yamashita, F., Asada, T., Iwabuchi, S., Samejima, H., 2005. Voxel-based morphometry to discriminate early Alzheimer's disease from controls. Neurosci Lett 382, 269-274.

Hsiang, T.C., 1975. A bayesian view on ridge regression. The statistician 24, 267-268.

Hsu, C.W., Chang, C.C., Lin, C.J., 2003. A practical guide to support vector classification.

Huang, J., Ling, C.X., 2005. Using AUC and Accuracy in Evaluation learning algorithm. IEEE Transactions on Knowledge and Data Engineering 17, 299 - 310

Hunt, E., 2003. Book Review: Adaptive Thinking: Rationality in the Real World. Evolutionary Psychology 1, 172-187.

Jaakkola, T.S., Haussler, D., 1999. Exploiting generative models in discriminative classifiers. Advances in Neural Information Processing Systems, 487-493.

Jack, C.R., Dickson, D.W., Parisi, J.E., Xu, Y.C., Cha, R.H., O'Brien, P.C., Edland, S.D., Smith, G.E., Boeve, B.F., Tangalos, E.G., 2002. Antemortem MRI findings correlate with hippocampal neuropathology in typical aging and dementia. Neurology 58, 750-757.

Jack, C.R., Jr., Lowe, V.J., Senjem, M.L., Weigand, S.D., Kemp, B.J., Shiung, M.M., Knopman, D.S., Boeve, B.F., Klunk, W.E., Mathis, C.A., Petersen, R.C., 2008. 11C PiB and structural MRI provide complementary information in imaging of Alzheimer's disease and amnestic mild cognitive impairment. Brain 131, 665-680.

Jolliffe, I.T., 1982. A note on the Use of Principal Components in Regression. Journal of the Royal Statistical Society, Series C (Applied Statistics) 31, 300-303.

Jolliffe, I.T., 2002. Principal Component Analysis, 2nd ed. Springer.

Kassubek, J., Juengling, F.D., Kioschies, T., Henkel, K., Karitzky, J., Kramer, B., Ecker, D., Andrich, J., Saft, C., Kraus, P., 2004. Topography of cerebral atrophy in early Huntington's disease: a voxel based morphometric MRI study. British Medical Journal 75, 213-220.

Kloppel, S., Chu, C., Tan, G.C., Draganski, B., Johnson, H., Paulsen, J.S., Kienzle, W., Tabrizi, S.J., Ashburner, J., Frackowiak, R.S., 2009. Automatic detection of preclinical neurodegeneration: presymptomatic Huntington disease. Neurology 72, 426-431.

Kloppel, S., Draganski, B., Golding, C.V., Chu, C., Nagy, Z., Cook, P.A., Hicks, S.L., Kennard, C., Alexander, D.C., Parker, G.J., Tabrizi, S.J., Frackowiak, R.S., 2008a. White matter connections reflect changes in voluntary-guided saccades in pre-symptomatic Huntington's disease. Brain 131, 196-204.

Kloppel, S., Stonnington, C.M., Barnes, J., Chen, F., Chu, C., Good, C.D., Mader, I., Anne Mitchell, L., Patel, A.C., Roberts, C.C., Fox, N.C., Jack, C.R., Jr., Ashburner, J., Frackowiak, R.S., 2008b. Accuracy of dementia diagnosis--a direct comparison between radiologists and a computerized method. Brain.

Kloppel, S., Stonnington, C.M., Chu, C., Draganski, B., Scahill, R.I., Rohrer, J.D., Fox, N.C., Ashburner, J., Frackowiak, R.S., 2008c. A plea for confidence intervals and consideration of generalizability in diagnostic studies. Brain.

Kloppel, S., Stonnington, C.M., Chu, C., Draganski, B., Scahill, R.I., Rohrer, J.D., Fox, N.C., Jack, C.R., Jr., Ashburner, J., Frackowiak, R.S., 2008d. Automatic classification of MR scans in Alzheimer's disease. Brain 131, 681-689.

Kriegeskorte, N., Goebel, R., Bandettini, P., 2006. Information-based functional brain mapping. Proceedings of the National Academy of Sciences 103, 3863-3868.

Kriegeskorte, N., Mur, M., Ruff, D.A., Kiani, R., Bodurka, J., Esteky, H., Tanaka, K., Bandettini, P.A., 2008. Matching categorical object representations in inferior temporal cortex of man and monkey. Neuron 60, 1126-1141.

