Classifier Ensembles for Streaming fMRI Data

PRIFYSGOL
BANGOR
UNIVERSITY

Catrin Oliver Plumpton

Bangor University

A thesis submitted for the degree of

Doctor of Philosophy

2011

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed: .... (Catrin O. Plumpton)

Date:

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed: .... (Catrin O. Plumpton)

Date:

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed: .... (Catrin O. Plumpton)

Date:

Acknowledgements

> In the study of the brain functions we rely upon a biased, poorly understood, and frequently unpredictable organ in order to study the properties of another such organ; we have to use a brain to study a brain. $\sim$ William C. Coning (from The Mind: Biological Approaches To Its Functions, 1968).

After nearly two years of working on the brain I can confidently say that I understand a little more than I did when I started. I am equally confident in saying that during this time I have been difficult to understand and frequently unpredictable. I would like to thank those who joined me on the journey, especially those who have lifted my spirits and guided me along the way. It has been no easy task, and without those who have stood by me throughout the last few years, this piece of work would almost certainly not exist. There are too many to thank everyone individually, however a few deserve a special mention. First and foremost, I would like to thank Professor Ludmila (Lucy) Kuncheva. Without her support and belief in me, this project would have never even begun, so Lucy, my heartfelt thanks go to you.

Thanks should also go to my parents for putting up with me and for convincing me to carry on when times were tough. A special mention also to Hannah and Nick Fleck, Rachael Hobbs, Richard Williams and Emma Prout for their unwavering support. I would like to thank my office mates over the years; Tom, Haider, James, David (Mez) and Rhys, and the members of BUCP, Salsa Bangor and Tango Bangor for providing hours of distractions! Last of all, I would like to thank Beau, an mostly tolerant (and very tall) horse, with an incredible knack of lifting me, or bringing me back down to earth (quite literally) as and when required.

Abstract

Functional Magnetic Resonance Imaging (fMRI) is an exciting technology which allows neuroscientists to gather data on activity within the brain. This activity corresponds to neural processes such as emotion or motor activities. By applying machine learning techniques to fMRI data, the patterns corresponding to these processes can be recognised and classified.

The ability to classify neural processes opens up a wealth of opportunities to neuroscientists. Early fMRI experiments focus on identifying regions of the brain involved in processes such as pain or emotion. Having identified these regions, it is possible to see how they react to stimuli differently in participants with different conditions, for example depression, autism or attachment disorder.

Most of this work is exploratory in nature, with analysis being carried out offline, that is, once the fMRI data collection is complete. More recent advances in classification speed and accuracy, and in fMRI technology, have allowed for real time experiments. During real time fMRI experiments the classifier is trained and then used during the course of the experiment. In what is termed a neurofeedback loop, the stimuli presented to the participant can be updated or altered dependent upon the classification result of the output data. Real time fMRI has been used in many proof of concept type experiments, such as navigating mazes or balancing a pendulum by using different thought processes.

In order to better facilitate real time fMRI classification, it is proposed here that an online classifier will be advantageous. During the course of a real time fMRI experiment, training data is often very limited, therefore the ability of a classifier to learn from new data during the course of the experiment will be of benefit. In order to maintain speed and accuracy, we propose a random subspace ensemble of linear classifiers.

Further to this, it is noted that in many cases, during the online phase, true class labels may not be known. The use of an online ‘naive labelling’ classifier within an ensemble framework is proposed as an alternative to a fixed pre-trained classifier. This is extended by the introduction of a ‘guided update’ strategy for the ensemble, whereby the classifiers within the ensemble are updated using the ensemble decision, rather than the individual decisions. Comparison of this strategy with a fixed classifier ensemble and an ensemble of classifiers with individual ‘naive’ updates is provided. Variations of the guided update strategy are also proposed, whereby classifiers within the ensemble are only updated when specific criteria are fulfilled. These criteria are based upon agreement with the ensemble decision, and confidence in the ensemble decision.

The proposed methods are shown to provide more accurate results than using a fixed classifier, and are tested across a variety of emotion based fMRI data sets.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The use of functional Magnetic Resonance Imaging (fMRI) as a tool for investigation of processes within the brain. Until recently, analysis of fMRI data has been univariate, considering data from each voxel in the scan independently. The activation patterns associated with brain processes however, are highly distributed throughout the brain. Univariate approaches are unable to capture and make use of this distribution. In order to improve the tools available to the psychologists and neuroscientists, multivariate classification techniques are now being used for fMRI data.

A move towards neurofeedback experiments, conducted in real-time, has meant that data handling and classification need to be fast, allowing the psychologist or neuroscientist to update the stimuli according to the results.

Due to the expense and time taken to organise and conduct fMRI experiments, it is advantageous to be able to complete an experiment in a single trial. Typically, classifiers will be trained on data from one trial, and then used for classification in subsequent trials. This work seeks to offer a classifier which can learn from a small batch of training data at the start of a trial, and then continue to learn and update throughout the course of the trial. There are many fields relating to this thesis, an introduction is offered to each area, however only those methods used in the experiments are explained in detail. Data was kindly supplied by psychologists from the School of Psychology, Bangor University.

1.2 Research Hypothesis

The human brain is a complex architecture, fascinating to both scientists and non-scientists alike. Lately, the powers of fMRI have received much media attention. Experiments hitting the headlines have included discovery of awareness in patients in a vegetative state [92], and 'love can ease pain' [134].

Multivariate analysis of fMRI has allowed classification of participants' 'brain states' when they are subjected to different stimuli. Advances in classifier and fMRI technology have meant that this analysis can now happen in real time.

fMRI data is complex in nature, and suffers from a very high feature-to-instance ratio, typically, with very few training examples available. Linear classifiers, in particular the support vector machine with linear kernel, are popular for fMRI analysis due to their speed and accuracy [23, 30, 75, 94].

Classifier ensembles are deemed to be more accurate than individual classifiers, and are less prone to overfitting the data than individual classifiers. This is particularly beneficial in data sets with a large feature-to-instance ratio. The Random Subspace (RS) ensemble framework [52] has been shown to be accurate for data sets displaying such properties; specifically when there is high redundancy in the feature set [116]. Experiments with the RS ensemble on fMRI data have shown promising results, [71], which this thesis aims to build upon.

Here it is hypothesised that a classifier for fMRI data would benefit from continuing to learn throughout the course of an experiment. An online classifier is capable of updating after each new instance is presented. This means that the classifier is able to continue learning beyond the initial training phase.

It is also noted that there may be cases when the true ‘label’ of the participant’s brain state may not be known. The use of naive labelling is proposed as a possible solution [73]. The drawback is the possibility of a ‘runaway’ classifier, where the uncorrected classifier progressively learns ‘the wrong thing’ [26]. By using naive labelling for online learning within a classifier ensemble, here, it is hypothesised that not only will the classifier continue to learn throughout the course of the experiment, but that the ensemble environment will counteract any runaway behaviours introduced by naive labelling.

To test the hypothesis, it is proposed to use the ensemble decision, rather than the individual classifier decisions for the updates of the online classifiers. As this approach may compromise the diversity of the ensemble, this work is concluded by offering different criteria which may be used to determine when to update the ensemble members.

1.3 Outline of Tasks

In order to move towards real-time classification of potentially unlabelled fMRI data, the overall task is broken down into a series of challenges.

1. To introduce and consider some of the challenges of analysing fMRI data.

2. To introduce the techniques which will be applied to fMRI analysis in this thesis, namely linear classifiers, online classification and semi-supervised learning.

3. To introduce classifier ensembles, specifically the random subspace ensemble. Parameters are derived for the random subspace ensemble for application to fMRI data.

4. The naive labelling strategy for semi-supervised classification of i.i.d fMRI data.

5. Supervised classification of non-i.i.d. fMRI data.

6. Exploration of update strategies for using naive labelling with classification of non-i.i.d. fMRI data.

1.4 Contributions

This thesis offers the following contributions:

1. Guidelines to parameter selection for the random subspace ensemble. As part of a collaboratory work recommendations for the parameters of the ensemble based upon criteria of usability, feature set diversity and coverage were derived. These values were tested on both synthetic and real fMRI data.

2. The use of naive labelling within an ensemble framework in order to reduce the likelihood of a runaway classifier. The online naive labelling strategy is compared with a fixed pre-trained classifier, and a classifier with supervised online updates. Both individual classifiers and random subspace ensembles are considered. The methods are tested on i.i.d. (shuffled) fMRI data.

3. The use of a random subspace ensemble of online linear classifiers for streaming fMRI data. Three online classifier models were compared, both as individual classifiers and as part of a random subspace ensemble.

4. The guided update strategy for a classifier ensemble. Two update strategies for the random subspace ensemble were compared with a fixed pre-trained classifier for streaming fMRI data.

5. Criteria upon which to update the guided ensemble. Three update criteria are introduced for the guided ensemble. Ensembles updated using these criteria were compared with the naive ensemble and guided ensemble.

Chapter 2

fMRI Data and Analysis

2.1 Brain and Behaviour

Human behaviour is controlled by the brain, and can be separated into processes including motor control, vision and emotion. Different regions of the brain are responsible for different processes, with some processes involving more than one region of the brain.

The brain can be categorised into three areas, the forebrain, the midbrain and the hindbrain [18]. The midbrain and the hindbrain are mainly concerned with primary support functions such as respiration and control of the pulse. The forebrain is responsible for the majority of high level functions such as memory and language. The forebrain is the area of the brain where conscious processes such as decision making takes place.

The forebrain is further sub-divided into four regions, known as lobes, illustrated in Figure 2.1. The function of the lobes is described below [39]:

The frontal lobe, is located at the front of the brain, and is where conscious decisions are made. It is involved in regulating behaviour and handling the planning and control of movement, and can be thought of as our management centre.

The parietal lobe, is located in the upper-rear part of the brain. This lobe is involved in processing information about sensations such as temperature, pain

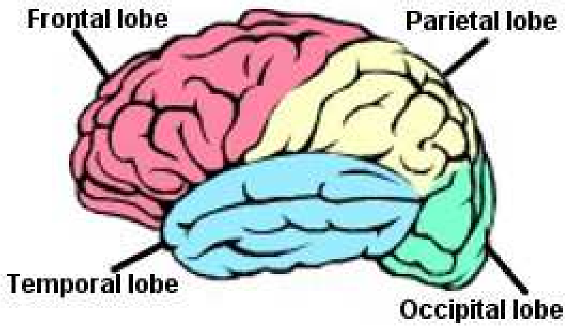


Figure 2.1: The location of the lobes in the forebrain.

or pressure.

The temporal lobe, located beneath the parietal lobe, is the region of the brain concerned with memory and language processes. Auditory and speech perception are processed in the temporal lobe.

The occipital lobe, is located at the rear of the brain and plays a key role in processing visual information.

Perhaps the most important part of the forebrain is the cerebral cortex. The cerebral cortex forms a thin surface layer on the outside of the forebrain. The layer is tightly folded in order to give a high surface area. Also referred to as the grey matter, it contains approximately ten billion brain cells. Below this surface layer lie bundles of myelin nerve fibres, known as the white matter. These nerve fibres transport information around the cortex and to other regions of the brain.

One goal of neuroscientists is to ascertain regions of the brain which are responsible for controlling specific processes. Much research has focused on emotion. Detecting and recognising emotion is the main focus of the studies in this thesis. Emotion has been found to be regulated by both the amygdala and the insular cortex.

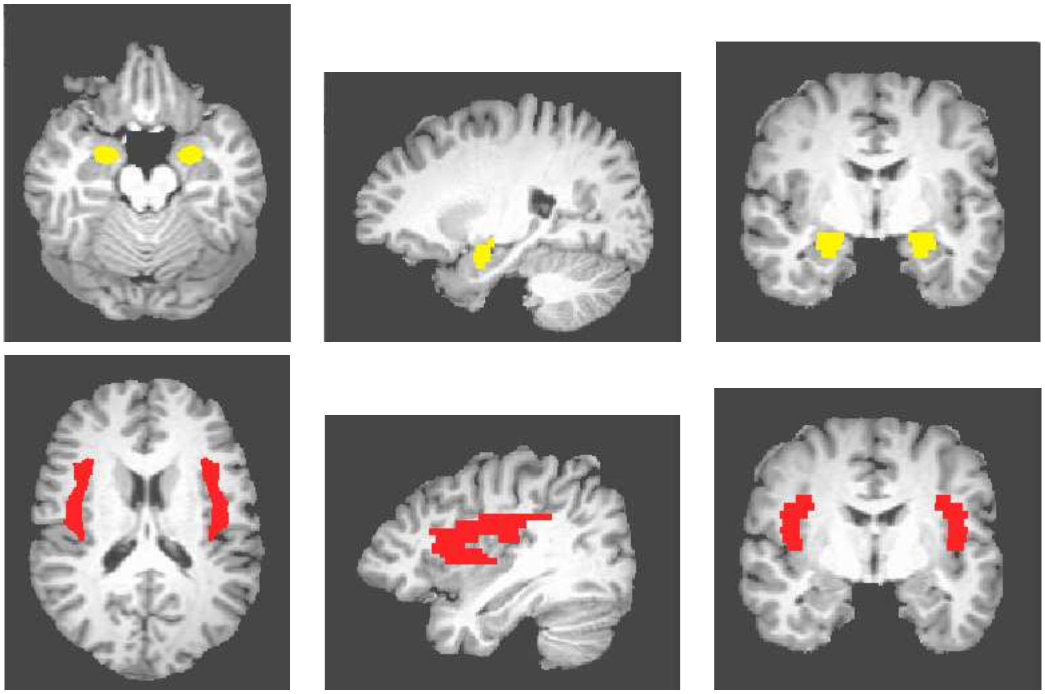


Figure 2.2: Top: Location of the amygdala, highlighted in yellow.
Bottom: Location of the insular cortex, highlighted in red.
Images created using the standard Talairach atlas in AFNI [24], images courtesy of N. N. Oosterhof of Bangor University.

The amygdala is located within the temporal lobes, as can be seen in Figure 2.2 (top). Experiments have shown different levels of activation in the amygdala across participants with certain psychological disorders such as borderline personality disorder, autism and depression. The emotional response of the amygdala has been used by scientists to distinguish between groups of people experiencing these conditions, and control groups [4, 88, 114].

The insular cortex is a portion of the cerebral cortex, as illustrated in Figure 2.2 (bottom). The insular cortex has been linked with cravings such as food or drug cravings [45]. It also has a role in pain, and basic emotions such as anger, fear, happiness or sadness.

2.2 Investigating the Brain

There are many techniques available for investigating the structure and behaviour of the brain. These techniques are categorised as being either invasive or non-invasive. Based upon the nature of the task in hand, one technique may be more appropriate than another. Factors affecting the choice of technique include differences in spatial and temporal resolution, and the ease of data acquisition - some setups are more portable than others. The use of radiation in some techniques also affects their appropriateness in certain circumstances. Six of the more common techniques are described below:

Computerised Axial Tomography (CAT scan) (non-invasive) involves taking a series of x-rays along an axis. These x-rays are stacked to generate a 3-dimensional volume image. Different intensity levels in the image can be used to distinguish between structures in the brain. CAT scans are typically used in diagnostics, for example locating tumours, blood clots or fractures which may occur as a result of a trauma.

Positron Emission Tomography (PET scan) (invasive) involves injecting a radioactive tracer isotope into the body. As the isotope decays, it loses energy until a point where it interacts with an electron. The process results in the release of a pair of photons. These photons are detected by the PET scanner. PET is used diagnostically and can be used to track response to cancer treatments. One advantage of PET over other techniques, is its ability to show how the body is responding to something over time, by tracking the progression of the isotope through the brain, rather than capture a snapshot at a single moment in time. PET has been used for neuroimaging to study the response of the brain to ageing [38] and in subjects with post traumatic stress disorder [102].

Magnetoencephalography (MEG) (non-invasive) records brain activity from the magnetic fields produced by electrical activity in the brain. One advantage of MEG is that it measures brain activity directly, (other techniques measure brain activity by a secondary physiological response). In addition to this, MEG has both excellent spatial and temporal resolution, in the order of a millimetre and milliseconds respectively. The major drawback of MEG is that the magnetic fields are only a few femoteslas in magnitude. A large amount of shielding is therefore required during scanning in order to shield external signals, such as those from the Earth's magnetic field. Neuroscience experiments using MEG include studying the response of the brain during movement [43], and studying the sotomotor cortex (a brain region involved in motor control) in tetraplegic patients [60].

Electroencephalogram (EEG) (non-invasive) is used to record electrical signals within the brain via a series of electrodes placed on the scalp. Spatial resolution of EEG is dependent upon the number of electrodes used, but is typically very low. Another potential drawback of EEG is that the information gathered comes only from the areas of the brain close to the surface of the skull. Deeper neural activity occurring within the folds of the cerebral cortex may not be accounted for.

Advantages of EEG are the high temporal resolution and high tolerance to subject head movement. Another huge benefit is the portability of the technology. Whereas other techniques require large static scanners, EEG experiments can be carried out using a small portable headset or cap. This makes EEG a popular choice for many experiments.

One example of an early experiment with EEG is the comparison of self-reported emotional response in depressed patients and a non-depressed control group [28]. A more recent application sees the use of EEG in automatic seizure detection [61].

Magnetic Resonance Imaging (MRI) (non-invasive) uses magnetic fields to align protons in the body. Once the field is turned off, the protons return to their previous state, releasing photons. These photons produce an electrical signal which is measured by the scanner. MRI like CAT scans, generates scans one slice at a time, with the slices being stacked to form a volume. MRI is typically used for diagnostics, particularly when soft tissue structures, or structures with low density contrast are involved.

Functional Magnetic Resonance Imaging (fMRI) (non-invasive) is used to measure changes in neural activity. Increased neural activity results in a higher demand for oxygen. The magnetic properties of oxygenated blood are different to those of de-oxygenated blood. An increased vascular response is therefore visible as a grey-level intensity change in the scan. This vascular response is known as the Blood Oxygen Level Dependent (BOLD) response.

fMRI is a non-invasive and powerful method for analysis of the operational mechanisms of the brain. The data acquired from fMRI provides a spatially accurate account of neural activity. Analysing the BOLD signal allows scientists to discover how brain states are mapped onto patterns of neural activity, giving insight into the functional architecture of the mind and the processes which control and reflect human behaviour. By taking repeated scans over a period of time, neural activity can be tracked and measured. Whilst the temporal resolution of fMRI is far lower than that of EEG, the spatial resolution is much higher, and it allows the measurement of activity throughout the brain, rather than restricted to the surface.

Many neuroscience experiments focus on fMRI, and it is data gathered using this technology which is considered for the remainder of this thesis.

2.3 Data Acquisition

fMRI data is acquired in a standard MRI scanner. The scanner takes the form of a large cylindrical magnet. The participant lies on a table which is slid inside the cylinder. Stimuli are presented to the participant whilst they are inside the scanner. Stimuli may be images, audio, or sensory (such as heat or pain). The brain is scanned after each presentation and the response is transferred to a server or computer which is attached to the scanner. Scans make up a time series and are collected and analysed once the experiment has concluded. The setup of the experimental process is illustrated in Figure 2.3.

A scan is taken as a series of slices. These slices are built up into data volumes. Each volume contains information from the full area that was scanned. Slice thickness is measured in millimetres, ranging from 3mm upwards. The participants head is typically braced in order to reduce any movement during the course of the experiment, and to maintain alignment between sequential slices and volumes.

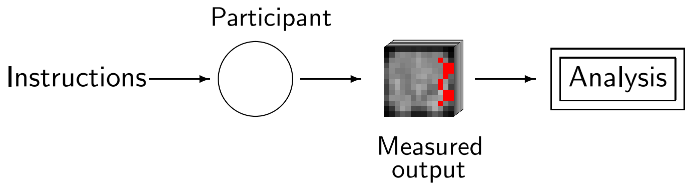


Figure 2.3: fMRI experimental setup. Stimulus is presented to participant. Scans are collected and analysed offline.

During the course of an fMRI experiment, many volumes may be acquired over time. The length of time between each volumetric scan is known as the repetition time, or TR. A typical TR will be somewhere between 1s and 3s. The lower the TR, the higher the temporal resolution of the data. The echo time, or TE, is a measurement of the scanner corresponding to the delay time between the initiation of the magnetic field pulse by the scanner and the peak of the measured response (echo). This measurement effectively determines the lag between the pulse and image sample being recorded.

The strength of the magnet in the scanner is measured in Tesla. The higher the Tesla rating of the scanner, the higher the spatial resolution of the scan. Higher spatial resolution means that finer detail can be examined. Currently, scanner ratings are typically 3-Tesla or 7-Tesla.

The activation of a voxel[1] in response to a stimulus is shown in the scan as a change in grey level intensity. Activation is expected to follow a given pattern known as the haemodynamic response function (HRF). Figure 2.4 illustrates the pattern of stimuli (black solid line) and HRF (red dashed line). The exact shape of the HRF is known to vary between subjects, psychological states, experimental conditions and different brain regions [50]. In addition to this, in practice, the response of the voxels is noisy, and hence will not follow this pattern exactly.

Changes in activation over time can be analysed. Analysing groups of voxels,

[1] A voxel is a three dimensional, volumetric pixel.

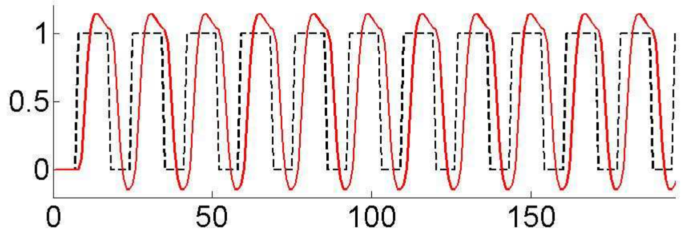


Figure 2.4: Plot of stimuli (black) and HRF (red). Time is measured on the x-axis.

or the whole volume image, allows for better interpretation of patterns representing neural processes.