Kurucz, M., Benczúr, A.A., Csalogány, K., 2007 Methods for large scale SVD with missing values. Proc. KDD Cup and Workshop 2007 in conjunction with KDD 2007.

LaConte, S., Anderson, J., Muley, S., Ashe, J., Frutiger, S., Rehm, K., Hansen, L.K.,

Yacoub, E., Hu, X., Rottenberg, D., Strother, S., 2003. The evaluation of preprocessing choices in single-subject BOLD fMRI using NPAIRS performance metrics. Neuroimage 18, 10-27.

LaConte, S., Strother, S., Cherkassky, V., Anderson, J., Hu, X., 2005. Support vector machines for temporal classification of block design fMRI data. Neuroimage 26, 317-329.

Lancaster, J.L., Tordesillas-Gutierrez, D., Martinez, M., Salinas, F., Evans, A., Zilles, K., Mazziotta, J.C., Fox, P.T., 2007. Bias between MNI and Talairach coordinates analyzed using the ICBM-152 brain template. Human Brain Mapping 28.

Langbehn, D.R., Brinkman, R.R., Falush, D., Paulsen, J.S., 2004. A new model for prediction of the age of onset and penetrance for Huntington's disease based on CAG length. Clinical genetics 65, 267-277.

Lao, Z., Shen, D., Xue, Z., Karacali, B., Resnick, S.M., Davatzikos, C., 2004. Morphological classification of brains via high-dimensional shape transformations and machine learning methods. Neuroimage 21, 46-57.

Lay, D.C., 1997. Linear algebra and its applications, 2nd ed. Addison-Wesley.

Lee, Y., Lee, K.Y., Lee, J., 2006. The Estimating Optimal Number of Gaussian Mixtures Based on Incremental k-means for Speaker Identification. International Journal of Information Technology 12.

Li, J., 2008. STAT597E/IST597E/CSE598E: Data Mining Course Material http://www.stat.psu.edu/~jiali/course/stat597e/notes2/lda.pdf

Logothetis, N.K., 2008. What we can do and what we cannot do with fMRI. Nature 453, 869-878.

Logothetis, N.K., Pauls, J., Augath, M., Trinath, T., Oeltermann, A., 2001. Neurophysiological investigation of the basis of the fMRI signal. Nature 412, 150-157.

Lyoo, I.K., Kim, M.J., Stoll, A.L., Demopulos, C.M., Parow, A.M., Dager, S.R., Friedman, S.D., Dunner, D.L., Renshaw, P.F., 2004. Frontal lobe gray matter density

decreases in bipolar I disorder. Biological Psychiatry 55, 648-651.

Ma, J., Miller, M.I., Trouve, A., Younes, L., 2008. Bayesian template estimation in computational anatomy. Neuroimage 42, 252-261.

Mackay, D.J.C., 1992. The evidence framework applied to classification networks. Neural Computation 4, 720-736.

MacKay, D.J.C., 1995. Probable networks and plausible predictions a review of practical Bayesian methods for supervised neural networks. Network: Computation in Neural Systems 6, 469-505.

MacKay, D.J.C., 1998. Introduction to Gaussian Processes. Neural Networks and Machine Learning 168, 133-165.

MacKay, D.J.C., 2002. Information Theory, Inference & Learning Algorithms, 1st ed. Cambridge University Press.

Magnus, J.R., Neudecker, H., 1999. Matrix Differential Calculus with Applications in Statistics and Econometrics. John Wiley.

Maguire, E.A., Gadian, D.G., Johnsrude, I.S., Good, C.D., Ashburner, J., Frackowiak, R.S.J., Frith, C.D., 2000. Navigation-related structural change in the hippocampi of taxi drivers. National Acad Sciences, pp. 4398-4403.

Malonek, D., Grinvald, A., 1996. Interactions between electrical activity and cortical microcirculation revealed by imaging spectroscopy: implications for functional brain mapping. Science 272, 551-554.

Mattis, S., 1988. Dementia Rating Scale: Professional Manual. Odessa, FL: Psychological Assessment Resources. Inc.

McKenzie, P., Alder, M., 1994. Selecting the optimal number of components for a Gaussian mixturemodel. Information Theory, 1994. Proceedings., 1994 IEEE International Symposium on, 393.