Data quality can be measured by the signal-to-noise ratio (SNR) and contrast-to-noise ratio (CNR). The *image* SNR represents the quality of each individual volumetric image, and is calculated by dividing the mean activation of voxels within a volume by the standard deviation of the activation. The *temporal* SNR is calculated on a voxel by voxel basis, as the mean value of the voxel over time divided by the standard deviation of the activation of the voxel over time. Theoretically, increases in the resolution of the scan (higher scanner rating), lead to higher spatial SNR. Increasing the TR increases temporal SNR [6].

The CNR represents the maximum differences in signal intensity recorded in a given voxel over the course of an experiment. CNR is defined over time for each voxel as the ratio of task related variability (contrast) to non-task related variability (noise) [46]. The contrast can be calculated as the difference in signal between classes. High CNR means that differences in brain response to different stimuli will be more easily detected. If CNR is low, then very little difference will be detected between two conditions of interest [6]. CNR varies with TE; the optimal TE depends upon the properties of the tissue being scanned.

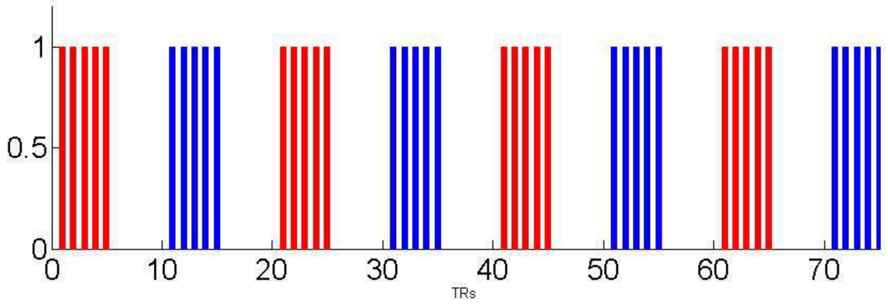


Figure 2.5: Block design fMRI experiment. Red and blue represent stimuli from two different classes.

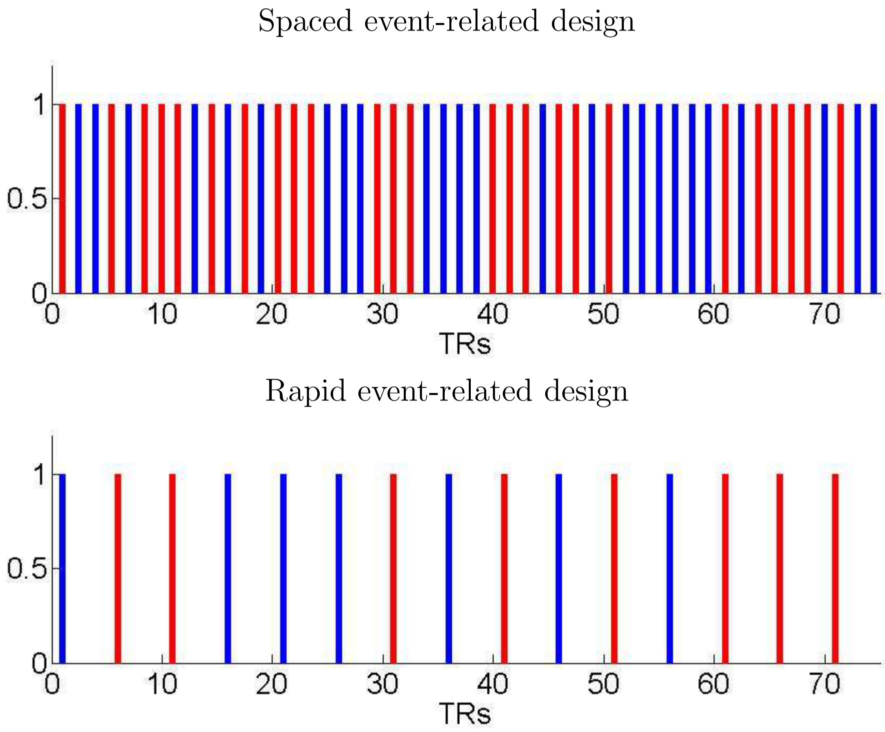
2.4 Experimental Design

The range of experimental designs available includes block design, event-related design and real time experiments.

Block design experiments repeat several different stimuli from the same class back-to-back in a 'block' in order to fully capture the response of the brain. A typical block may last between 15 and 40 seconds. After each block of stimuli, it is common to allow for a rest period in order to allow the brain to return to a resting state. This may consist of either the presentation of neutral stimuli, or no stimuli at all. Figure 2.5 illustrates an example presentation of stimuli in a block design experiment. Each vertical line represents the presentation of a stimulus. Examples of possible stimuli are the presentation of an image, sound, or instruction.

Event-related designs display stimuli from different classes in a random order of presentation. There may be rest periods after each stimulus (spaced event-related) or stimuli may be displayed successively (rapid event-related). Examples of the presentation of stimuli for event-related designs can be seen in Figure 2.6.

Real time experiments involve adjusting the stimulus based upon the response of the subject. Real time experiments are discussed further in Section 2.11.


Spaced event-related design

Rapid event-related design

Figure 2.6: Event-related fMRI experimental designs. Red and blue represent stimuli from two different classes.

Figure 2.6 illustrates the stimuli presentations in block and event-related designed experiments. Time is represented on the x-axis, the colours red and blue correspond to stimuli from two different classes.

fMRI experiments may be carried out in a single run, or over the course of several runs. Over the course of an experiment several participants may attempt the same task. Analysis across several participants is more complex than with single participant data. It is therefore beneficial to be able to analyse results using methods which can be tailored to each individual participant.

2.5 Labelling fMRI Data

Assigning class labels to fMRI data is not a trivial task. In addition to the delay from the TE, there is a delay (referred to as the haemodynamic delay) between the reaction to a stimulus and the change in the BOLD response being measured.

Assuming the experiment has a block design, as is the case with our experimental data, the first decision is how to handle the TRs. One solution is to average the responses of the TRs within a given block (temporal compression), resulting in a single data point per block. This method can reduce the effect of noise within the data, however it also significantly reduces the amount of data points available.

The alternative to temporal compression is to include each TR as a separate data point. In doing this, some noise is included in the data, however the increase in the number of data points available is a big advantage. The two approaches are compared, among others, by Mourao-Miranda et al, [95], where temporal compression was found to improve the accuracy of the analysis. Whilst this may be the case, the approach is not feasible for analysis of fMRI data in real time, where scans are required to be processed individually. Considering each TR as a separate data point makes a step towards the analysis of real time fMRI data.

Having chosen to treat each TR as a separate data point, a further step is to choose how to assign class labels. As previously mentioned, the response of the brain to a stimulus is not instantaneous, the transition between brain states is gradual, and it is unclear at which point one state ceases and the next begins. This leads to a series of potential labelling scenarios varying with the extent of inclusion of the haemodynamic delay. The simplest and most popular methods of label assignment include:

'Ignoring' the haemodynamic delay - simple box car model - As a simple solution, the instances (brain volumes) are labelled by taking the class label for each brain volume to be consistent with the stimulus being presented. This is line with one of the protocols suggested by Pereira at al [101]. It is acknowledged that this assignment does not specifically consider the delay in the brain response, however some delay is accounted for in the TE.

Taking into account the haemodynamic delay - shifted box car model - Class labels are shifted by 1 TR in order to take into account for the delay in the BOLD response.

Convolution of the signal with the HRF - In order to take into account the haemodynamic delay, labels are derived from the expected activation, rather than the presented stimuli. At any one time, the label assigned, corresponds to the stimulus with the highest activation.

2.6 Data Preprocessing

In its raw form, fMRI data is noisy, and volumes or time slices may be misaligned due to head movement or change in brain shape (due to the pulse for example). Data acquired from the scanner requires several preprocessing steps before analysis can begin. Strother, [120], provides a detailed review of the preprocessing pipeline. The most frequently used preprocessing steps are summarised below.

Slice timing correction: As each fMRI volume is acquired one slice at a time, there is a slight time difference between acquiring the first and last slice of a volume. In this time it is possible that physiological changes have occurred, due to respiration or head motion, as well as changes in the BOLD signal. The time series of the voxels within the volume are therefore shifted slightly to compensate for this, or else the same event would appear to be initiated at different times throughout the volume. That is, it would appear to start later in the first slice than the last slice of the volume.

Motion correction: Subject head movement is a significant cause of artifacts. Although the use of restraints significantly reduces head motion, slight movement may still occur due to the nature of the experimental task, or due to discomfort in the scanner. In addition to this, respiration and the pulse can cause motion and changes to the shape of the brain. The motion may appear either as a translation or a rotation or a combination thereof. Volume scans are therefore re-aligned in order to compensate for this head motion.

Spatial filtering: Spatial smoothing is achieved by convolution of each fMRI volume image with a gaussian kernel. In general, it is not a single voxel which will respond to a stimulus, rather a group of neighbouring voxels. Hence, whilst the noise factor for a given voxel is generally independent of other voxels, the activation will typically extend over several adjacent voxels.

Whilst spatial smoothing has a benefit of increasing the SNR [42] of the image, it comes at the cost of reducing the spatial resolution of the data. It is therefore important that an appropriate gaussian kernel is used to ensure a balance is achieved.

Temporal filtering: High and low-pass filtering can be applied to the time series to correct for artefacts and physiological noise. High-pass filters are used to correct low frequency physiological noise, caused by respiration (∼0.25Hz) or the pulse (∼1Hz) [42]. Low pass filtering, or temporal smoothing, can be used for denoising [120].

Preprocessing steps are usually carried out by software packages such as Brainvoyager QX (Braininnovation, Maastricht, The Netherlands) [47], AFNI [24] or SPM [42].

2.7 Data Preparation

Even when the data has been pre-processed, further steps may be considered prior to analysis. These preparatory steps are used to make the data set more manageable, and must be used with caution in order to avoid introducing bias into the data set. Bias is discussed further in Section 2.8.

2.7.1 Voxel Mask

fMRI data analysis may be carried out on the full brain scan, or on parts of the scan corresponding to specific brain regions. To extract a region of the brain a voxel mask may be applied to the data volumes. One such mask is the grey matter voxel mask. Recall that the grey matter is the area of the brain where the majority of neural activity occurs. By excluding the white matter and non-brain regions from the scan, irrelevant voxels are removed from the analysis. It is particularly desirable to avoid 'discriminatory' voxels appearing outside the brain, leading to false-positives, in a scenario similar to that of the dead salmon[2] [5]. Voxel masks may also be applied to extract regions such as the amygdala or insular cortex, in experiments focusing on emotions for example.

[2] In an experiment by Bennett et al, a frozen salmon was placed in an fMRI scanner and subjected to a series of emotionally charged images. Results from the anaysis showed the salmon to be 'responding' to the stimuli. The work highlights the danger of false positives in fMRI analysis.

There are many methods available for deriving a grey matter mask. The standard method is to derive the mask using the anatomical MRI data where the resolution is higher, and colour intensities correspond to the white and grey matter. Such a mask can be extracted using software such as the SPM toolbox [42] or Brainvoyager QX [47].

Here, an alternative method is offered for deriving an approximate grey matter mask directly from the BOLD signal.

Deriving a mask from the BOLD signal

For working with fMRI data in real time, it is advantageous to derive a grey matter mask quickly, and from the BOLD data itself. Similar to the masking procedure described for removing artifacts by Cohen, [22], two properties are considered for each voxel; its mean and variance in activation throughout its time-series.

- Voxels with a mean above a threshold are discarded - these are likely to correspond to white matter.
- Voxels with a mean below a threshold are discarded - these are likely to be outside the brain.
- Voxels with variance above a threshold are discarded - very noisy voxels are likely to appear on the edge of the brain due to image registration issues.
- Voxels with variance below a threshold are discarded - voxels with low variance are likely to be outside the brain or in the white matter.

Figure 2.7 shows examples of grey matter masks derived in this way, to illustrate the effect of changing the parameters. The sub-figures show parameter combinations relating to different thresholds (measured as percentiles) for the mean and variance. The upper threshold on the mean can be seen to alter the amount of white matter included. A lower threshold results in the area outside the brain being selected.

Altering the thresholds on the variance does not appear to have much effect visually on this 'slice' of the brain. Looking at the number of voxels included however, it can be seen that widening the range of the variance increases the number of voxels.

2.7.2 Normalisation

Another next step in data preparation is normalisation. Normalisation is carried out across the time series for each individual voxel. Normalising the signal in this way ensures that the activation values of each voxel are in the same range. This allows better identification of genuinely discriminative voxels, rather than voxels which simply have a large range of activation values.

2.7.3 Feature Selection

Recall that fMRI data has a large feature-to-instance ratio, having many more features than instances, and typically contains a large amount of irrelevant voxels. Irrelevant voxels should not be confused with redundant voxels. An irrelevant voxel does not offer any discriminatory power to the classifier. It can be thought of as noise. A redundant voxel is one whose discriminatory properties may be mirrored by at least one other voxel, or combination of voxels, in the voxel set. A large number of voxels will make the classifier prone to over fitting the training data. A preliminary voxel selection step may therefore be used. Instead of sampling from the whole set of voxels, a smaller subset of voxels may be selected first [101]. Several techniques exist for this step, and may be univariate or multivariate:

ANOVA or t-test (univariate) To test the hypothesis that the class means are equal, statistical tests such as the t-test or ANOVA calculate a statistic for each voxel. This gives an indication of the discriminatory power of that voxel based on the data points and labels passed to the test. The F-statistic, t-statistic or corresponding p-value can be used as a method to 'rank' voxels according to

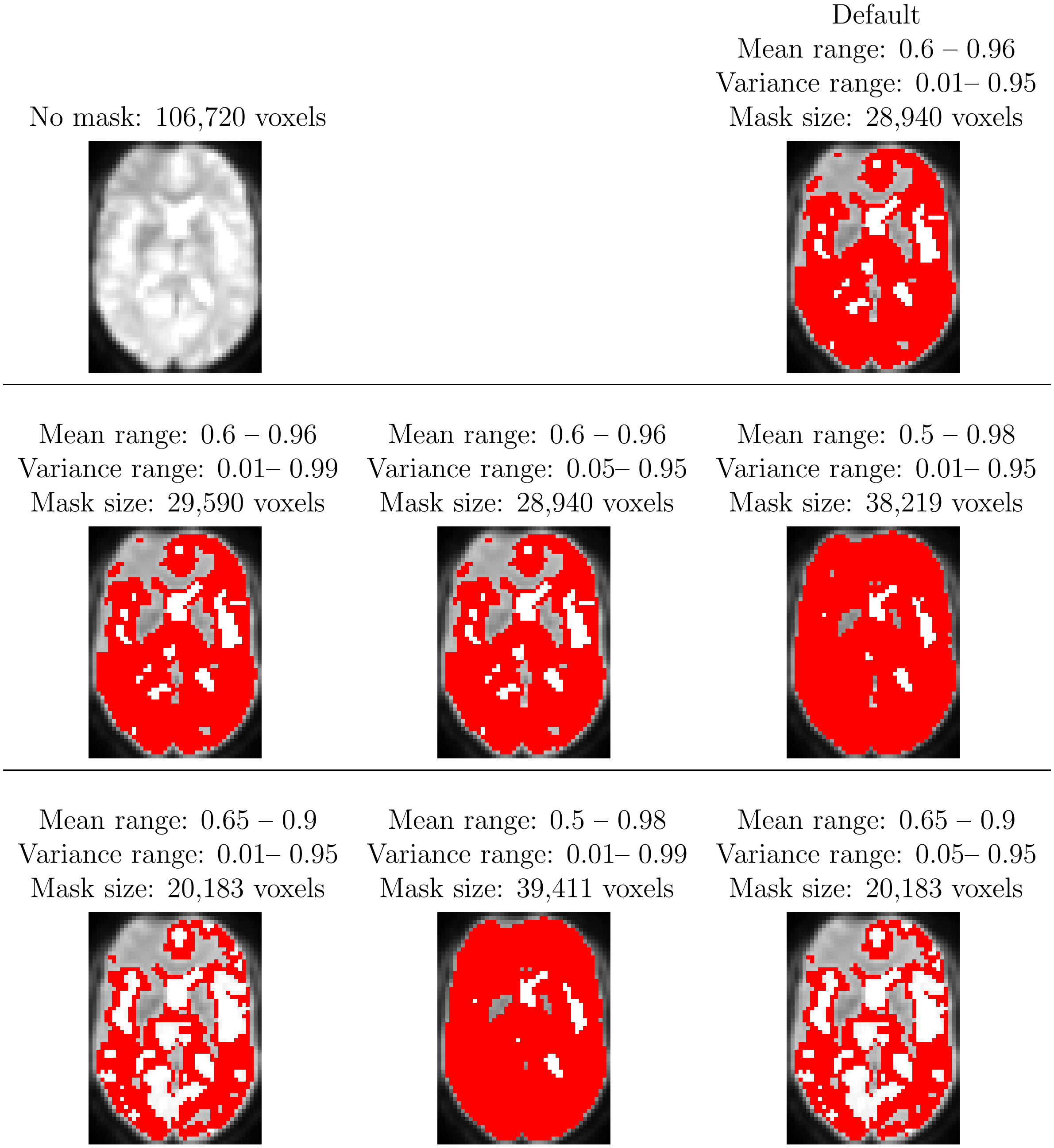


Figure 2.7: Examples of voxel masks derived from BOLD signal. Top row shows the unmasked volume slice and the default mask. Red voxels indicate those selected. Voxel mask sizes correspond to the complete volume mask.

their importance. The voxels with the lowest p-values, corresponding to the highest F or t-statistics are selected.

Maximum activation (univariate) This method considers the maximum activation of each voxel in the data set. Voxels are ranked according to their maximum activation levels. The theory is that voxels with higher maximum activation contribute more to the classification.

Two methods exist for selecting voxels from the ranked list. The first approach considers the whole data set, the overall highest ranked voxels are selected as the feature set. The second approach considers voxels on a class by class basis. For each class the highest ranked voxels are selected, the resulting class based voxel sets are merged to make up a final feature set [71]. The class based approach is a step towards ensuring that the defining features of each class are represented in the feature set, however requires a priori knowledge of the classes. Where maximum activation has been used for feature selection in this thesis, it is the overall ranking which has been considered.

SVM method (multivariate) By applying a classifier such as a support vector machine (SVM) (described in Section 3.2.1) to the data set, coefficients or weights can be derived for each of the voxels. The weights are used to rank the voxels. Those voxels with higher weights are those which contributed most to the classification, and are therefore more discriminative than those with lower weights. The voxels with the higher weights are therefore the ones which are carried forward.

Recursive feature elimination (RFE) (multivariate) This method follows a similar principle to the SVM method. Rather than selecting the highest weighted voxels, RFE eliminates the voxels with the lowest weight. The classification and weight calculation process is repeated on the new reduced voxel set. As some voxels have been eliminated, weights and the ranked order of weights may change. The voxels with the lowest weights in the new classification are eliminated. The process is repeated until only the desired number of voxels remain [49].

Other methods There are many other more complicated voxel selection methods available. These range from the searchlight algorithm [64] and Monte Carlo mapping [9] to genetic and memetic algorithms [1, 10]. Whilst techniques such as these may offer a more sophisticated solution, they pose major complexity challenges in terms of application to real time scenarios, due to their iterative nature.

Admittedly, the univariate approaches do not consider potential relationships between voxels. Some voxels may not be indicative individually, but may form part of an indicative group, which multivariate methods may identify. On balance however, the speed and simplicity of univariate techniques mean that they remain a popular choice for feature selection.

2.8 Avoiding Bias

When analysing data from fMRI experiments it is important not to introduce any bias into the data set. The first step in the analyses presented in this thesis is to separate the data set into two further data sets, the offline training data set, and the online streaming data set. So as not to introduce any bias into the analysis, it is important that the feature selection and normalisation steps, if used, are performed in the correct order. If a voxel mask is going to be applied, this should be the first step, else features may be selected from outside the desired region. Feature selection should follow, it is important that this takes place on the training data alone. The maximum activation method of feature selection in particular, will not work if normalisation takes place first. Allowing the rest of the data to be seen at this stage is known as peeking or double-dipping [65, 101]. After the features have been selected, then the remaining data may be normalised, normalisation coefficients should be calculated for the offline training data, and then applied to the streaming data volumes as they are acquired.

2.9 Traditional Analysis of fMRI Data

Much fMRI analysis focuses on the General Linear Model (GLM). When used for fMRI analysis, the GLM models the activation of voxels across the time series. The model takes the form

$$\mathbf{y} = \mathbf{x}\beta + \epsilon \tag{2.1}$$

where $\mathbf{y}$ is the response variable (BOLD response), $\mathbf{x}$ is a matrix of predictors and β is a matrix of coefficients corresponding to the relative contributions of the predictors.

Consider a simple case of a block design experiment where periods of stimulation are alternated with periods of rest. The BOLD signal of a voxel over time is the response variable y. The predictor variables $\mathbf{x}$ are the so called 'design matrix', which is the time series of the stimuli as shown for a single stimulus in Figure 2.4 (in black). Alternatively, $\mathbf{x}$ could be the stimuli convolved with the haeomodynamic response function (HRF), plotted in red in Figure 2.4. The magnitude of the coefficients, β, determines how closely the voxel output y is related to the stimuli. If for some stimulus the signals y and x_i were to be identical, then $\beta_i = 1$, all other $\beta_j = 0$, $j \neq i$ and $\epsilon = 0$. The collection of βs across all voxels in the brain can be further analysed to determine which voxels are truly related to the stimuli, and which have high β by chance. Statistical tests involving multiple comparisons can be used for this purpose.