McRobbie, D.W., Moore, E.A., Graves, M.J., Prince, M.R., 2007. Mri From Picture To Proton, 2nd ed. Cambridge University Press.

Miller, M.I., 2004. Computational anatomy: shape, growth, and atrophy comparison via diffeomorphisms. Neuroimage 23 Suppl 1, S19-33.

Minka, T., 2001. A Family of Algorithms for Approximate Bayesian Inference. Massachusetts Institute of Technology.

Moler, C., 2006. Professor SVD. Newsletters - The MathWorks News & Notes. The MathWorks.

Morch, N., Hansen, L.K., Strother, S.C., Svarer, C., Rottenberg, D.A., Lautrup, B., Savoy, R., Paulson, O.B., 1997. Nonlinear versus linear models in functional neuroimaging: Learning curves and generalization crossover. Lecture Notes in Computer Science 1230, 259-270.

Mortimer, J.A., Borenstein, A.R., Gosche, K.M., Snowdon, D.A., 2005. Very early detection of Alzheimer neuropathology and the role of brain reserve in modifying its clinical expression. J Geriatr Psychiatry Neurol 18, 218-223.

Moser, E.I., Kropff, E., Moser, M.B., 2008. Place cells, grid cells, and the brain's spatial representation system. Annu Rev Neurosci 31, 69-89.

Mourao-Miranda, J., Bokde, A.L., Born, C., Hampel, H., Stetter, M., 2005. Classifying brain states and determining the discriminating activation patterns: Support Vector Machine on functional MRI data. Neuroimage 28, 980-995.

Mourao-Miranda, J., Friston, K.J., Brammer, M., 2007. Dynamic discrimination analysis: a spatial-temporal SVM. Neuroimage 36, 88-99.

Mourao-Miranda, J., Reynaud, E., McGlone, F., Calvert, G., Brammer, M., 2006. The impact of temporal compression and space selection on SVM analysis of single-subject and multi-subject fMRI data. Neuroimage 33, 1055-1065.

Mueller, S.G., Weiner, M.W., Thal, L.J., Petersen, R.C., Jack, C.R., Jagust, W., Trojanowski, J.Q., Toga, A.W., Beckett, L., 2005. Ways toward an early diagnosis in Alzheimer's disease: The Alzheimer's Disease Neuroimaging Initiative (ADNI). Alzheimer's & dementia: the journal of the Alzheimer's Association 1, 55-66.

Ni, Y., Chu, C., Saunders, C.J., Ashburner, J., 2008. Kernel methods for fMRI pattern

prediction. Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, Hong Kong, pp. 692-697.

Nichols, T.E., Holmes, A.P., 2002. Nonparametric permutation tests for functional neuroimaging: a primer with examples. Human Brain Mapping 15, 1-25.

Noble, W.S., 2006. What is a support vector machine? Nat Biotechnol 24, 1565-1567.

Paulsen, J., Hayden, M., Stout, J., Langbehn, D., Aylward, E., Ross, C., Guttman, M., Nance, M., Kieburtz, K., Oakes, D., Shoulson, I., Kayson, E., Johnson, S., Penziner, E., 2006. Preparing for preventive clinical trials: the Predict-HD study. Archives of Neurology 63, 883-890.

Pekalska, E., Duin, R.P.W., 2005. The Dissimilarity Representation for Pattern Recognition: Foundations And Applications World Scientific Publishing Company.

Peng, J.Y., Aston, J.A.D., Gunn, R.N., Liou, C.Y., Ashburner, J., 2008. Dynamic Positron Emission Tomography Data-Driven Analysis Using Sparse Bayesian Learning. IEEE Transactions on Medical Imaging 27, 1356-1369.

Penney Jr, J.B., Vonsattel, J.P., Macdonald, M.E., Gusella, J.F., Myers, R.H., 1997. CAG repeat number governs the development rate of pathology in Huntington's disease. Annals of Neurology 41.

Perneczky, R., Wagenpfeil, S., Komossa, K., Grimmer, T., Diehl, J., Kurz, A., 2006. Mapping scores onto stages: mini-mental state examination and clinical dementia rating. Am J Geriatr Psychiatry 14, 139-144.

Petersen, R.C., Kokmen, E., Tangalos, E., Ivnik, R.J., Kurland, L.T., 1990. Mayo Clinic Alzheimer's Disease Patient Registry. Aging (Milano) 2, 408-415.