This technique can be used to test responses to different stimuli, determine regions of interest, or to compare differences between different participant subgroups [56, 57, 88, 114]. The technique has also been used in studies about pain perception [62, 103], and more recently to investigate the brain activity of patients in a vegetative state [92].

2.10 Multivariate Analysis of fMRI Data

Whilst univariate techniques have provided insight into the functional map of the brain, multivariate machine learning techniques can advance knowledge by taking into account inter-voxel relationships. Sitaram et al [115], present the argument against univariate and region-of-interest approaches arguing that perceptual, cognitive, or emotional activities generally recruit a distributed network of brain regions rather than single locations. By applying multivariate analysis, these spatiotemporal relationships can be captured and utilised. A classifier is trained to predict which stimuli are being presented to the participant, based upon the entire fMRI volume images.

Linear classifiers are preferred for the classification of fMRI data because they are simple, fast, reasonably accurate and interpretable. Given the extremely large feature-to-instance ratio associated with fMRI data, linear classifiers are expected to outperform many other classifiers because they are less prone to over fitting the data[3].

The spectrum of linear classifiers applied to fMRI data include the linear discriminant classifier (LDC) and penalised versions thereof [48], the Gaussian Naïve Bayes [90] (linear if all variances are assumed to be equal), sparse logistic regression [131] and more. The classifier used most often, however, is the support vector machine classifier (SVM) [23, 29, 30, 66, 75, 93–95, 127, 132].

2.11 Real Time fMRI

Recently there has been interest in the development and application of real time fMRI [22, 25]. Real time experiments are typically achieved via a brain computer interface (BCI). BCI technology is by no means unique to fMRI, and is also been used with EEG, PET and MEG [31, 123, 130]. The BCI forms part of a loop known as a

[3] The classifiers mentioned in the following sections are described in more detail in Chapter 3

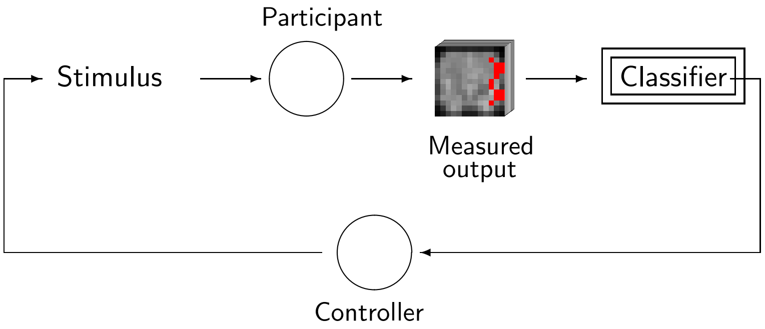


Figure 2.8: The neurofeedback loop. Based on the classification output, the participant is instructed to perform a mental exercise that will drive the brain pattern closer to one corresponding to the desirable behaviour.

neurofeedback loop. The neurofeedback loop is sketched in Figure 2.8. The participant receives initial instructions and possibly some stimuli. Next, the participant's state of mind is measured and classified.

By applying online and real time classification to the data as it is collected, neuroscientists are able to acquire feedback during the course of the trial. The efficiency and precision of real time fMRI for brain control has been demonstrated by participants carrying out tasks such as navigating through computer-generated mazes [53, 91, 133], balancing a virtual inverted pendulum [37], predicting decisions in an economic game [54], and moving an arrow towards a target [77].

Real time fMRI classification allows for self-regulation experiments with fMRI. Self-regulation is the ability to regulate ones' own emotions or behaviour. Self-regulation is achieved by controlling brain subnetworks, for example those involved in pain perception [32] or sadness [106]. Based on the measured brain state, feedback is given to the participant who then attempts to adjust his/her brain state to improve task performance. A possible application of self-regulation is treating alcohol or drug addiction. In order for self-regulation to be viable, delay between brain activity and feedback to the participant needs to be minimal [128].

Weiskopf et al, [129], provide a review of real time fMRI for BCIs and self-regulation. The review outlines technical issues raised by real time fMRI; such issues include delays in getting data from the scanner and artifact control. Researchers working with real time fMRI have to cope with artifacts such as head motion and scanner drift in real time. This is of course in addition to the speed requirements of the classifier. Hollmann et al [53, 55] tackle these issues by developing their own programming language, experiment description language (EDL), in order to unify the tasks.

Whilst many fMRI experiments are considered to be real-time, variations on the setup exist. A range of analysis setups are discussed below.

Offline analysis The traditional case, where fMRI data is collected and then analysed offline.

Offline training, real-time experiment Currently, whilst there are increasing numbers of real time studies being conducted, very few update the classifiers during the course of the fMRI run. The majority rely on a previously trained offline classifier.

In this case, two or more runs of the experiment are conducted. The first run is used for training and familiarisation with the task, data is collected and analysed offline. The classifier is trained offline on the data, and is used for real-time classification in subsequent runs. It is assumed that task performance and neural activity are static, thus the classifier trained offline on data acquired during the training run will be sufficiently accurate for classification during the testing run.

Anderson et al [2], train the classifier one day, and test on another. In this case, classification is carried out on a series of predefined brain regions. Rather than to consider each individual voxel as a feature, the method considers $4 \times 4 \times 4$ cubes of voxels. Based upon the offline data, for each voxel block, a baseline activation level is calculated as the average activation of the 64 voxels in the block, averaged over time. For classification, the features are calculated on a block by block basis as the difference between the average activation of the block in the current scan, and the value calculated as baseline for that block. Reducing the number of features in this way reduces the likelihood of overfitting, however also runs the risk that properties of a potentially indicative voxel may be diluted if it is surrounded by less indicative voxels.

Papageorgiou et al [100] also train an SVM classifier on a separate run to the neurofeedback experiment. There may be circumstances where this setup is not feasible. fMRI scanning is expensive and time-consuming. It is often desirable to complete the entire experiment, training and neurofeedback in a single run.

In a study by Eklund et al [37], subjects are required to ‘control’ a pendulum. By activating the motor cortex, participants are able to shift the pendulum either to the left, or to the right. The perceptron classifier is trained during an initial offline phase. During the real time phase, the fMRI data is classified and the stimulus (balanced pendulum) is updated accordingly. During this setup, the participant is responding independently of any stimuli. In this phase, there are no clearly defined class labels, rather a desired goal to balance the pendulum and an assumption that the participant wishes to succeed in the task! Whilst primarily chosen for its speed and ease of implementation, the perceptron was found to provide sufficient discrimination for this classification task.

Online classifier updates Given that it is not always practical to conduct multiple runs of an experiment, due to time and cost restraints, a classifier which is capable of training during the course of a run is beneficial. A weak classifier may be trained on the first few TRs of a run, and then continue to learn and adapt through the course of the experiment.

> Laconte [76], presents a series of future challenges when working with real time fMRI data. The author notes that future experiments may be designed in such a way as that the brain response is *expected* to change over time. It is suggested that future classifiers should be able to adapt and learn with the data throughout the course of the run. Online classification techniques have already been applied for EEG classification in BCI settings [11, 17, 81, 121].

Real time fMRI techniques have also been used to allow subjects to form words using a character map [36], where the cursor is moved by different motor control tasks. Classification was carried out using a single layer neural network and a multi-class SVM. The neural network showed the best results, and was preferred due to its' capability of handling multiple classes. The authors claim that for optimal performance, SVM requires classes to be independent. Practically speaking, this is not always the case. For example, in this experiment, tasks relating to the different classes involve bilateral movement - left and right hand and toe movement. Some voxels will respond to both left and right movements, thus reducing the independence between the classes. The authors hypothesise that this is one reason for the poorer performance of the SVM in this study. The authors comment that their fMRI classification system is not flawless. However beyond curiosity, such experiments serve as proof of concept: fMRI classification can be a fast and accurate component of the BCI for the purposes of neurofeedback.

The length of time taken to feed back results in some circumstances is up to a minute. Immediate feedback is defined by Weiskopf et al [129] as being recieved in under 2 seconds, this approximately equates to 'within a TR'. For real time experiments, this is the target aimed for.

A recent paper by Hollmann et al [54], shows that brain responses can be predicted 1 or 2 seconds before the participant revealed a decision. The experiment was based upon the ultimatum game, where two subjects are to share a sum of money. The first participant chooses how to split the money and the second participant has to decide whether to accept or reject the proposed split. Should the split be rejected, neither participant receives anything. The brain response of the responder was analysed using an online relevance vector machine classifier (RVM) in conjunction with their experiment description language [53, 55].

Aside from the neurofeedback impact that real time fMRI experiments can have, there are also bonuses in terms of conducting the experiment. By receiving feedback during the course of the experiment, a researcher is quickly able to spot any technical errors and respond to problems as they arise.

When analysing real time fMRI data, there are two different approaches, the incremental approach and the sliding window approach. The incremental approach calculates statistics based upon all data presented up until a given moment in time. The sliding window approach considers a subset of the total data collected within a given time span. Whilst the sliding window approach relies less on stationarity, and therefore may adapt better to concept drift, this comes at the cost of loss of statistical power [130].

2.12 Software Packages

Various software packages and toolboxes are available for fMRI analysis. The most frequently used packages include Brainvoyager QX [47], AFNI [24] and SPM [42]. Data analysis in this thesis has been carried out using Matlab [87], and the Matlab statistics toolbox, with fMRI preprocessing being carried out in Brainvoyager QX.

2.13 Data Sets

Four fMRI data sets are considered as examples in this thesis. Full descriptions are given below with a summary provided in Table 2.1.

2.13.1 Emotion_Negative (EN1 and EN2)

EN1 and EN2 are two runs of the same experiment, corresponding to single runs with two different participants. Participants were instructed to up-regulate their target region activity, evoking emotion, for periods of 20s using negative emotional imagery. Periods of emotion are alternated with baseline periods of rest, of 14s. There were 12 blocks of up-regulation and rest. The classification task is to distinguish between periods of emotion, and periods of rest.

2.13.2 Emotion_Both (EB)

A single participant viewed a series of emotionally charged images in a block design experiment. A sequence of fMRI brain scans was obtained from a single run. There were 12 blocks of images with positive valence type, 12 blocks with neutral valence type and 12 blocks with negative valence type. Each block of images lasted for a period of 6 s (4 pictures presented for 1.5s) followed by a period of fixation (12s duration)[4]. Fixation TRs are removed from the data set. The classification task is to distinguish between positive, negative and neutral emotion.

For EN1, EN2 and EB, data was collected on a 3 Tesla Philips Achieva MR scanner (TR = 2s, TE = 30ms, 30 slices, in-plane resolution $2 \times 2\text{mm}^2$, 3mm slice thickness). Slices were positioned such that the bottom slice was 30mm ventral to the anterior commissure and angled to encompass all of the ventral prefrontal cortex.

Preprocessing of the data was performed using Brainvoyager QX. The data were corrected for intra-subject angular and translational motion and filtered to remove long-term drift [58].

[4] The images were selected from a benchmark database, International Affective Picture System (IAPS) [79], where each image has been rated on scales from 0 to 10 on two dimensions: arousal (calm to excited) and valence (negative to positive)

Table 2.1: Summary of the three fMRI data sets

Name	# Instances	Volume	# Voxels	Classes	# Runs
	size				per run
EN1	204	60 × 31 × 44	81840	2	1
EN2	204	59 × 32 × 44	83072	2	1
EB	109	60 × 62 × 45	167400	3	1
Bangor 2	360	58 × 40 × 46	106720	3	3

2.13.3 Bangor 2

The participant viewed visual stimuli in 14 second blocks. The stimuli were taken from three categories: faces, places and objects, plus a control block of fixation. Three runs were carried out. For each category and fixation period, in each run, there were six presentations from each category. Within each block, the individual stimuli were presented at a rate of 1 Hz. Each run consisted of 120 TRs with blocks of stimuli permuted across the runs. The data is pooled across the runs, giving an overall total of 360 data points. The size of each scanned volume was 59 × 32 × 44, resulting in 106720 voxels per volume. Data was collected on a 1.5 Tesla Philips Achieva MR Scanner (TR = 2s, TE = 50ms, 20 slices, in-plane resolution $4 \times 4\text{mm}^2$, 5 mm slice thickness). Preprocessing was again performed using Brainvoyager QX.

2.14 Summary

This chapter has introduced techniques for investigating the brain, with a focus on functional magnetic resonance imaging (fMRI). fMRI data will be used throughout this thesis. Different experimental designs for fMRI experiments are introduced. Steps for preprocessing fMRI data prior to analysis have been described, including optional extra steps such as feature selection, for preparing the data. In order to extract the relevant voxels from the brain, a technique for deriving an approximate grey matter mask from the BOLD signal has also been introduced.

The progression of fMRI analysis is discussed, from traditional univariate techniques such as correlations with the GLM, through to multivariate classification and real time experiments. For classification, methods of assigning class labels are discussed.

Finally, the chapter introduces EN1, EN2, EB and Bangor 2, the four fMRI datasets which are used throughout this thesis.

Chapter 3

Linear Classifiers for Streaming Data

3.1 Classification

A classifier can be thought of as a 'black box' into which an instance (data point) is passed, and a class label is produced as output, as in Figure 3.1.

Taking N training instances, each described by n features; each instance comes from one of C classes. Each instance $\mathbf{x}_i \in \Re^n$, $i \in \{1, \ldots, N\}$, together with its corresponding label y_i, $y \in \{1, \ldots, C\}$ is passed to a training algorithm. The training algorithm outputs a function, or series of functions, which are used to discriminate between the classes. These are based upon the features of the training set. Subsequently, when a new instance is passed to the classifier, the classifier calculates the class label by combining the properties of the new instance with the functions calculated during training. If the label assigned by the classifier does not match the true label, then the classifier is noted as having made an error. For non-streaming data, a separate testing data set, Te may be used. The error is measured as the proportion

Data Point
Classifier
Class Label

Figure 3.1: A classifier is a 'black box' into which a data point is passed as input, and a class label is produced as output.

of instances from Te which are incorrectly classified.

Maximum achievable classifier accuracy depends upon the distribution of the classes, and the nature of the classifier chosen. Some common types of classifier are discussed with examples below.

3.2 Offline Linear Classifiers

This work focuses mainly on linear classifiers due to their simplicity and speed. A linear classifier can only achieve 0% error on the training data in cases where classes are linearly separable. As noted in [89], for fMRI data, LDA and linear SVM can actually perform better than non-linear classifiers, possibly because the latter are more prone to overfitting.

3.2.1 Support Vector Machine (Linear Kernel)

The support vector machine (SVM) finds the maximum margin hyperplane between two classes [13, 124]. The SVM uses a kernel to map features into a high dimensional feature space in which the classes are separable. There are many types of kernel available, with linear, polynomial or radial basis functions being the most popular. Here the focus is upon the linear kernel.

The SVM takes as input the data set and labels. The labels are in the form $y \in \{-1, 1\}$. The SVM seeks to find the maximal margin hyperplane, $\mathbf{w}^T x_i + b = 0$ such that $\mathbf{w}^T x_i + b < 0$ when $y_i = -1$ and $\mathbf{w}^T x_i + b > 0$ when $y_i = 1$.

By finding the maximal margin hyperplane, that is, the boundary with maximal distance to the nearest data points, the risk of misclassification of unseen data points is minimised. The support vectors correspond to those data points which control the width of the margin, and thus are those closest to the boundary.

Closely related to the SVM is the relevance vector machine (RVM). The form of the RVM is similar to the SVM, however instead of outputting class labels, the RVM outputs a probabilistic classification.

3.2.2 Linear Discriminant Classifier

The Linear Discriminant Classifier (LDC) can be described as the minimum-error classifier for classes with normal distribution and equal covariance matrices. The LDC is a robust classifier, and thus can also yield good results even when the classes are not normally distributed. Linear discriminant classifiers rely on a simple assumption of Gaussianity of the data, which is often not met in practice. However, even when the assumption does not hold, linear classifiers have been found to be surprisingly accurate [51].

Denote by $P(y)$ the prior probability for class y. The prior probability for class y can be estimated as the proportion of training data coming from class y. The mean and covariance matrix of the data set are represented by μ and Σ respectively. For a training set of N data points, we represent estimates of the mean and covariance matrix by $\mathbf{m}_N$ and S_N respectively[1]. For the training data set, these are calculated as

$$\mathbf{m}_N = \frac{1}{N}\sum_{j=1}^{N}\mathbf{x}_j \tag{3.1}$$

$$S_N = \frac{1}{N}\sum_{j=1}^{N}(\mathbf{x}_j - \mathbf{m}_N)(\mathbf{x}_j - \mathbf{m}_N)^T \tag{3.2}$$

Note that these are the biased estimates, non-biased estimates can be calculated by replacing $\frac{1}{N}$ by $\frac{1}{N-1}$.

The discriminant function for class y is described by

$$g_y(\mathbf{x}) = \log[P(y)] - \frac{1}{2}\mu_y^T\Sigma^{-1}\mu_y + \mu_y^T\Sigma^{-1}\mathbf{x} \tag{3.3}$$

which becomes

$$g_y(\mathbf{x}) = \log[P(y)] - \frac{1}{2}(\mathbf{m}_N^{(y)})^T S_N^{-1}\mathbf{m}_N^{(y)} + (\mathbf{m}_N^{(y)})^T S_N^{-1}\mathbf{x} \tag{3.4}$$

[1] Assume all vectors are column vectors

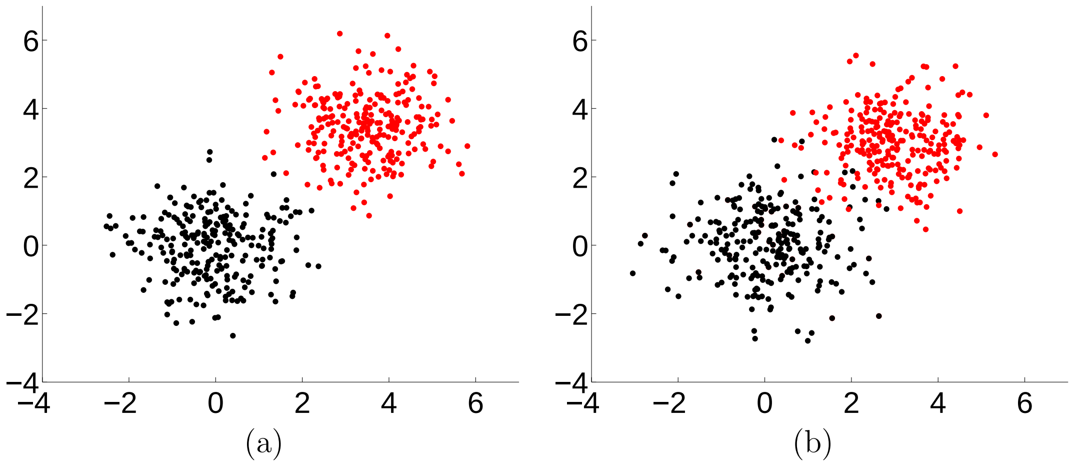


Figure 3.2: (a) Data set 1: Two classes which are linearly separable in two dimensions. (b) Data set 2: Two classes which are not linearly separable in two dimensions.

When an unseen data point is presented to the classifier for testing, the classifier calculates $g_y(\mathbf{x})$ for $y = \{1, \ldots, C\}$. The class y corresponding to the highest value of g_y is assigned to $\mathbf{x}$.

3.2.3 Examples

Examples are calculated based on two synthetic two-dimensional data sets. Both data sets are generated from gaussian distributions, separated and labelled to form two classes. In one data set the data points are labelled such that the two classes can be linearly separated in two dimensions. The second data set is labelled in such a way that the classes can not be separated by a linear boundary. Each data set has 250 instances coming from each class. Figure 3.2 illustrates the two data sets. For each data set a stratified sample of 40% of the data is used for training the classifier. The remaining 60% is used for testing.