Petersen, R.C., Stevens, J.C., Ganguli, M., Tangalos, E.G., Cummings, J.L., DeKosky, S.T., 2001. Practice parameter: early detection of dementia: mild cognitive impairment (an evidence-based review) Report of the Quality Standards Subcommittee of the American Academy of Neurology. Neurology 56, 1133-1142.

Platt, J., 1999a. Sequential minimal optimization: A fast algorithm for training support

vector machines. Advances in Kernel Methods-Support Vector Learning 208.

Platt, J.C., 1999b. Probabilities for SV machines. Advances in Neural Information Processing Systems, 61-74.

Pohar, M., Blas, M., Turk, S., 2004. Comparison of Logistic Regression and Linear Discriminant Analysis: A Simulation Study. Metodološki zvezki 1, 143-161.

Pontil, M., Verri, A., 1998. Properties of support vector machines. Neural Computation 10, 955-974.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical recipes in C: the art of scientific computation. Cambridge U. Press, Cambridge, Mass.

Qiu, A., Younes, L., Wang, L., Ratnanather, J.T., Gillepsie, S.K., Kaplan, G., Csernansky, J., Miller, M.I., 2007. Combining anatomical manifold information via diffeomorphic metric mappings for studying cortical thinning of the cingulate gyrus in schizophrenia. Neuroimage 37, 821-833.

Rakotomamonjy, A., 2003. Variable selection using svm based criteria. The Journal of Machine Learning Research 3, 1357-1370.

Rasmussen, C.E., Quiñonero-Candela, J., 2005. Healing the relevance vector machine through augmentation.

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning The MIT Press

Ray, S., Britschgi, M., Herbert, C., Takeda-Uchimura, Y., Boxer, A., Blennow, K., Friedman, L.F., Galasko, D.R., Jutel, M., Karydas, A., Kaye, J.A., Leszek, J., Miller, B.L., Minthon, L., Quinn, J.F., Rabinovici, G.D., Robinson, W.H., Sabbagh, M.N., So, Y.T., Sparks, D.L., Tabaton, M., Tinklenberg, J., Yesavage, J.A., Tibshirani, R., Wyss-Coray, T., 2007. Classification and prediction of clinical Alzheimer's diagnosis based on plasma signaling proteins. Nat Med 13, 1359-1362.

Rey, A., 1964. L'examen clinique en psychologie [The clinical examination in psychology]. Paris: Presses Universitaires de France.

Ritchie, M.D., White, B.C., Parker, J.S., Hahn, L.W., Moore, J.H., 2003. Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases. BMC Bioinformatics 4, 28.

Rosenblatt, F., 1962. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books.

Sackett, D.L., 1997. Evidence-based medicine. Elsevier, pp. 3-5.

Sato, J.R., da Graca Morais Martin, M., Fujita, A., Mourao-Miranda, J., Brammer, M.J., Amaro Jr, E., 2008a. An fMRI normative database for connectivity networks using one-class support vector machines. Human Brain Mapping.

Sato, J.R., Thomaz, C.E., Cardoso, E.F., Fujita, A., Martin Mda, G., Amaro, E., Jr., 2008b. Hyperplane navigation: a method to set individual scores in fMRI group datasets. Neuroimage 42, 1473-1480.

Schacter, D.L., Buckner, R.L., Koutstaal, W., Dale, A.M., Rosen, B.R., 1997. Late onset of anterior prefrontal activity during true and false recognition: An event-related fMRI study. Neuroimage 6, 259-269.

Schlkopf, B., Smola, A.J., 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press.

Scholkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., 2001. Estimating the support of a high-dimensional distribution. Neural Computation 13, 1443-1471.

Scholkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L., 2000. New support vector algorithms. Neural Computation 12, 1207-1245.

Shawe-Taylor, J., Cristianini, N., 2004. Kernel Methods for Pattern Analysis. Cambridge University Press.

Shinkareva, S.V., Mason, R.A., Malave, V.L., Wang, W., Mitchell, T.M., Just, M.A., 2008. Using FMRI brain activation to identify cognitive States associated with perception of tools and dwellings. PLoS ONE 3, e1394.

Smith, A.M., Lewis, B.K., Ruttimann, U.E., Ye, F.Q., Sinnwell, T.M., Yang, Y., Duyn, J.H., Frank, J.A., 1999. Investigation of low frequency drift in fMRI signal. Neuroimage 9, 526-533.