Linearly Separable Data Points

The SVM with linear kernel and the LDC are compared for the linearly separable data set. The results are shown in Figure 3.3. The boundary derived from the LDC is indicated in blue, with the boundary derived from the SVM in green. Data points

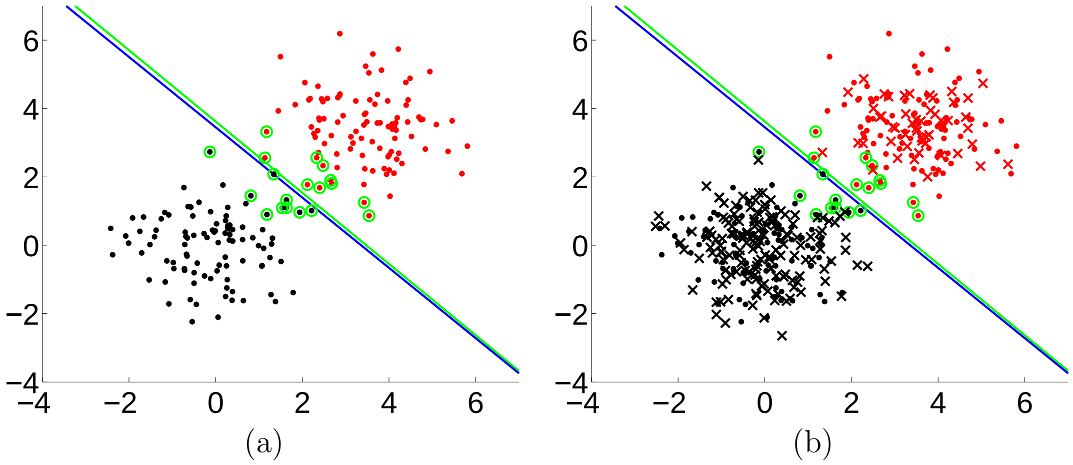
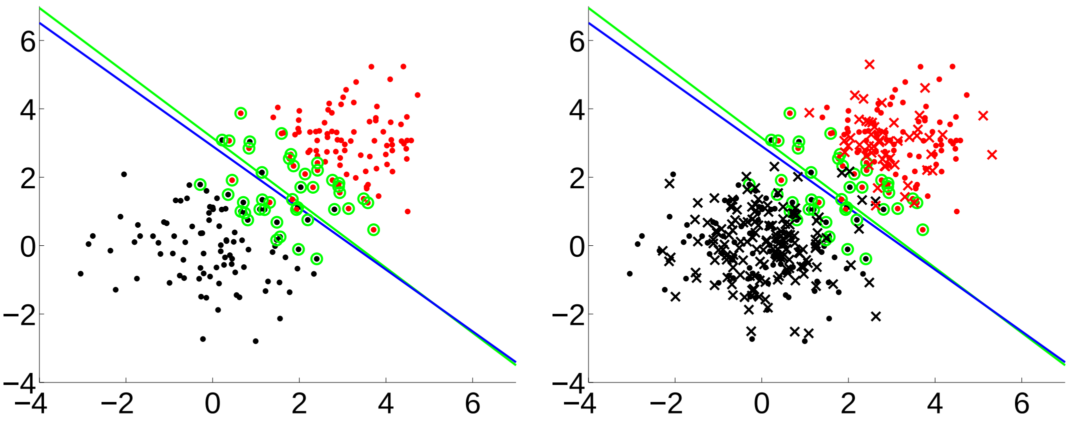


Figure 3.3: LDC (blue) and SVM (green) for linearly separable data.
(a) Training data
(b) Training and testing data.

circled in green correspond to the support vectors of the SVM. For this data set where both training and testing data points are linearly separable, both classifiers achieve 100% accuracy on both training and testing data. There is little difference between the class boundaries derived from the different classifiers.

Data Points Which Cannot be Separated by a Linear Boundary

The SVM with linear kernel and the LDC are also compared for the data set where data points are not linearly separable. The results are shown in Figure 3.4. The boundary derived from the LDC is indicated in blue, with the boundary derived from the SVM in green. Data points circled in green correspond to the support vectors of the SVM. For this data set, the data points are not linearly separable in either the training or the testing data. The training and testing error of both LDC and SVM is noted in Table 3.1. There is no significant difference in the performance of the two classifiers.



(a)

(b)

Figure 3.4: LDC (blue) and SVM (green) for non linearly separable data.
(a) Training data
(b) Training and testing data.

Table 3.1: Training and Testing Error for SVM and LDC (%)

	SVM	LDC
Training error	4.5	4
Testing error	2	2
Overall error	3	2.8

Table 3.2: Training and Testing Error for SVM and LDC (%) on fMRI Data

	SVM	LDC
Training error	0	0
Testing error	20.11	23.37
Overall error	21.08	18.14

3.2.4 fMRI Data

The two classifiers are also tested on fMRI data using the data set EN1, as described in Section 2.13. A grey matter mask was calculated to remove irrelevant voxels. A stratified sample of 9 data points per class was taken for training - this reflects the amount of data which would be available for training in a neurofeedback experiment. ANOVA was used on the training data to extract 2000 voxels to use as features in the classification. The data was normalised, using coefficients calculated on the training data, again reflecting the methodology used in real time fMRI analysis. The remaining 186 data points were used for testing. Table 3.2 shows the training and testing errors for the two classifiers.

In this case the SVM performs better than the LDC, however the ability of the LDC to handle multiple classes online makes it more desirable for the purpose of this thesis. Accuracy in fMRI experiments varies dependent upon the experiment. Hollmann et al [54] report accuracies of 70% using an RVM in a multi-subject trial. Experimental design also influences the acheivable classifier accuracy. Mourao-Miranda et al [94], compare classifier accuracy for using averaged TRs (temporal compression) versus treating each TR as a separate data point. Using temporal compression the acccuracy was 90%, whist treating each TR as a separate training example accuracy was 74%. The latter approach is the one used in this thesis as it is most representative of the real-time scenario.

3.3 Online Linear Classifiers

Many domains where data is received in a streaming format are suited to online learning. Such data comes from areas including credit card transactions, telecommunications and internet searches [33, 34, 119].

Internet search data [21] and detection of suspicious URLs [85] are two applications of online classification. Another major application of online classification is spam filtering, with online linear classifiers [107, 112, 125]. More recently, online classifiers have been applied to EEG data [11, 17, 81, 121].

An initial offline classifier will be trained on a small offline training sample. During the subsequent online classification each data point is classified by the 'current' classifier as it becomes available. For the purpose of comparing approaches, it is assumed that true class labels are recovered immediately after classification. If the 'current' classifier misclassifies the incoming data point, then the classifier is updated by adding the new data point to the training set, and recalculating the parameters of the classifier. A classifier which updates only when a data point is misclassified in this way is known as an *error driven* classifier.

3.3.1 Linear Perceptron for Streaming Data

The perceptron is an online linear classification algorithm for two classes, developed by Rosenblatt [111]. The classifier consists of a single discriminant function, which acts as a boundary between the two classes. The algorithm first initialises coefficients, or weights, $\mathbf{w} = [w_0, \ldots, w_n]^T$ as small random numbers. A learning parameter η is also defined. The learning parameter corresponds to the 'readiness to learn' of the algorithm, and defines the weighting of new data points compared to past data.

Assuming i data points have already been presented to the classifier, denote the next data point as $\mathbf{x}_{i+1} \in \Re^n$ with its true label y_{i+1}, initially unavailable. The data point is augmented, $\mathbf{z} = [1 \quad \mathbf{x}_{i+1}^T]^T$, where the first element, 1, will multiply the bias coefficient w_0. The data point is then classified by the 'current' classifier. The predicted label (+1 or −1) for $\mathbf{x}_{i+1}$ is calculated as $y_{\text{predicted}} = \text{sign}\left(\mathbf{z}^{\mathrm{T}}\mathbf{w}\right)$, where $\text{sign}(a) = 1$ if $a \geq 0$ and $\text{sign}(a) = -1$ if $a < 0$.

If the data point is misclassified, the weight vector is updated as $\mathbf{w} \leftarrow \mathbf{w} - \eta\, \mathbf{z}\, y_{i+1}$.

3.3.2 Balanced Winnow

The balanced winnow, [83], follows a similar principle to the perceptron, however has two sets of weights; a positive set $\mathbf{w}^+$ and a negative set $\mathbf{w}^-$. Both sets of weights are initialised as positive random numbers, and a learning rate β is chosen. The predicted label for $\mathbf{x}_{i+1}$ is calculated as $y_{\text{predicted}} = \text{sign}\left(\mathbf{z}^{\mathrm{T}}\left(\mathbf{w}^+ - \mathbf{w}^-\right)\right)$.

Following misclassification of a new data point, the $n+1$ weights of the balanced winnow are updated. If $y_{i+1} = +1$, then $w_j^+ \leftarrow \beta^{-z_j} w_j^+$ and $w_j^- \leftarrow \beta^{z_j} w_j^-$, else if $y_{i+1} = -1$, then $w_j^+ \leftarrow \beta^{z_j} w_j^+$ and $w_j^- \leftarrow \beta^{-z_j} w_j^-$, for $j = 0, 1, \ldots, n$.

3.3.3 Online Linear Discriminant Classifier

The online linear discriminant classifier (O-LDC) is an adaptation of the linear discriminant classifier. One immediate advantage of the O-LDC over the perceptron or winnow, is the capability to classify data sets of more than two classes. In order to carry out online updates the means and *inverse* covariance matrix require updating after each misclassified data point. Let $\mathbf{m}_{i_y}^{(y)}$ be the estimate of the mean for class y, where i_y is the number of points from class y thus far. The total number of points in the series is $i = i_1 + i_2 + \ldots + i_C$. Let S_i be the estimate of the common covariance matrix calculated from the i observations. Suppose that $\mathbf{x}_{i+1}$ comes from class y. The recursive update for the mean of class y is calculated as

$$\mathbf{m}_{i_y+1}^{(y)} = \frac{1}{i_y+1}\left(i_y \mathbf{m}_{i_y}^{(y)} + \mathbf{x}_{i+1}\right). \tag{3.5}$$

The inverse covariance matrix for class y is updated as

$$S_{i+1}^{-1} = \frac{i+1}{i}\left(S_i^{-1} - \frac{S_i^{-1}\,\mathbf{z}\,\mathbf{z}^T\,S_i^{-1}}{\frac{i(i_y+1)}{i_y} + \mathbf{z}^T\,S_i^{-1}\,\mathbf{z}}\right), \tag{3.6}$$

where $\mathbf{z} = \mathbf{x} - \mathbf{m}^{(y)}_{i_c+1}$. The prior probabilities are also updated, for class y, as $P^{(y)}_i = \frac{i_y+1}{i+1}$, and for all other classes, $P^{(k)}_i = \frac{i_k}{i+1}$, where $k \neq y$. The O-LDC update is lossless. This means that the recursively calculated estimates of $\mathbf{m}^{(y)}_{i_y}$ and S_i are the equivalent of those calculated using all i data points [35, 68].

3.3.4 Examples

For illustration, the same two data sets are used to show the progress of the online classifiers. For each data set an initial stratified sample of 10% of the data is used for an offline training data set, T. The remaining instances are shuffled and presented as an online streaming data set, S. The classifier will update after each streaming data point. The accuracy progression of the online classifiers is illustrated by a series of plots. In each figure, the top left plot shows discriminant function as calculated on the training data. Subsequent plots illustrate the performance of the classifier after different amounts of online data have been presented.

In this thesis, two approaches are considered for the measurement of online error. Due to the streaming nature of fMRI data, instead of a single error score, it is useful to consider an error progression, in terms of time. The first approach, as with the non-streaming case, is to consider a separate training data set Te, of size N_{Te}. The error rate is measured as the proportion of instances from Te which are misclassified by the current classifier at time t.

$$\frac{\sum_{i=0}^{N_{Te}} e(i)}{N_{Te}} \tag{3.7}$$

where $e(i) = 0$ if the classifier has labelled point i correctly, and 1, otherwise.

The second approach uses the online dataset, S to measure error. At time t, instance $\mathbf{x_t}$ is classified by the current classifier prior to training. If $\mathbf{x_t}$ is misclassified, then the classifier is noted as having made an error at time t. The error rate of the classifier can be calculated as the number of data points incorrectly classified divided by the number of data points presented so far. The error rate can be represented mathematically as an equation,

$$\frac{\sum_{j=0}^{t} e(j)}{t} \tag{3.8}$$

where $e(j) = 0$ if the classifier has labelled the point at time j correctly, and 1, otherwise.

Linearly Separable Data Points

Figures 3.5, 3.6 and 3.7 show the perceptron, winnow and O-LDC for linearly separable classes. For each figure, the top left plot shows the boundary derived from the training data. Subsequent plots show the progression of the boundary after online updates. The title of each plot indicates the amount of online training data which has been processed. New instances, that is those which have been presented since the last plot, are illustrated by an x.

Not Linearly Separable Data Points

Figures 3.8, 3.9 and 3.10 show the perceptron, winnow and O-LDC for the case where the training data is not linearly separable. Once again, the top left subplots illustrate the classifier decision boundary after the training data, with subsequent plots illustrating new instances as an x.

Cumulative Error

Figure 3.11 shows the cumulative error plots for the online classifiers for both data sets. Cumulative error is calculated using Equation 3.8. In Figure 3.11(a) it appears that there is no error progression for the O-LDC. This is due to the O-LDC achieving 100% accuracy on this data set, thus the error rate appears as a flat line at 0, coincinding with the x-axis.

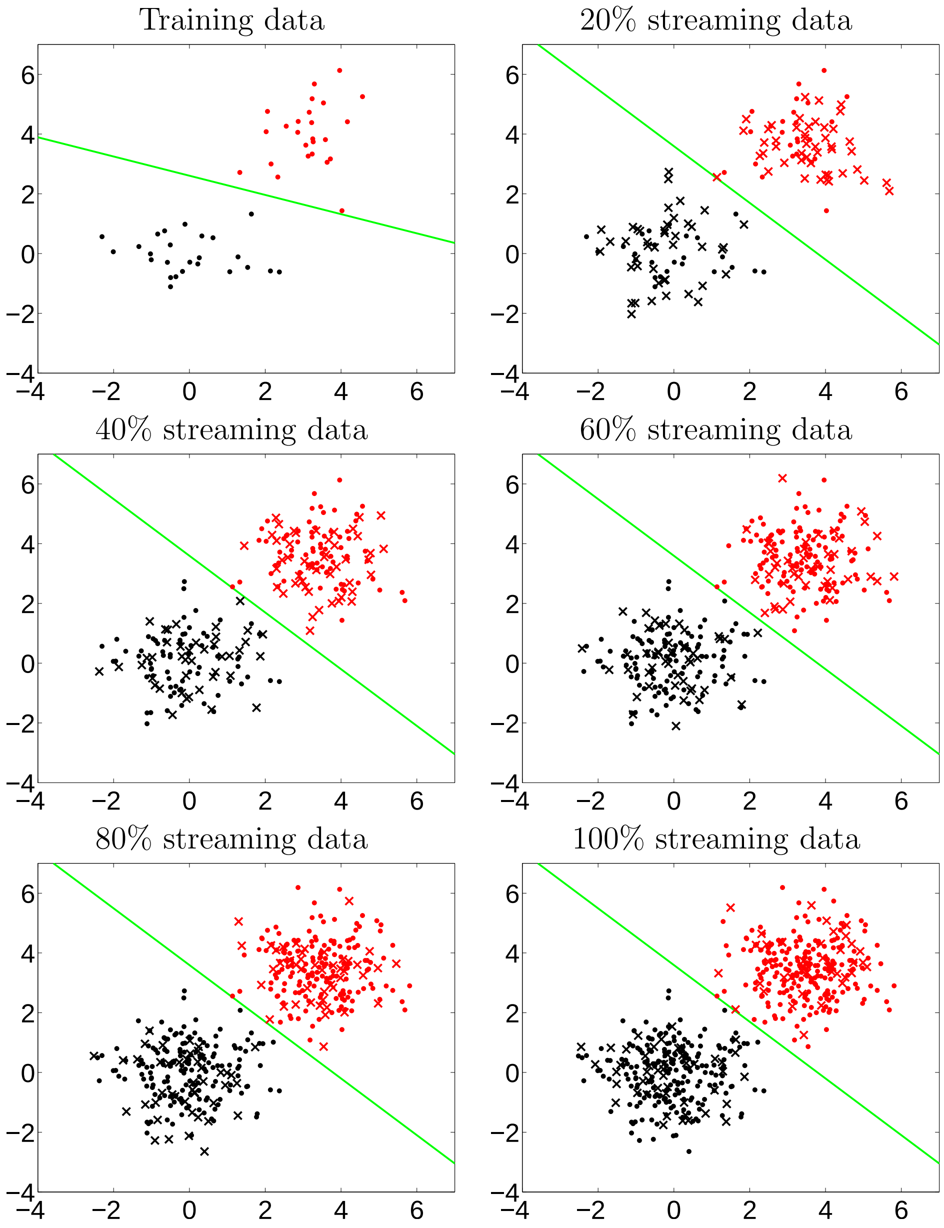


Figure 3.5: The online Perceptron for linearly separable data.

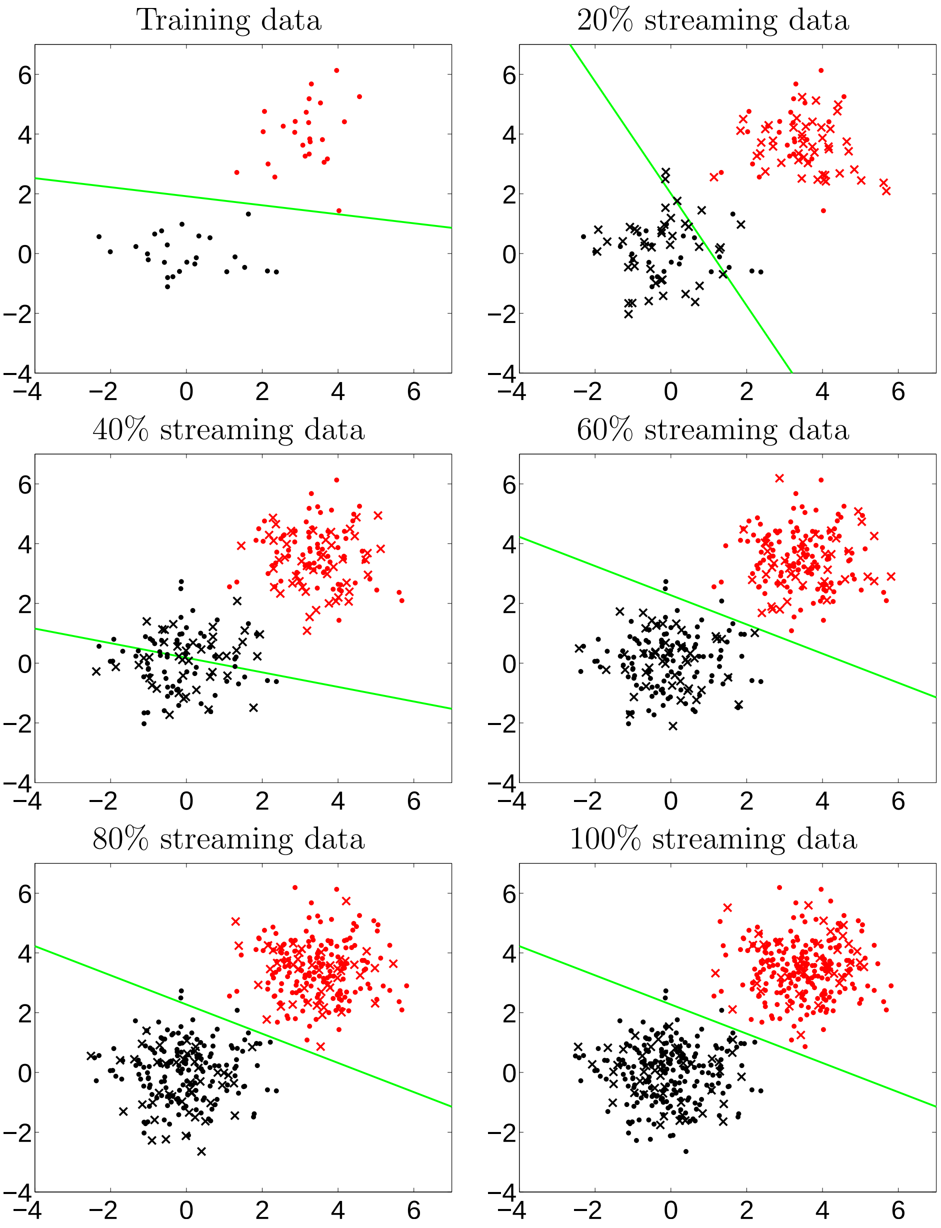


Figure 3.6: The online Winnow for linearly separable data.

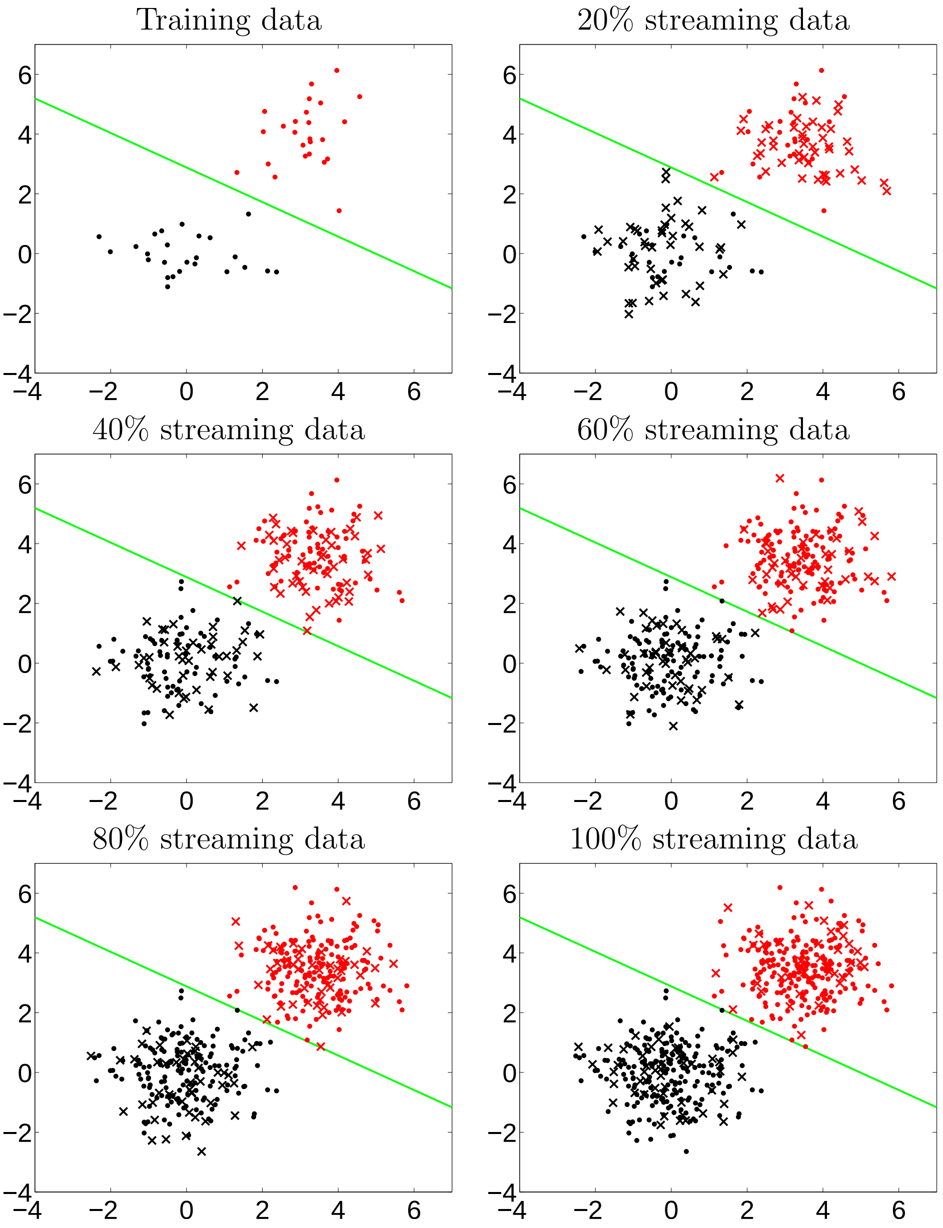


Figure 3.7: Online linear discriminant classifier for linearly separable data.

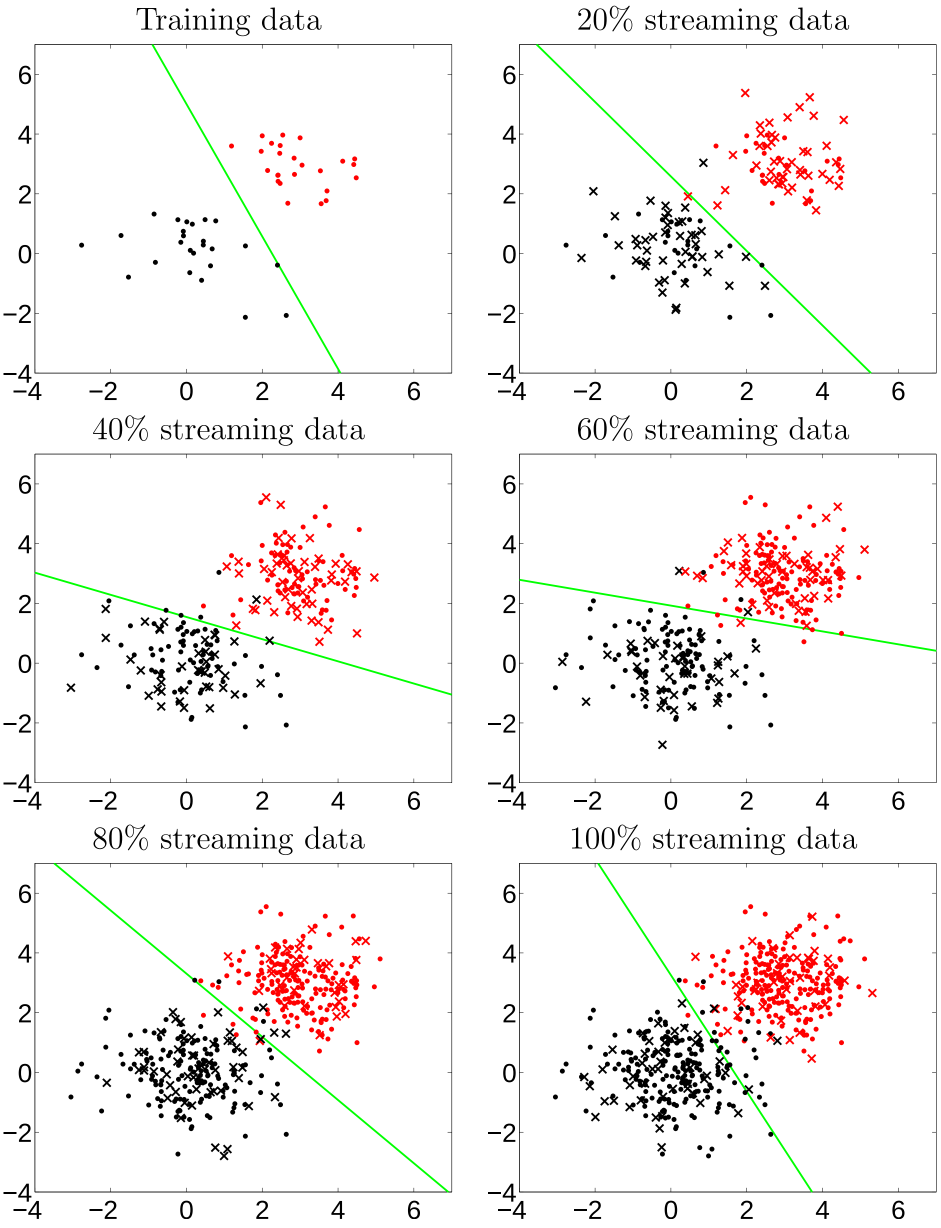


Figure 3.8: The online Perceptron for non linearly separable data.

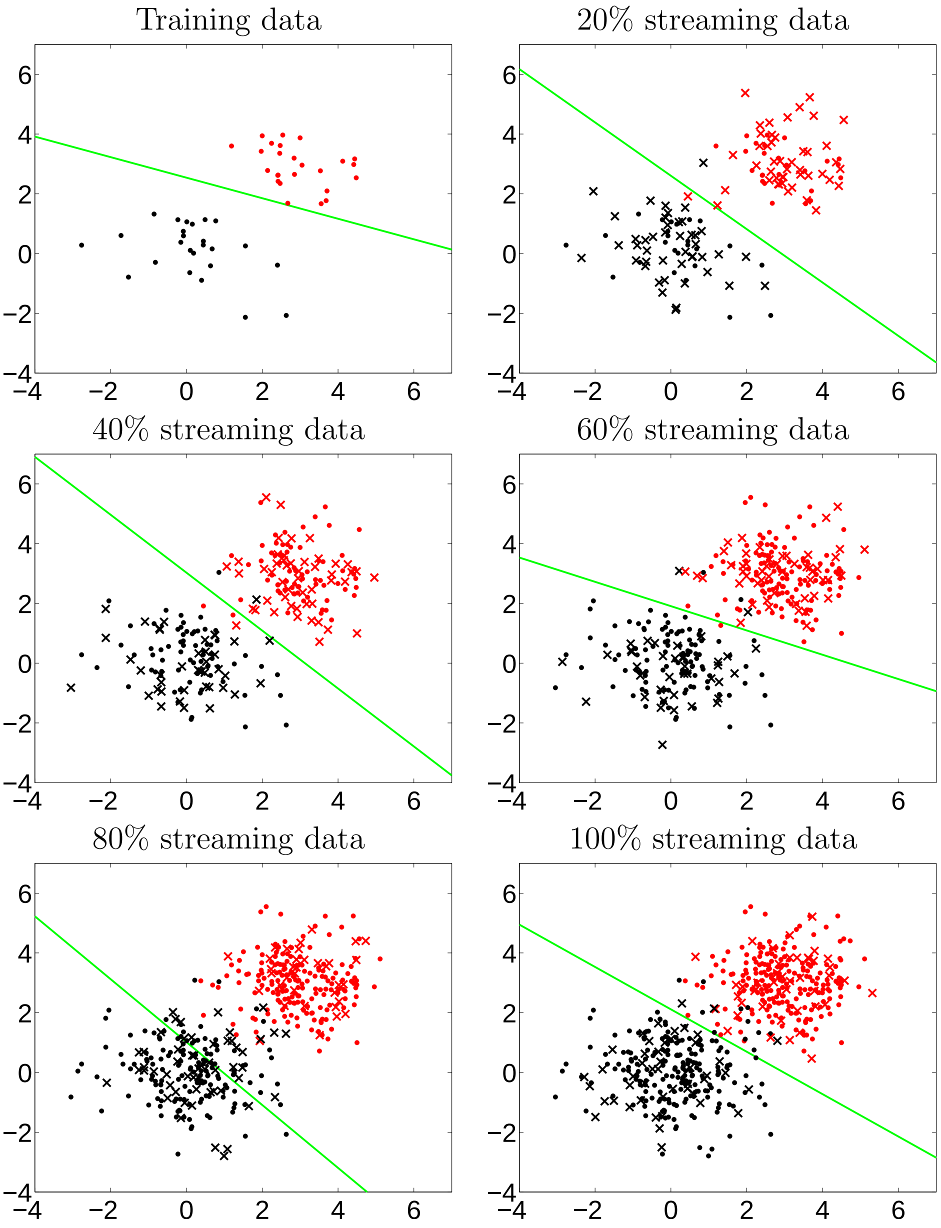


Figure 3.9: The online Winnow for non linearly separable data.

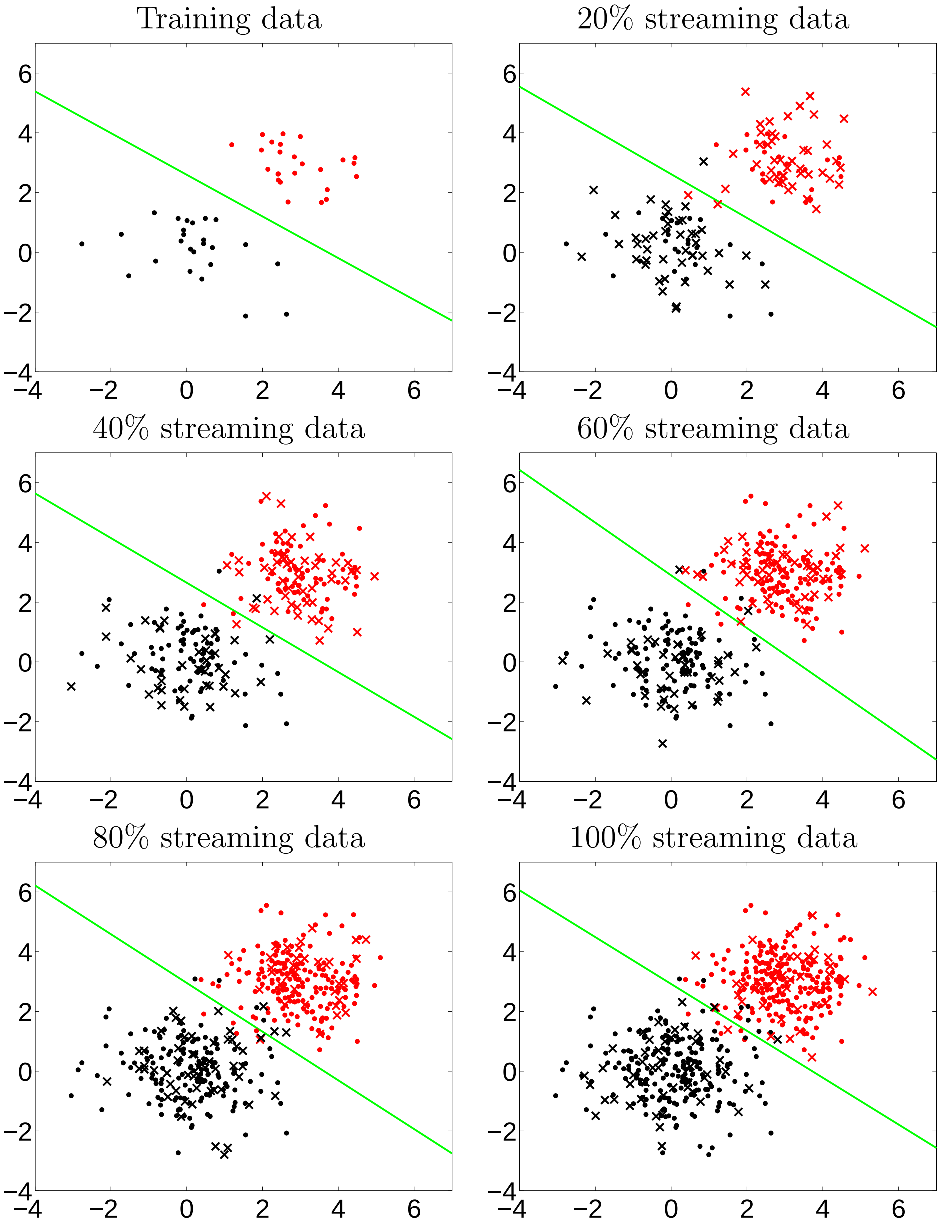
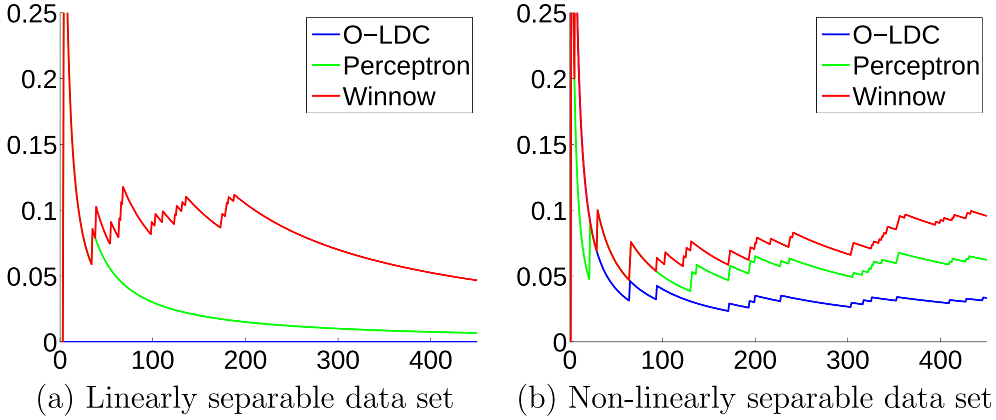


Figure 3.10: Online linear discriminant classifier for non linearly separable data.



(a) Linearly separable data set (b) Non-linearly separable data set

Figure 3.11: Cumulative error progression.

3.4 Comparison of Online Linear Classifiers for Real Data

An example is presented here from a previous work [104], to illustrate the choice of classifier for online data. The three online linear classifiers, the perceptron, balanced winnow and O-LDC, are compared across a selection of two-class i.i.d. data sets.

3.4.1 Data Sets

The data sets used for the tasks are summarised in Table 3.3.

Experimental results with the balanced winnow indicated some discrepancies when the features had a large range of ranges and variances. To compensate for this, all data was normalised with mean $\mu = 0$ and variance $\sigma^2 = 1$ for each feature.

3.4.2 Method

Each data set is prepared in the same way. Firstly, a testing data set, Te, is generated by taking a stratified sample of 10% of the data points. A second stratified sample is taken to make up an offline training data set, T. It is said, that to accurately train a classifier, the size of the training data should be approximately $10 \times n \times c$, where n is the number of features and c is the number of classes in the data set [96]. As such,

Table 3.3: Data sets used in the experiment

Data set	Features	Classes	Data points	Source
sonar	60	2	208	UCI
laryngeal1	16	2	213	Collection[1]
votes	16	2	232	UCI
breast	9	2	277	UCI
heart	13	2	303	UCI
liver	6	2	345	UCI
spect	44	2	349	UCI
ionosphere	34	2	351	UCI
wbc	30	2	569	UCI
laryngeal2	16	2	692	Collection
pima	8	2	768	UCI

[1] Collection `http://pages.bangor.ac.uk/~mas00a/activities/real_data.htm`

we set N_T, the cardinality of the training data, to be $N_T = 1 \times n \times c$. The remaining data points make up the online streaming data set, S. S is shuffled in order to remove bias and ensure that the data is i.i.d..

Offline versions of the classifiers (perceptron, winnow and O-LDC) are trained on T. Data points from S are then passed to each classifier in a streaming fashion. The classifiers are tested on the incoming data point. If the classifier makes an error, then the classifier is updated accordingly. The task is repeated 100 times for each classifier on each data set. The error of the classifier is calculated on Te after each presentation of a new data point in the online phase, using Equation 3.7.

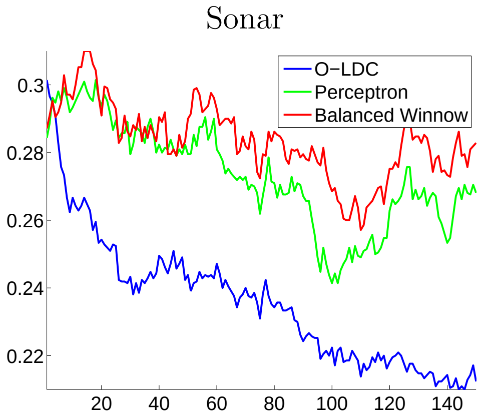
3.4.3 Results

The results are organized as follows: Figure 3.12 to 3.14 show results of the comparison of the online algorithms. For each graph, the y-axis represents the error score on Te. The x-axis represents the number of data points from S which have been processed. The blue, red and green lines represent the O-LDC, perceptron and balanced winnow respectively. The plots show the progression of error scores. The data tables show the average and final testing error scores for the data sets.

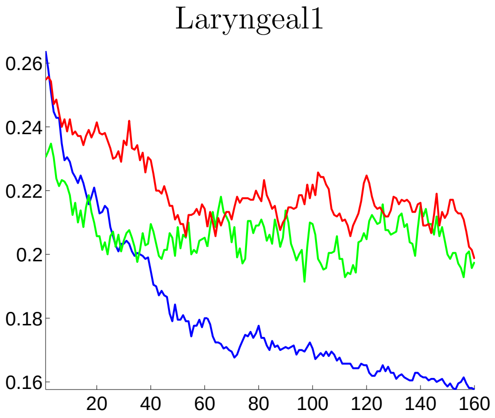
3.4.4 Discussion

Figures 3.12 to 3.14 show the comparisons between the three online classifiers for each data set. The end points of the plots give an indication as to how well each classifier has performed on the particular data set, and represent error scores calculated after the entirety of S has been presented. The precise values for these error scores can be seen in the corresponding tables. For three of the data sets, 'spect', 'ionosphere' and 'wbc', the performance of the three classifiers are comparable, with no distinctly better model. For the other eight data sets the results show the O-LDC to be a better model than the perceptron or balanced winnow.

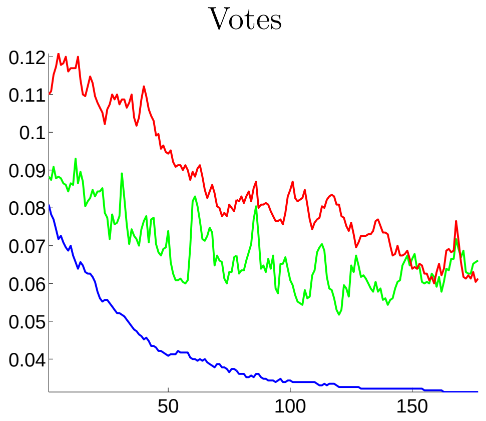
Whilst the end results of the O-LDC are impressive, the learning patterns of the classifiers are also of interest. The learning patterns can be seen from the shapes of the curves. The learning patterns of the O-LDC are markedly better than those of the other two algorithms. The O-LDC is seen to converge whilst the perceptron and balanced winnow oscillate. The learning rate of the classifiers can also be described numerically as the average error of the classifier throughout the online run. Having a lower average error indicates a better learning pattern, as the classifier has converged faster. The values for the average error is also given in the error tables. s Paired t-tests were carried out on both the final error and average error scores across the data sets. The significance level for the paired t-tests was set at 0.05. The results of the paired t-tests can be seen in Figure 3.15. The plots show wins versus losses for the three algorithms. As there are three algorithms and eleven data sets, $3 \times 11 = 33$ comparisons are made. Each classifier is part of 22 comparisons, thus the best point is marked at 22 wins and no losses, and the worst point at 0 wins and 22 losses. For the paired t-test on the final error score, the O-LDC had 16 significant wins and 1 loss. The number of wins and losses for each algorithm is represented by (wins,



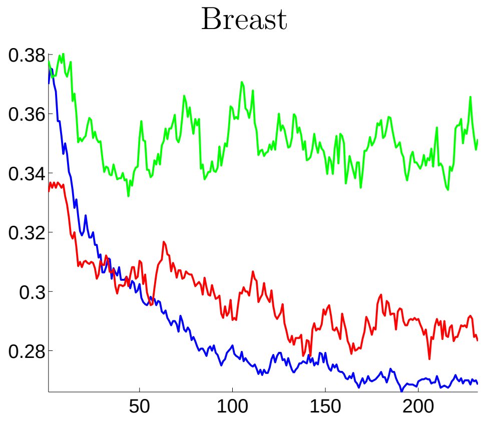
	Average	Final
O-LDC	0.2474	0.2314
Perceptron	0.2772	0.2729
Winnow	0.2824	2690



	Average	Final
O-LDC	0.1862	0.1648
Perceptron	0.2009	0.1914
Winnow	0.2185	0.2067

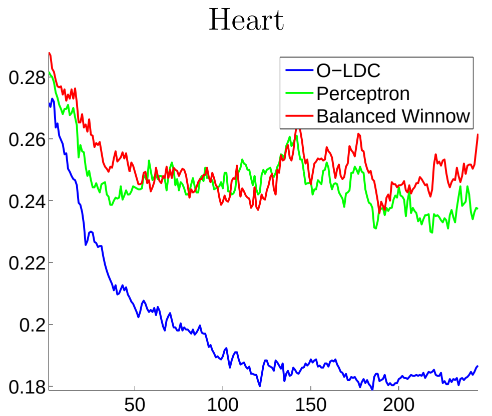


	Average	Final
O-LDC	0.0448	0.0317
Perceptron	0.0746	0.0687
Winnow	0.0855	0.0700

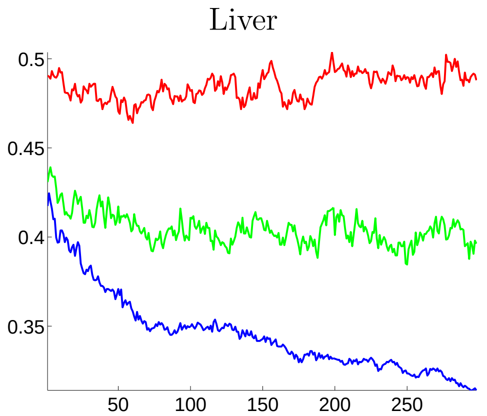


	Average	Final
O-LDC	0.2729	0.2593
Perceptron	0.3375	0.3514
Winnow	0.2926	0.2814

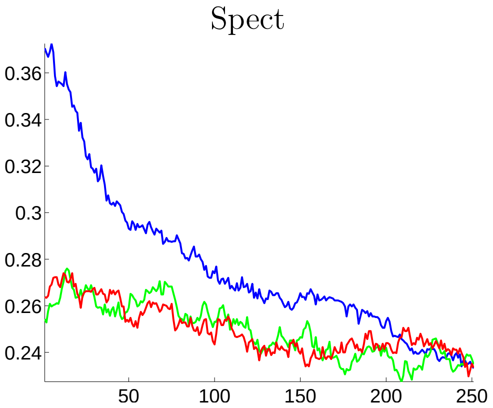
Figure 3.12: Online error plots and tables.