Smola, A.J., Olkopf, B.S., 2003. A tutorial on support vector regression. Statistics and Computing.

Solomon, P.R., Murphy, C.A., 2005. Should we screen for Alzheimer's disease? Geriatrics 60, 26-31.

Spitzer, R.L., 1994. DSM-IV casebook: a learning companion to the diagnostic and statistical manual of mental disorders. American Psychiatric Publishing, Inc.

Stonnington, C.M., Tan, G., Kloppel, S., Chu, C., Draganski, B., Jack, C.R., Jr., Chen, K., Ashburner, J., Frackowiak, R.S., 2008. Interpreting scan data acquired from multiple scanners: a study with Alzheimer's disease. Neuroimage 39, 1180-1185.

Strother, S.C., 2006. Evaluating fMRI preprocessing pipelines. IEEE Eng Med Biol Mag 25, 27-41.

Su, E.C., Chiu, H.S., Lo, A., Hwang, J.K., Sung, T.Y., Hsu, W.L., 2007. Protein subcellular localization prediction based on compartment-specific features and structure conservation. BMC Bioinformatics 8, 330.

Tanabe, J., Miller, D., Tregellas, J., Freedman, R., Meyer, F.G., 2002. Comparison of detrending methods for optimal fMRI preprocessing. Neuroimage 15, 902-907.

Tarantola, A., 2004. Inverse Problem Theory. Society for Industrial and Applied Mathematics.

Tax, D., 2001. One-class classification. Unpublished doctoral/dissertation, Delft University of Technology.

Thieben, M.J., Duggins, A.J., Good, C.D., Gomes, L., Mahant, N., Richards, F., McCusker, E., Frackowiak, R.S.J., 2002. The distribution of structural neuropathology in pre-clinical Huntington's disease. Brain 125, 1815-1828.

Thomaz, C.E., Gillies, D.F., 2005. Using a maximum uncertainty lda-based approach

to classify and analyse mr brain images. In Proc. MICCA'04 LNCS 3216, 89-96.

Tipping, M.E., 2000. The Relevance Vector Machine. Advances in Neural Information Processing Systems 12.

Tipping, M.E., 2001. Sparse bayesian learning and the relevance vector machine. Journal of Machine Learning Research 1, 211-244.

Twamley, E.W., Ropacki, S.A., Bondi, M.W., 2006. Neuropsychological and neuroimaging changes in preclinical Alzheimer's disease. J Int Neuropsychol Soc 12, 707-735.

Ulusoy, I., Bishop, C.M., 2005a. Comparison of generative and discriminative techniques for object detection and classification.

Ulusoy, I., Bishop, C.M., 2005b. Generative versus discriminative models for object recognition.

Vapnik, V., 1995. The nature of statistical learning theory. NY Springer.

Vapnik, V.N., 1998. Statistical Learning Theory. Wiley.

Vemuri, P., Gunter, J.L., Senjem, M.L., Whitwell, J.L., Kantarci, K., Knopman, D.S., Boeve, B.F., Petersen, R.C., Jack, C.R., 2008. Alzheimer's disease diagnosis in individual subjects using structural MR images: Validation studies. Neuroimage 39, 1186-1197.

Wahlund, L.O., Almkvist, O., Blennow, K., Engedahl, K., Johansson, A., Waldemar, G., Wolf, H., 2005. Evidence-based evaluation of magnetic resonance imaging as a diagnostic tool in dementia workup. Topics in Magnetic Resonance Imaging 16, 427.

Wang, L., Beg, M., Ratnanather, J., Ceritoglu, C., Younes, L., Morris, J.C., Csernansky, J.G., Miller, M.I., 2007. Large deformation diffeomorphism and momentum based hippocampal shape discrimination in dementia of the Alzheimer type. IEEE Transactions on Medical Imaging 26, 462-470.

Watanabe, S., 1970. Knowing and guessing, A quantitative study of inference and information. Journal of Information Theory 16, 361-362.

Williams, C.K.I., 1999. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. Learning in Graphical Models, 599-621.

Williams, C.K.I., Barber, D., 1998. Bayesian classification with Gaussian processes. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 1342-1351.

Zoubir, A.M., Boashash, B., 1998. The bootstrap and its application in signal processing. IEEE signal processing magazine 15, 56-76.