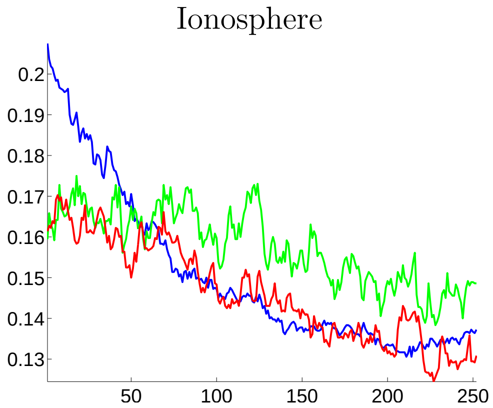
	Average	Final
O-LDC	0.2063	0.1830
Perceptron	0.2506	0.2333
Winnow	0.2532	0.2543



	Average	Final
O-LDC	0.3456	0.3169
Perceptron	0.4030	0.3983
Winnow	0.4861	0.4974

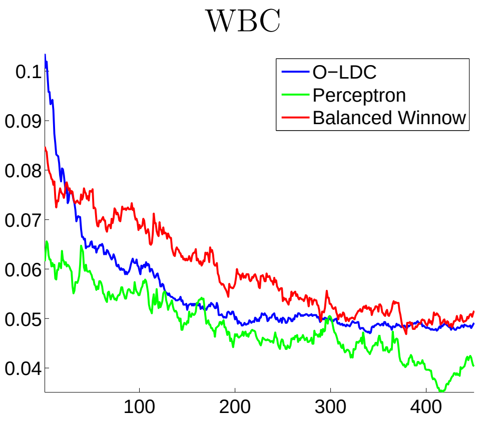


	Average	Final
O-LDC	0.2774	0.2420
Perceptron	0.2466	0.2403
Winnow	0.2416	0.2391

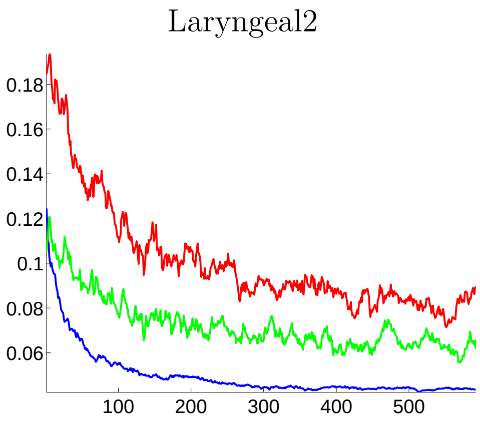


	Average	Final
O-LDC	0.1594	0.1403
Perceptron	0.1754	0.1700
Winnow	0.1576	0.1461

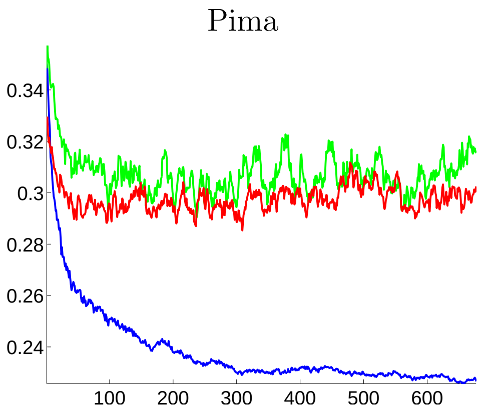
Figure 3.13: Online error plots and tables.



	Average	Final
O-LDC	0.0534	0.0439
Perceptron	0.0523	0.0458
Winnow	0.0593	0.0532

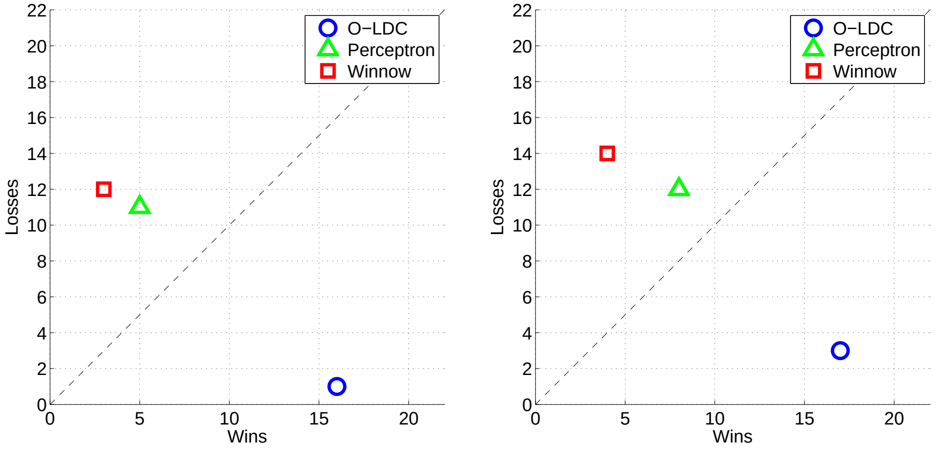


	Average	Final
O-LDC	0.0522	0.0464
Perceptron	0.0733	0.0612
Winnow	0.1039	0.0749



	Average	Final
O-LDC	0.2426	0.2303
Perceptron	0.3087	0.3147
Winnow	0.2982	0.3000

Figure 3.14: Online error plots and tables.



(a) Final error

(b) Average error

Figure 3.15: Wins versus losses for the three algorithms. The diagonal line corresponds to wins = losses.

losses). For the paired t-test on the final error score, the O-LDC scored (16, 1), the perceptron scored (5, 11) and the balanced Winnow scored (3, 12). For the paired t-test on the average error score, the O-LDC scored (17, 3), the perceptron scored (8, 12) and the balanced Winnow scored (4, 14). These results show the O-LDC to have performed significantly better than either the perceptron or balanced winnow, in both terms of final error and average error.

3.5 Semi-supervised Learning

It is generally assumed that true class labels are available throughout the classifier training process. Classifiers can be updated online using these labels. In practice, the true class labels may not be available, beyond an initial training phase. Semi-supervised learning offers techniques where unlabelled data may be used to update the classifier. Learning techniques can be broadly divided into three categories:

Supervised learning During supervised learning true class labels are known. This is the category into which classification traditionally falls.

Unsupervised learning During unsupervised learning labels are not known. The goal is to seek information about the distribution of the data. Clustering is an example of unsupervised learning.

Semi-supervised learning In reality, gathering labelled data can be expensive, time consuming, destructive or dangerous, and very often requires professional expertise [73]. On the other hand, large volumes of unlabelled data are relatively easy to come by. Semi-supervised learning algorithms combine elements of both supervised and unsupervised learning [97, 113]. Many paradigms exist, including co-training [12, 63], link-based classification [84], Gaussian processes [80], expectation maximisation (EM) [59] and naive labelling [73].

In the context of on-line classification for fMRI data, the choice of semi-supervised learning algorithm is restricted by the streaming nature of the data. It is assumed that a small initial labelled data set will be available, followed by a stream of unlabelled data points. Iterative procedures such expectation maximisation are therefore not suitable in the context of on-line classification.

Co-training is a method whereby two classifiers (C_1 and C_2) are trained on the labelled data set, using different subsets of features. The classifiers are used to predict labels for the unlabelled data. The data point and label combinations most confidently predicted by C_1 are used to train C_2 and vice versa. For application to on-line data, it is possible to use co-training in a 'batch' approach where the algorithm is applied after a given number of unlabelled data points have been collected. Whilst the high feature-to-instance ratio of fMRI data makes it well suited to splitting the feature space, this batch approach is infeasible for providing real-time feedback.

By relaxing the requirements of the model, the predicted label of a classifier can be used for updates after each data point. In doing this, the model becomes truly on-line, and can be used for streaming data.

The remainder of this thesis focusses upon *naive-labelling*, whereby the classifier is updated by adding the new data point to the training set and taking the *predicted* label as the true label. This approach should be taken with caution, however, guarding against the possibility of a run-away classifier that progressively learns 'the wrong thing' [26]. A more refined approach to using predicted labels for unlabelled examples is dynamic labelling [44]. In this case, using a similar principle to co-training, out of a pool of unlabelled data, the data point whose label is most confidently predicted is used to update the classifier. The process is iterated until no unlabelled examples remain. This method is only suited however, to cases where the full pool of unlabelled data is immediately available.

3.6 Naive Labelling

Naive labelling is a semi-supervised learning protocol whereby in the absence of ground truth, updates are carried out using the label predicted by the classifier. Without knowing the true class labels, there is a choice of using the predicted labels, or using a fixed, pre-trained classifier throughout. This scenario is particularly relevant for neurofeedback experiments. In these circumstances the fixed classifier will have been trained on a small offline data set which may not be representative of the data set as a whole. A classifier which has been trained offline on a small data set will be likely to show a high error rate. In addition to this, any concept drift will render a fixed classifier useless.

Training a classifier with naive labelling does not come without risk. The classifier may be led astray should updates occur using incorrectly predicted class labels. This may lead to 'run-away' behaviour where the classifier becomes less accurate as training progresses [26, 27]. The likelihood of runaway classifiers is related to the amount of offline training data and on how well the underlying data distribution model is guessed when designing the classifier [73]. It is expected that the lower the amount of training data is, the higher are the chances of a runaway classifier appearing in the ensemble.

3.7 Summary

This chapter has touched upon many areas of machine learning and classification. These areas are far broader and deeper than the scope of this thesis. The techniques and methods discussed here have been selected with the end goal and challenges of classifying streaming fMRI data in mind.

Recall that classification of fMRI data raises challenges such as a limited amount of time to collect pilot data with the participant and a large feature-to-instance ratio. Given these challenges the initial classifier may be of insufficient accuracy. Here it is hypothesised that applying online classification to fMRI data is desirable. As such, simple online linear classifiers have been introduced. These classifiers are capable of processing large volumes of data quickly, with a low risk of overfitting.

Considering neurofeedback type real-time fMRI experiments, there is a realistic possibility that class labels will not be available during the online phase. Concepts such as semi-supervised learning are therefore introduced, specifically naive labelling, which will be applied to fMRI data in Chapters 6 and 7.

Chapter 4

The Random Subspace Ensemble

4.1 Combining Classifiers

A classifier ensemble is made up of several individual member classifiers. The ensemble output can be considered as a decision made by a consensus of experts. The output from the individual classifiers is fed into a 'combiner' and a decision is made. This system is illustrated in Figure 4.1

A classifier ensemble is less sensitive to noise and redundant features than an individual classifier. The problems associated with over-fitting are therefore less prevalent in classifier ensembles than individual classifiers. Classifier ensembles are also deemed to be more accurate than individual classifiers [67].

A good ensemble should be made up of *diverse* classifiers. If all classifiers in

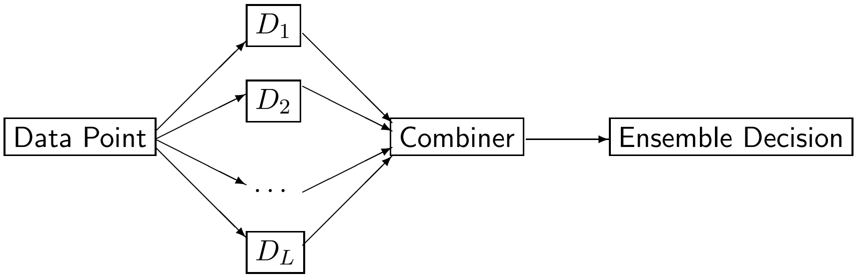


Figure 4.1: Illustration of classifier ensemble.

the ensemble were to be the same, or very similar, then an ensemble would have little or no advantage over an individual classifier in terms of accuracy, and may be computationally more expensive. Diversity, and measures of diversity within classifier ensembles is something to which much attention has been devoted [74]. In this work kappa-error diagrams are adopted as a measure of ensemble diversity, kappa error diagrams are discussed further in Section 6.2.1.

4.1.1 Ensemble Design

When designing a classifier ensemble there are four possible factors to take into consideration [67]. At the data level, member classifiers may be trained on different subsets of the data in order to generate diversity. Another way to generate diversity is to train the member classifiers on different feature subsets. There is also a choice as to which base classifiers to use. Finally, there are different forms of combiner.

Perhaps the most simple and intuitive combiner is the majority vote method. For the majority vote, the class label with the highest number of 'votes' across the individual classifiers is taken to be the ensemble output. Despite its simplicity, in many settings, the majority vote has been found to be equally accurate as other more complicated combiners [78]. The simple majority vote is the combination method used throughout this work. It is noted that for cases of two-class data sets, in order to avoid ties when using the majority vote, it is sensible to have an odd number of classifiers in the ensemble. One major advantage of the majority vote over other methods, is that once the individual classfiers are trained, no further training is required to construct the ensemble.

For ensembles which train individual classifiers on different portions of the data set, or on different feature subsets, several alternatives exist for selecting which data points or features are used. Bagging (**b**ootstrap **agg**regat**ing**) creates training subsets by sampling with replacement from the set of data points (a bootstrap sample) [14].

The consequence of sampling in this way is that a given data point may appear once, more than once or not at all across the individual classifiers in the ensemble.

Boosting algorithms, with AdaBoost [41] being the most well known, also generate diversity by using different training examples. Instead of sampling uniformly with replacement, boosting algorithms maintain a weight vector for the contribution of each training example. At each iteration, the weight vector is updated. Training examples which are misclassified are assigned higher weights. This causes the 'difficult' examples to be featured more often in the training set, and forces their properties to be taken into account, encouraging each new classifier to make different errors to the previous one, thus generating diversity. Variants of boosting algorithms have been used in a wide range of settings from credit card fraud detection (algorithm AdaCost) [19] to optical character recognition [3].

Another algorithm, the Random Subspace (RS) method, trains ensemble members on different *feature* subsets, selected at random. This method is described in more detail in Section 4.2. Skurincha and Duin [116], compare the performance of bagging, boosting and the RS ensemble with different base classifiers for two artificial and five real data sets. The usefulness of each ensemble method was shown to be dependent upon the training sample parameters and the base classifier chosen.

More recent additions to the field of classifier ensembles include the random oracle [70], spherical oracle [109] and rotation forest [110].

Classifier ensembles have many applications, typically in areas where an individual classifier is prone to high error. These include face recognition, remote sensing and medicine, each with their own challenges. For example, challenges presented by remote sensing include huge volumes of data, with a large number of features. Typical challenges tackled by classifier ensembles include those where there is too much data, too little data, or too little of a specific type of data [98], for example when there are very few instances from one class compared to another.

For working with fMRI data, one of the greatest challenges is the feature-to-instance ratio. Choosing an ensemble which trains on subsets of instances would only exacerbate this challenge. As such, an ensemble which reduces the dimensionality of the data set is the most appopriate. Whilst other approaches such as principal component analysis also reduce dimensionality, in doing so, they transform the feature space. The advantage of the random subspace approach, is that the output can be directly related back to the location of the voxels within the brain.

Thus the remainder of this work focuses on the RS ensemble, which is described in more detail below.

4.2 Random Subspace Ensemble

In general, when performing classification, the more features that are available, the better the resulting classifier. It is however possible to 'overfit' the classifier on the training set, especially in data sets with a high feature-to-instance ratio. The Random Subspace ensemble (RS), introduced by Ho [52], is a classifier ensemble whereby ensemble members are trained on feature subsets rather than the entire feature set. The reduction in the dimensionality of the feature set while retaining the number of training data points makes RS ensembles particularly suitable for data sets with a large feature-to-instance ratio.

The RS ensemble requires two parameters. These correspond to the number of classifiers in the ensemble, L, and the cardinality of the feature subsets, M. Define $\mathbf{X} = \{x_1, \ldots, x_n\}$ to be the total set of n features. To create an RS ensemble, L feature subsets of size $M < n$ are generated by drawing at random without replacement from a uniform distribution over $\mathbf{X}$. Each of these L subsets makes up the feature set for one of the L classifiers. The L member classifiers are trained and tested using the respective M features.

The RS ensemble framework has been used with a variety of base classifiers in a number of settings. In the original paper, Ho tests a random subspace ensemble with decision tree classifiers against a single decision tree, and against ensembles generated through bagging and bootstrapping [52]. Whilst the RS method received close competition from bagging and bootstrapping for some data sets, Ho concluded by saying that "the method is expected to be good for recognition tasks involving many redundant features". Sun et al, [122], compare ensemble methods, including the RS ensemble, using EEG data, and found that the best choice of ensemble was dependent upon the base classifier and choice parameters.

The RS ensemble has been shown to be highly effective in other data sets with large feature-to-instance ratios. Micro-array data, like fMRI, suffers from high dimensionality of the feature space. Coupled with a low number of training examples, problems with overfitting arise. The RS ensemble with linear SVM provides an elegant and accurate solution [7, 8].

RS ensembles and variations thereof, have also been applied to face recognition [20, 126, 135, 136]. The algorithm is believed to be of benefit due to the "inherent sparsity and small sample size of data" [20]. This is a similar set of challenges to the ones faced with fMRI data. In the face recognition study conducted by Zhu et al [136], a variation termed semi-random subspace is used. The initial feature space is broken down into several local regions, the RS approach is applied to each region and the base classifiers are combined to make a final decision. Two combination approaches were tested, firstly taking all the classifiers in parallel and combining the outputs to make a final decision. The second approach is the hierarchical approach, generating an intermediate decision for each local region. These local decisions are then combined for the final ensemble output. By capturing any spatial relationships between features in the local regions the semi-RS method was shown to improve upon the results of the RS method for this task. In fMRI, the voxels which respond to a given stimulus are highly distributed throughout the whole brain. Whilst the semi-RS method was shown to be successful for face recognition, selecting and separating regions of the brain in this manner may break important relationships and not capture the full response of the brain.

The RS ensemble approach has also been adopted by Polikar et al in an algorithm for handling missing features [105].

fMRI data poses a severe classification challenge because of the extremely large feature-to-instance ratio. The RS ensemble therefore seems a natural choice to apply to fMRI data, especially as the algorithm is computationally inexpensive due to the reduced number of features per ensemble member. Kuncheva and Rodriguez [71], compare eighteen classifier methods for fMRI data. The experiments were conducted with a variety of voxel selection methods and parameters. The RS ensemble with SVM was shown to perform best across the trials conducted.

A random subspace based technique has also been used on fMRI data for brain mapping [9,117,118], and as another application, Richiardi et al [108] use an ensemble of decision trees to classify fMRI connectivity graphs[1].

For data sets with a large number of irrelevant features, the RS ensemble has been used in conjunction with an initial feature selection step. Bertoni et al [7] use this technique for microarray data. The feature selection step is used in order to remove noisy and irrelevant, uninformative genes from the analysis. The aim is to improve the accuracy of the classifiers within the ensemble in order to further improve the ensemble performance. The authors use a univariate significance based feature selection approach, and acknowledge that any feature selection algorithm can be used.

There is a lack of clarity about the number of features, which ought be selected as 'relevant' or 'statistically significant'. One approach is to set, instead of a predefined number of voxels, a threshold on the significance level of a univariate statistic. This

[1] An fMRI connectivity graph describes the relationships between brain regions across a time series, as the participants are subjected to different stimuli [108].

may be a value such as $\alpha = 0.05$, or a corrected value. Correction approaches include the Bonferroni correction for multiple comparisons or the false discovery rate [99]. Dependent upon the choice of feature selection method, and the choice of correction applied, there may be a dramatic difference in the number of voxels selected [99]. This is best illustrated by an example. Taking the EN1 data set, there are 81,840 voxels. Applying an ANOVA to the un-masked data set[2], 43,332 voxels are found to be significant at a significance level of $\alpha = 0.05$.

The Bonferonni correction takes into account the number of statistics being calculated, and the family-wise error. If n test statistics are to be drawn from a distribution with probability α of being greater than a threshold, the probability of *all* statistics being less than the threshold is $(1 - \alpha)^n$. The family-wise error rate is the probability that one or more values are greater than α, $1 - (1 - \alpha)^n$, which for small α can be approximated as $n\alpha$ [42]. Applying the Bonferonni correction, the significance threshold required can be re-calculated as $\frac{\alpha}{n}$.

For EN1, applying the Bonferroni correction reduces the number of significant voxels to 11,446. For the three class data set, EB, there are 86,400 voxels. In the same procedure, using an ANOVA, 7,806 voxels are found to be significant at $p = 0.05$. Using a corrected p-value, this drops to 1 voxel. For consistency across data sets, the approach used for this thesis is to set a predefined number of voxels, K, rather than define a significance level.

4.3 Deriving Parameters for the RS Ensemble for fMRI Analysis

In comparison with many other ensemble frameworks, the RS ensemble has an advantage of requiring only two parameters, the number of classifiers in the ensemble, L, and the number of features in each feature subset, M. There is very little guidance as

[2] As this is for illustration purposes only, the ANOVA was applied to the entire data set.

to how best to select values for these parameters. Based upon a notion of 'important' features, a theoretical approach to derive values for L and M is proposed[3].

Due to the large number of voxels in a typical fMRI scan, important discriminative information may be contained in a relatively small number of voxels. An assumption is made that there are Q of these 'important' voxels, contained in a set $\mathcal{I} = \{q_1, \ldots, q_Q\}$, $\mathcal{I} \subset X$, where $\mathbf{X}$ is the set of all voxels, $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ and $|\mathcal{I}| = Q << n$, and the remaining $n - Q$ voxels are random noise. Also, it is assumed that the cardinality of the subspace M is much smaller than n. Recommended values for L and M are sought in terms of n and Q, based upon the theory that accurate and diverse individual classifiers are a prerequisite for better ensembles [15, 16, 67].

As the accuracy of the classifiers within the ensemble is not known a priori, properties of the feature subsets are used as potential indicators. Three such properties; usability, coverage and feature set diversity are introduced.

Usability A classifier built upon random noise alone, will be no more accurate than random chance. It is therefore of interest to have at least one $q \in \mathcal{I}$ in each feature subset. A classifier is defined as being *usable* if its feature subset contains at least one 'important' voxel $q \in \mathcal{I}$.

To calculate the probability of drawing a feature subset that represents a usable classifier, take Y to be the number of 'important' features in a subset of size M, drawn without replacement from X. Y is a random variable which has a hypergeometric distribution with probability mass function

$$P(Y = i) = \frac{\binom{Q}{i}\binom{n-Q}{M-i}}{\binom{n}{M}}, \quad i = 0, 1, \ldots, Q.$$

Thus the probability of drawing a usable classifier is

$$P(\text{usable classifier}) = 1 - P(Y = 0) = 1 - \frac{\binom{n-Q}{M}}{\binom{n}{M}}$$

[3] This work was conducted in collaboration with L.I.Kuncheva and J.J.Rodriguez [69, 72]

The usability of an ensemble is defined as the proportion of classifiers from L which are defined as being usable. Define a completely usable ensemble as an ensemble where all L member classifiers are usable. As feature subsets are sampled independently, the probability of a completely usable ensemble is

$$P(U=1) = P(\text{usable classifier})^L = \left(1 - \frac{\binom{n-Q}{M}}{\binom{n}{M}}\right)^L .$$

This can be simplified as

$$P(U=1) = \left(1 - \prod_{i=0}^{M-1}\left(1 - \frac{Q}{n-i}\right)\right)^L .$$

Since it is assumed that $M << n$, the equation can be further simplified to

$$P(U=1) \approx \left(1 - \left(1 - \frac{Q}{n}\right)^M\right)^L .$$

This approximation is equivalent to replacing the hypergeometric distribution with a binomial distribution. Given the size of n for fMRI data, it can be said that sampling *with* replacement is approximately equivalent to sampling *without* replacement. Y can therefore be approximated with a binomial distribution with parameters M and $p = \frac{Q}{n}$. The probability of a completely usable classifier in this case would be $1 - \left(1 - \frac{Q}{n}\right)^M$.

For a given parameter set (L, M, n and Q), the expected degree of usability of the ensemble, $E[U]$, can be calculated. Denote by Z a random variable which expresses the number of usable classifiers in the ensemble. Then Z has a hypergeometric distribution. Define the *total* as the number of possible samples, without replacement, of size M from X, that is, $\binom{n}{M}$. The number of *usable* classifiers is calculated by taking the number of non-usable classifiers, $\binom{n-Q}{M}$, from the total. The number of *selected* classifiers at a time is L. The expected value of Z is $\frac{\text{Selected}\times\text{Usable}}{\text{Total}}$, therefore the expected usability of the ensemble is

$E[U] = \frac{1}{L}E[Z]$

$$E[U] = \frac{1}{L} \times L \times \left(1 - \frac{\binom{n-Q}{M}}{\binom{n}{M}}\right) = 1 - \frac{\binom{n-Q}{M}}{\binom{n}{M}}.$$

The expected usability of the ensemble does not depend on the ensemble size L. It is hypothesized that ensembles with a higher degree of usability will be more accurate.

Coverage In order to best use the available information for classification, it is desirable to include as many $q \in \mathcal{I}$ in the feature subsets as possible. Ideally, each $q \in \mathcal{I}$ to be selected at least once in the L samples of M features. The *degree of coverage of the ensemble*, C, is defined as the proportion of features $q \in \mathcal{I}$ (out of Q) which are selected for one or more of the L classifiers.

In order to calculate the degree of coverage, as the feature subsets are sampled independently from X, the binomial approximation to the hypergeometric distribution can be used once again. The probability of selecting a given $q \in \mathcal{I}$ in sample of size M, can be calculated as $\frac{M}{n}$. Conversely, the probability that q is not selected is $1 - \frac{M}{n}$. The probability that q is not selected in any of the L feature subsets, denoted $P(\bar{q})$, is $P(\bar{q}) = \left(1 - \frac{M}{n}\right)^L$. The probability of q being selected in at least one of the L feature subsets is therefore $1 - P(\bar{q})$.

The probability of all features being covered is

$$P(\text{Complete coverage}) = P(C = 1) = \left(1 - \left(1 - \frac{M}{n}\right)^L\right)^Q.$$

Denote by Z the number of covered features out of Q. Z has binomial distribution with parameters Q and $p = 1 - \left(1 - \frac{M}{n}\right)^L$. The expected coverage is

$$E[C] = \frac{1}{Q}\left(1 - \left(1 - \frac{M}{n}\right)^L\right)Q = 1 - \left(1 - \frac{M}{n}\right)^L.$$

The expected coverage depends on the ensemble size L and the subset size M but not on Q. The assumption here is that the higher the degree of coverage, the more accurate the ensemble.

For fixed Q and n, $E[U]$ is monotonically increasing with M and $E[C]$ increases with both L and M. This suggests that a larger ensemble with a larger feature sample is best. Note that in the extreme, the case of $M = n$, the ensemble will be made up of classifiers built upon identical subsets. This defeats the object in having an ensemble in the first place. It is also likely, that with an excessive feature-to-instance ratio, the classifiers may overfit the data. A third property, termed feature set diversity, is therefore introduced.

Feature set diversity In order to maintain a level of diversity within the ensemble, it is important that the feature subsets used to construct the L classifiers are non-identical. Given the assumption that $n - Q$ voxels are random noise, and do not contribute to the classification. Feature set diversity is calculated based upon the contribution of the Q important features.

Denote by $S_1, S_2, \ldots, S_L$ the L feature subsets sampled from X. Consider $S_1, S_2 \subset X$ such that $|S_1| = |S_2| = M$. Denote by $I_1 \subseteq \mathcal{I}$ and $I_2 \subseteq \mathcal{I}$ the respective subsets of 'important' features in S_1 and S_2 respectively. Feature Set Diversity (D) is defined as

$$D(S_1, S_2) = |I_1 \cup I_2| - |I_1 \cap I_2|.$$

Two classifiers are *non-identical* if their feature subsets differ by at least one 'important' voxel. Each feature $q \in \mathcal{I}$ may or may not contribute to D. A value of 1 will be added if q is in either set but not in both. Then the expected diversity for any pair of subsets S_1 and S_2 is

$$E[D] = \sum_{i=1}^{Q} P(q_i \in I_1)P(q_i \notin I_2) + P(q_i \notin I_1)P(q_i \in I_2).$$

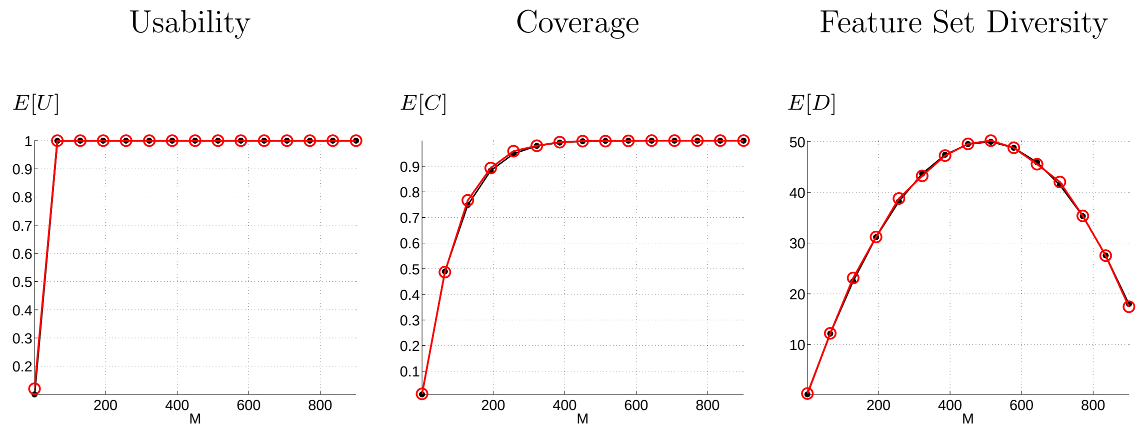


Figure 4.2: Theoretical (red, circles) and simulation curves (black, dots) coincide for the expected values of U, C and D for $n = 1000$, $Q = 100$ and $L = 10$. The empirical curve is calculated as an average of 10 ensembles with randomly sampled $L = 10$ sets of M features.

> Since all features in $\mathcal{I}$ have equal chance of being selected in a subset of size M, and the subsets are drawn independently,
>
> $$E[D] = 2Q\frac{M}{n}\left(1 - \frac{M}{n}\right). \qquad (4.1)$$
>
> This calculation disregards non-usable classifiers. An ensemble can be still be diverse even if it contains non-usable classifiers for which $I_1 = I_2 = \emptyset$.

The expected theoretical value of $E[U]$. $E[C]$ and $E[D]$ are calculated for $n = 1000$, $Q = 100$ and $L = 10$. Simulations, calculated as an average of 10 ensembles are run with $L = 10$ randomly sampled sets of M features. The resulting theoretical and empirical curves, shown in Figure 4.2, are seen to coincide. Varying L had little effect in the shape or position of the curve. Based upon these results we claim that values of M close to $\frac{n}{2}$ are optimal as all three criteria reach their maxima.

Chapter 5

Random Subspace Ensemble for Non-i.i.d. Streaming fMRI Data

5.1 A Simulation Experiment

Having defined the three criteria and derived suggested values for L and M, a simulation is used to check the relationship between the criteria and the accuracy of the RS ensemble for fMRI type data.

5.1.1 Data

In order to illustrate and simulate the theory, a synthetic data set with realistic properties is generated. This allows control of n and Q. This synthetic 2 class data set is based upon the first two classes of the Bangor 2 data set. To create the data set, the Contrast to Noise Ratio (CNR) for each voxel of the real fMRI data set (Bangor 2: Classes 1 and 2) is calculated. The CNR of each voxel is then used to rank the voxels. The means and covariance matrix are stored for the top Q voxels. These correspond to the Q important features in the data. Multivariate Gaussian distributions were simulated for each class. The remaining $n - Q$ features were simulated as random noise, with mean zero and standard deviation equal to the mean CNR for the Q important features.

5.1.2 Experimental Protocol

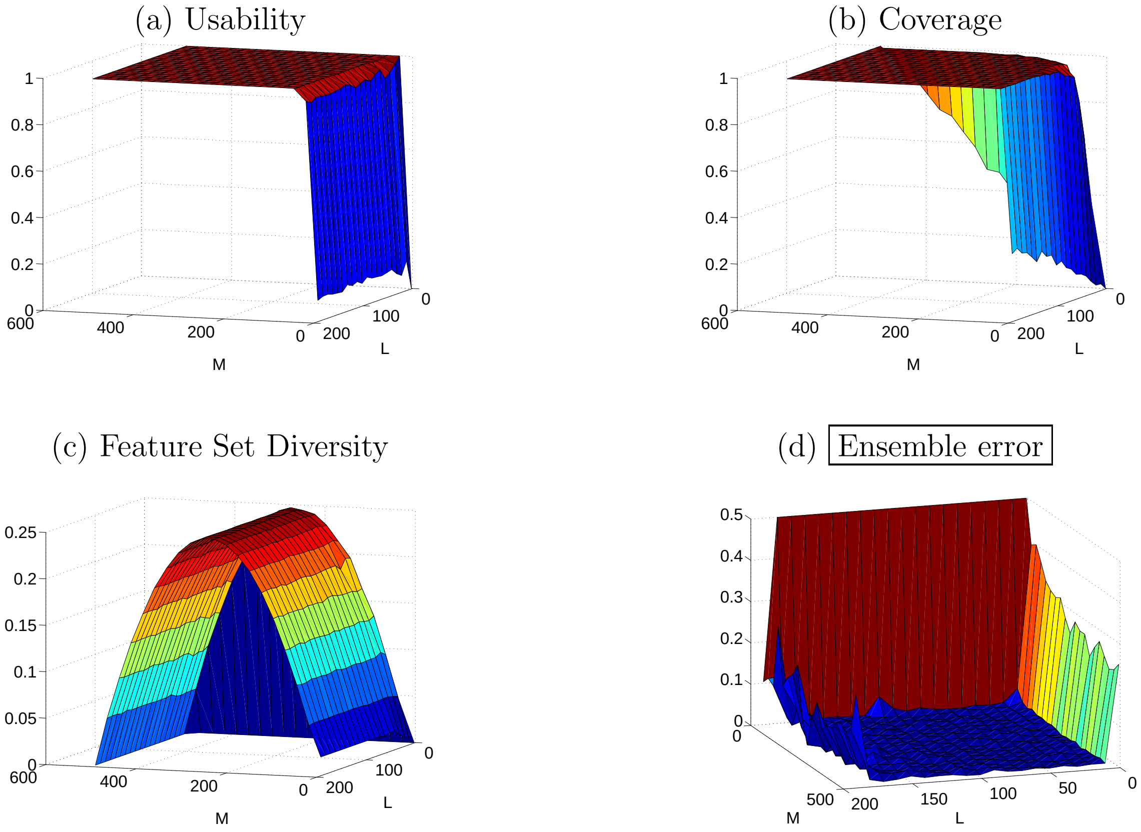
For each parameter combination (M, L, Q, n), detailed below, 10 data sets were generated. Each data set had 10 training examples per class (total 20 training examples) and 100 testing examples per class (total 200 testing examples). This ratio was chosen to reflect that of real fMRI data sets. Data sets were generated with total number of features, $n = \{200, 500, 1000\}$, number of important features, Q, such that $\frac{Q}{n} = \{0.02, 0.05, 0.1, 0.25, 0.5, 1\}$. For the ensemble parameters, M took 20 evenly spaced values from 1 to n, L took evenly spaced values between 1 and 200.

As a base classifier, an SVM was used. For each parameter combination the error of the RS ensemble is calculated, along with the observed values for usability, U, coverage, C and feature set diversity D for the ensemble.

5.1.3 Results

From the results of the simulations surfaces for U, C, D are obtained along with the error of the ensembles. These are visualised on a (L, M) grid. Figure 5.1 shows an example of the surfaces for the distribution where $n = 500$ and $Q = 50$. Each point is calculated as an average across 10 simulations, with data being drawn independently from the chosen simulated distribution. The shapes of the surfaces were consistent across all six ratios of $\frac{Q}{n}$. The shape of the surfaces confirms the hypothesis that larger values of U, C and D lead to more accurate ensembles.

From subplot (a) it can be seen, that as expected, usability does not depend upon L. U quickly raises to 1 as M increases. Feature set diversity, subplot (c), produces a 'tent'-shaped surface. Once again, the values of D depend upon M, but not on L. The largest values of D are achieved for $M \approx \frac{n}{2}$. Coverage, as seen in subplot (b), has a value of 1 for the largest part of the grid. C is the only property which depends upon L. The plot suggests that small to medium sized values of L are sufficient. As



(a) Usability

(b) Coverage

(c) Feature Set Diversity

(d) Ensemble error

Figure 5.1: The three RS characteristics and the ensemble error as functions of the ensemble size L and the feature set size M. Each of the 2 classes in the data set was sampled from a Gaussian distributions with $Q = 50$ relevant and $n - Q = 450$ noise features.

a rule of thumb, it is therefore suggested to use $M = \frac{n}{2}$ and $L = \frac{n}{10}$ for fMRI data. The observed error is shown in subplot (d).

Table 5.1 summarises the simulation results. The table shows the $\frac{Q}{n}$ ratio, the average error rate of the RS ensemble over the whole (L, M) grid, denoted $\bar{E}$, as well as the error using the *recommended* values, denoted $\bar{E}^*$. $\bar{E}$ is seen to be greater than $\bar{E}^*$ across all values of $\frac{Q}{n}$. The table also shows the correlation coefficients between the RS ensemble error E, on one hand, and U, C, and D, on the other. These coefficients support the hypothesis that large values of usability, coverage and feature-set diversity are beneficial for the ensemble.

Table 5.1: Summary of the simulation results. $\bar{E}$ is the average RS ensemble error across the (L, M) grid. $\bar{E}^*$ is the value of the ensemble error for the recommended parameter values, $M = \frac{n}{2}$ and $L = \frac{n}{10}$.

					Correlation with E		
$\frac{Q}{n}$ ratio	$\bar{E}$	$\bar{E}^*$	Range of E	$\sigma(E)$	Usability	Coverage	Diversity
0.02	7.99	1.25	0.70–50	16.43	−0.513	−0.873	−0.649
0.05	5.80	1.75	0.80–50	12.62	−0.602	−0.865	−0.566
0.10	5.73	1.40	0.60–50	11.83	−0.467	−0.791	−0.628
0.25	6.97	0.90	0.35–50	14.51	−0.174	−0.746	−0.645
0.50	5.47	2.15	0.35–50	10.94	−0.035	−0.540	−0.579
1.00	8.58	1.05	0.25–50	11.90	N/A	−0.367	−0.543

5.1.4 Experiment with Real fMRI Data

The RS ensemble with SVM was run on classes 1 and 2 (faces and places) of the Bangor 2 data set. Firstly, $k = 1000$ voxels were pre-selected using the SVM method. Three-fold cross-validation was then applied to test the RS ensemble. The ensemble was tested on a 10×10 grid of values for M and L. M was varied from 1 to k at equal intervals, and L was varied from 1 to $k/5$. The ensemble error is illustrated as a surface over the (L, M) grid in Figure 5.2. The recommended values of $M = 500$ and $L = 100$ are marked as lines across the 3-D surface. The lines intersect near the minimum of the error surface, confirming empirically the recommendation for L and M. The average error across the whole grid was 0.2138, the error at $M = 500$ and $L = 100$ was 0.0521.

5.2 Online Classifier Ensembles for fMRI Data

Before applying naive labelling to streaming fMRI data, supervised online learning is considered. Three online linear classifiers are tested on the EN1 data set. The classifier models chosen are the perceptron, winnow and online linear discriminant classifier (O-LDC). The questions of interest are

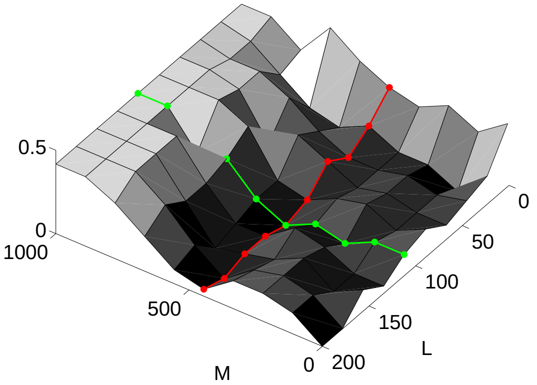


Figure 5.2: RS error on the real fMRI data set as a function of the ensemble size L and the feature size M. The recommended values for L and M are overlayed on the surface.

1. Are random subspace ensembles more accurate than individual classifiers for fMRI data?
2. Does online learning benefit the classifiers and/or the ensembles?
3. Which individual classifier is best suited for (supervised) classification of streaming fMRI data?

5.2.1 Method

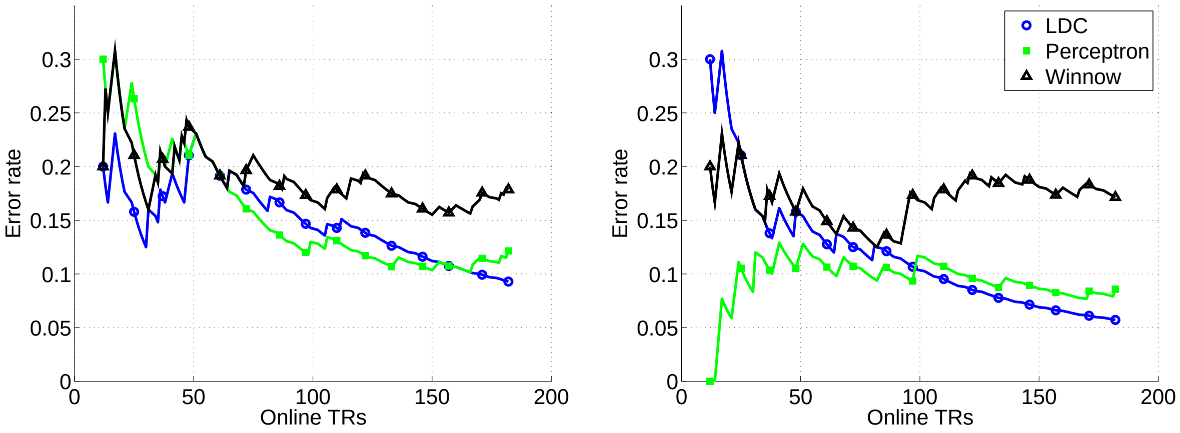
For this experiment the EN1 data set used as it is a 2-class problem, which suits the perceptron and winnow. The data set was labelled using the standard box-car method. The data set is prepared by taking an offline training data set, T consisting of the first blocks of each presentation type, that is, the first block of emotion, and first block of rest, of cardinality $N_T = 17$. The remaining 187 data points, in sequence, make up the online data set, S.

Experiment 1: Individual Classifiers For each base classifier (perceptron, winnow and O-LDC), an offline (batch) version is trained on T, using class labels 1 and 2 to correspond to emotion and fixation respectively. This may be regarded as an initialization period in an fMRI experiment. After this initial training period, the online data points are presented one at a time. As each data point is presented, the 'current' classifier is tested; if the data point is misclassified then the classifier is updated.

Experiment 2: Random Subspace Ensemble For each base classifier an RS ensemble is trained on T. Based upon the $M = n/2$ recommendation, $L = 11$ and $M = 1000$ are chosen as the ensemble parameters. The online data points are presented one at a time, and the individual classifiers within the ensemble are tested. Each individual classifier which incorrectly classifies the data point is updated. The accuracy of the ensemble is recorded.

5.2.2 Results

The cumulative error is calculated at each streaming data point in S, using Equation 3.8. The error progressions for the individual classifiers and RS ensembles are illustrated in Figure 5.3 (a) and (b) respectively.



(a) Individual classifiers

(b) Ensemble classifiers

Figure 5.3: Cumulative error progression for online data.

The O-LDC outperforms the other classifiers both individually and as an ensemble, which is seen by the end point of the progression showing a lower error score.

The perceptron comes second whilst the winnow yields the worst results. The error progression for the perceptron appears to start at zero as the perceptron correctly classifies the first few data points in S. Table 5.2 summarises the final cumulative error scores, at $TR = 187$. The results show that for all base classifiers, the RS ensembles perform better than the individual classifiers. The O-LDC is seen to improve dramatically during the online learning phase, and is the most accurate base classifier in both experiments. This reinforces the results from the i.i.d. example, which shows the O-LDC to be better than the perceptron and winnow.

Table 5.2: Final errors for individual classifiers and classifier ensembles (%), taken at $TR = 187$.

	O-LDC	Perceptron	Winnow
Individual classifier	9.29	12.14	17.86
Classifier ensemble	5.71	8.57	17.14

Kappa-error Diagrams

As the online instances are presented and the classifiers in the ensemble are updated, the mean pairwise error and kappa diversity scores change, creating a trajectory. Figure 5.4 plots the trajectories of the means of the kappa-error clouds for the RS ensembles over time, one trajectory for each base classifier model. The endpoint of each trajectory is indicated with a marker. Take the RS ensemble of the perceptron as an example, represented by a green trajectory and red square marker in Figure 5.4. Following offline training, the pairwise error of the ensemble is $e \approx 0.21$, the corresponding kappa diversity score is $\kappa \approx 0.3$. In the case of the perceptron, the general trend of the trajectory is downwards (indicating decreasing pairwise error) and to the right (indicating decreasing diversity). The final pairwise error for the RS ensemble of perceptron classifiers is $e \approx 0.125$, with $\kappa \approx 0.7$.

The initial high diversity of the perceptron may explain its early accuracy. This diversity decreases over time. Only the trajectory of the O-LDC ensemble shows an improvement of both diversity and accuracy over time.

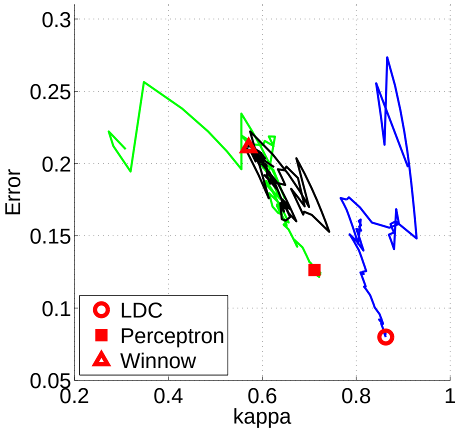


Figure 5.4: Trajectory of means of kappa error diagrams.

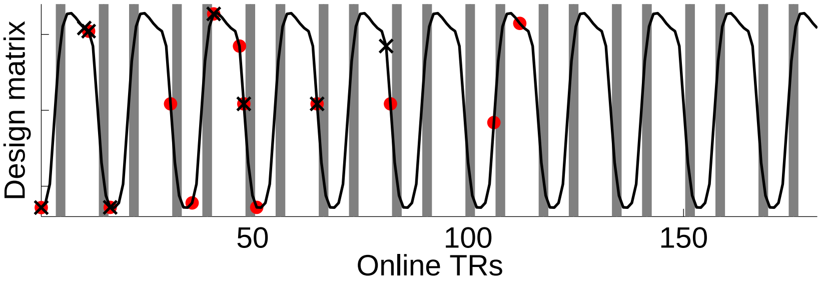


Figure 5.5: Design matrix highlighting occurrence of individual and ensemble errors, plotted against the design matrix (black line). Individual errors represented by red dot, ensemble errors by black cross. Transitions between classes are illustrated by vertical grey stripes.

To compare the performance of the ensemble and individual classifiers, it is interesting to see *when* classification errors occurred. Figure 5.5 shows the 'design matrix' corresponding to expected levels of neural activity. Peaks correspond to negative emotion and valleys to rest (no emotion). Transition TRs, between the classes, are marked by grey vertical stripes. These TRs are where errors in classification are expected to be made. Errors by the individual O-LDC are marked by red dots whilst black x's mark errors made by the O-LDC ensemble. For both classifiers, errors are predominantly made in the first half of the experiment. This is a strong indication that the classifiers improve over time. This supports evidence from the error progressions in Figures 5.3(a) and (b). In Figure 5.3(a), the error progression individual O-LDC classifier is seen to make errors beyond TR = 100, whilst in Figure 5.3(b) the ensemble is seen to stop making errors at approximately TR=80. In addition to this, from Figure 5.5 it can be seen where the errors occur in terms of the design of the experiment. Fewer errors occur at peaks and valleys than in the transition periods, with those errors that do occur in peaks and valleys appearing early in training. Errors made by the ensemble are seen to occur less frequently than those made by the individual classifier. The ensemble is also seen to stop making errors sooner than the individual classifier.

5.3 Processing Time

The O-LDC has been shown to perform more accurately than either the Perceptron or Winnow. Also, the classifier ensembles have been shown to outperform the individual classifiers. However in order to perform in real-time the updates need to be sufficiently fast, it is important that the classifier ensemble can update within the required amount of time ($\approx$ 1TR). Experiments were conducted in order to test the processing times for training and updating the O-LDC.

5.3.1 Method

This experiment used the EN1 data set. A stratified sample of 9 data points per class was taken as an initial training sample. A grey matter mask was calculated and applied, reducing the feature space from 83,072 voxels to 28,118. The time taken to calculate the mask was 0.49s[1]. The time taken to perform two voxel selection

[1] Experiments were repeated 20 times, with mean times being reported. Experiments were carried out using Matlab [87] on a laptop with an AMD Turion 64 x2 2GHz processor and 2GB of memory.

Table 5.3: Update times (s) for individual classifiers

K	Initial training time	Update time
20	0.0029	0.0016
50	0.0050	0.0018
100	0.0119	0.0026
250	0.0811	0.0117
500	0.5745	0.0790
1000	3.6419	0.5015
2000	27.1080	3.3411

methods were compared, the ANOVA and maximum activation method. To select 2000 voxels from the 28,118, the ANOVA took 70.33s whilst the maximum activation method took only 0.04s.

5.3.2 Results

Table 5.3 shows the processing time for training and updates for an individual classifier, with different numbers of features, K. Update times include time taken to select the voxels (from the grey matter mask and ANOVA) and normalise the data point, using the coefficients derived from the training data. Table 5.4 shows the initial training times for classifier ensembles for different combinations of L and M. Table 5.5 shows the update times for the ensembles.

It can be seen that above $K = 250$ features the individual classifier slows dramatically. For the ensemble, even with $L = 13$ and $M = 250$ the updates occur well within the required time ($\approx$ 2s), and it has already been shown that in this setting, the ensemble outperforms the its individual component classifiers. This supports the hypothesis that the ensemble is the better choice for real time classification.

5.4 Discussion

The experiments show that for streaming fMRI data the random subspace ensemble performs better than the individual online linear classifiers. This is based upon

Table 5.4: Training times (s) for classifier ensembles

	$L = 5$	$L = 9$	$L = 11$	$L = 13$
$M = 20$	0.0102	0.0135	0.0166	0.0188
$M = 50$	0.0182	0.0309	0.0366	0.0438
$M = 100$	0.0568	0.0989	0.1210	0.1449
$M = 250$	0.4054	0.7340	0.8977	1.0650

Table 5.5: Update times (s) for classifier ensembles

	$L = 5$	$L = 9$	$L = 11$	$L = 13$
$M = 20$	0.0083	0.0154	0.0197	0.0215
$M = 50$	0.0094	0.0171	0.0207	0.0235
$M = 100$	0.0131	0.0247	0.0298	0.0356
$M = 250$	0.0626	0.1088	0.1325	0.1586

comparison between the mean performance of the individual classifiers and the mean performance of the classifier ensembles. It may be argued that certain individual classifiers perform better than the ensemble, however it is not possible to establish in advance which individual classifiers these would be, hence the ensemble offers a more robust output.

Across both the individual and ensemble experiments, the online linear discriminant classifier (O-LDC) is seen to be more accurate than either the perceptron or the winnow. As a linear classifier, the O-LDC is fast to train and has demonstrated accurate results. The O-LDC is therefore the best choice of the base classifiers tested here, for use in online pattern classification studies of the human brain.

For this study it has been assumed that the true class labels are known during the online training phase. The situation when this is not the case is considered in Chapters 6 and 7.

Chapter 6

Classification of i.i.d. fMRI Streaming Data

A study is proposed to determine whether a classifier benefits from naive labelling when the data comes as a series of fMRI brain volume images. Whilst a run-away classifier is a realistic adversity when naive labelling is used, it is hypothesized that by using an RS ensemble, a sufficient number of classifiers *within the ensemble* will be improved beyond their offline accuracy, and thus the ensemble will counteract any adverse effects on an individual ensemble member.

6.1 Experimental Protocol

Three data sets are used, EN1, EN2 and EB. The data sets are labelled using a shifted box-car method as described in Section 2.5, where the labels are shifted by 1 TR (corresponding to an offset of one data point). This means that the number of instances in the experiment is 1 less than in the original data descriptions. For each data set, a voxel mask is derived, based upon the BOLD signal, using the method outlined in Section 2.7.1. This mask is applied to reduce the feature set, predominantly to the grey matter of the brain. The resulting data sets are summarised in Table 6.1.

The data points from each data set were split into two subsets: T, a data set used for offline (batch) training, and S, a data set which is prepared and presented

Table 6.1: Summary of data sets after applying the grey mask

Name	Voxels (Features)	Data Points	Classes
EN1	28426	203	2
EN2	28662	203	2
EB	29865	108	3

to the classifier as online (streaming) data. The initial data points were shuffled and sampled to form T. It is noted that this breaks the autocorrelation of the fMRI signal, however in order to explore semi-supervised learning for streaming fMRI data, the method first needs to be shown to work for stationary, independent and identically distributed (i.i.d.) data. This issue is discussed later in relation to presenting the online data stream.

Having selected T, the remaining data points are oversampled to construct S with 500 data points[1]. This is the closest approach to constructing i.i.d. sets. In order to reduce the dimensionality of the feature set, following the recommended procedure by De Martino et al. [30], a fixed amount (K), of voxels are pre-selected. This is achieved by taking the K voxels with maximum activation, based on T. Both training and testing data are normalised, using the mean and standard deviations calculated for T.

A 'fixed' offline random subspace ensemble is trained on T alone. Three scenarios are considered:

Scenario A: No Updates (Fixed). The online data points from S are presented to the ensemble one at a time. The ensemble is *not updated* during the online phase. The cumulative error is measured in order to compare whether using naive labelling is better than the no-action scenario.

[1] The 500 data points are sampled independently from the remaining data points. Data points may appear in S once, more than once, or not at all.

Scenario B: Supervised Updates (Supervised). The true class labels are assumed to be immediately available. As each online data point is presented to the ensemble, each ensemble member is updated using the true class label.

Scenario C: Unsupervised Updates (Naive). The true class labels are assumed to be unavailable. As each online data point is presented to the ensemble, each ensemble member is re-trained using its individual *predicted* label as the true label.

The experiments were conducted across a range of parameters; number of pre-selected voxels $K = 500$, ensemble size $L = [5, 9, 11]$ and feature set cardinality $M = [20, 50, 100]$. The cardinality of the training sets were $N_T = [20, 40, 100]$. Due to the random nature of the feature selection for the RS ensemble, experiments were repeated 50 times, and the results were averaged.

For each scenario, the error rates of the individual member classifiers which make up the ensembles are also considered.

Computational costs of the preprocessing and classification are not assessed quantitatively here. This is not expected to be a major obstacle, however, given that the classifiers have relatively low computational costs, and that earlier studies have demonstrated the feasibility of real-time classification [37, 53, 54, 77], and processing speed of computers is increasing over time.

The study seeks to answer the following questions:

Individual vs ensemble. For classification of unlabelled fMRI data, does an individual classifier or classifiers in an ensemble framework yield better results? In line with previous and existing research, classifier ensembles are expected to have higher accuracy than an individual classifier. This may not be true if the individual classifiers deteriorate progressively. At some point the ensemble will become worse than the average individual classifier.

Fixed vs untrained updates. For streaming fMRI data, is it advantageous to update the ensemble using naive labels?

6.2 Results

The cumulative error progression is calculated for each time step using Equation 3.8. The 'final' errors for the three data sets, taken at time $t = 500$ are summarised as colour plots in Figure 6.1, full error values can be found in Tables A.1, A.2 and A.3. For each combination of M and N_T, ensembles of $L = [5, 9, 11]$ classifiers are generated, giving a total of 25 individual classifiers. The individual error is taken as the mean error of these 25 classifiers at time $t = 500$. Each column of the table represents a value of L, with the last column showing the mean error of the individual classifiers, titled 'I'. Each row of the table corresponds to a value of N_T. Within each coloured grid, rows correspond to values of M, and columns to the three RS ensemble methods, Fixed (F), Naive (N) and Supervised (S).

As expected, the supervised classifier is superior to the fixed and naive classifiers. The final error scores at $t = 500$ are compared for the fixed ensemble and the naive ensemble in order to see which scenario works best for unlabelled data. The results of this comparison are summarised in Table 6.2. A '+' indicates that the naive ensemble performs better than the fixed ensemble. A '$-$' indicates the naive ensemble performs worse than the fixed ensemble. Significance was calculated using a paired t-test, uncorrected for multiple comparisons. Significant results at $\alpha = 0.05$ are indicated by $\oplus$ and $\ominus$.

For these parameters the results suggest that the naive ensemble is on a par or better than the fixed ensemble (21 $\oplus$, 23 +, 35$-$ and 4 $\ominus$). For $M \geq 50$, the naive ensemble performs much better than the fixed ensemble, (20 $\oplus$, 19 +, 13 $-$ and 2 $\ominus$). Data set EB was the most 'difficult' for the classifiers, as this is where different

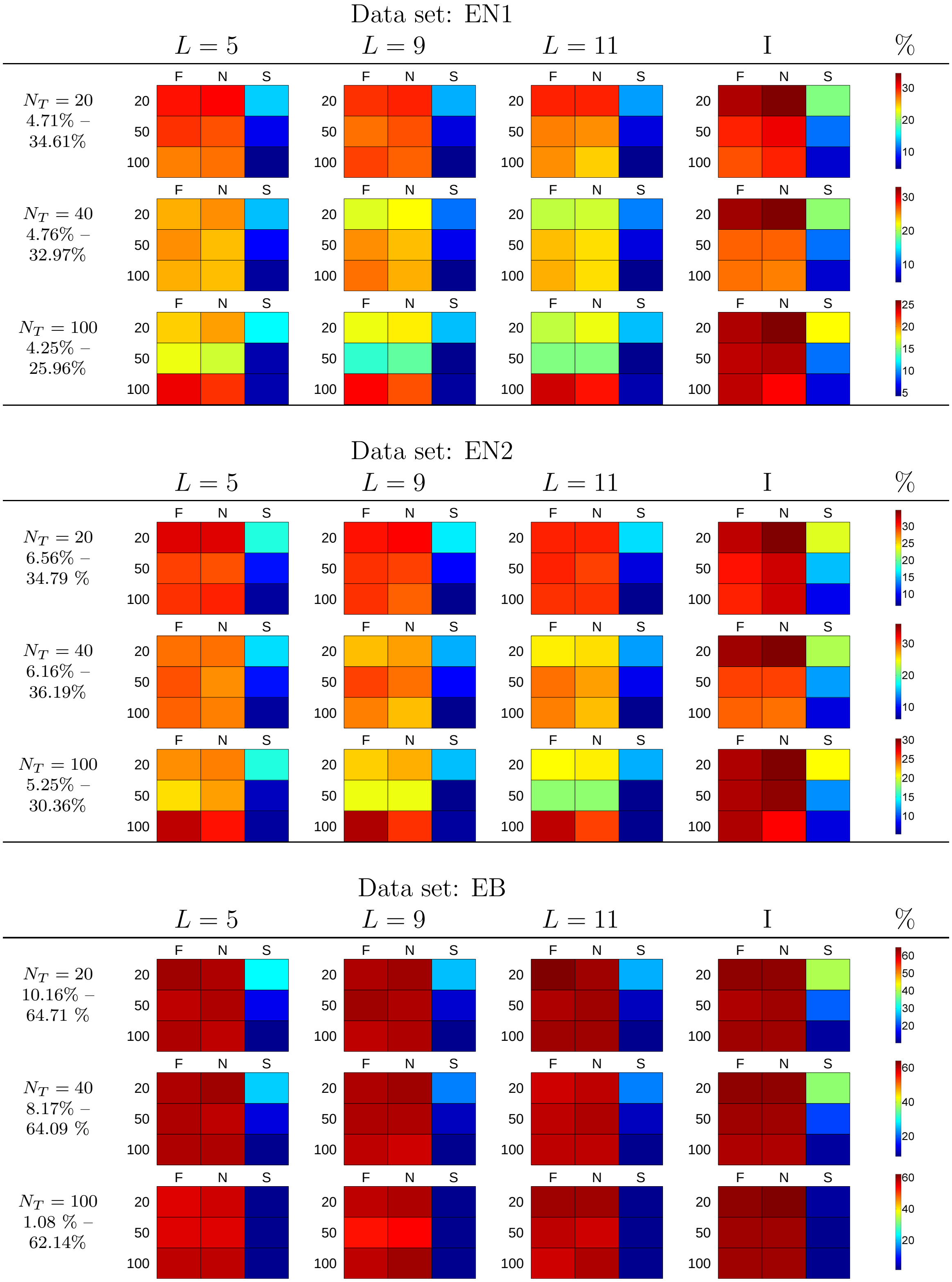


Figure 6.1: Final cumulative error scores (%). Error scores are coloured from blue through to red representing low and high error respectively. Values of M are shown as rows of each coloured grid. 'F', 'N' and 'S' correspond to fixed, naive and supervised ensembles. 'I' corresponds to mean individual error of classifiers for a given M and N_T.

Table 6.2: Direct comparison of fixed and naive ensembles. '+' and '-' respectively, represent a 'win' or 'loss' by the naive ensemble. A circle surrounding the + or - indicates that the result is statistically significant at significance level $\alpha = 0.05$. L is ensemble size, M is cardinality of feature subsets, N_T is cardinality of training data set.

		EN1			EN2			EB		
	N_T	20	40	100	20	40	100	20	40	100
	M									
$L = 5$	20	-	-	⊖	-	-	-	+	-	-
	50	+	⊕	+	+	⊕	⊖	-	+	+
	100	-	+	⊕	-	⊕	⊕	+	+	+
$L = 9$	20	-	-	-	-	-	-	-	-	-
	50	-	⊕	⊖	+	⊕	-	+	+	-
	100	+	⊕	⊕	⊕	⊕	⊕	-	+	-
$L = 11$	20	+	-	⊖	+	-	-	⊕	-	+
	50	+	⊕	-	⊕	⊕	+	-	-	+
	100	⊕	⊕	⊕	+	⊕	⊕	-	+	-

emotions are being recognised. The other challenge with this data set is the addition of a third class.

6.2.1 Kappa-error Diagrams

Kappa-error diagrams are now an accepted tool for comparing classifier ensembles [86]. Each pair of classifiers in the ensemble generates one point on the diagram. The x-axis of a kappa-error diagram is the diversity of the pair, κ. Lower values of κ indicate higher diversity. Kappa measures the level of agreement between the classifiers while correcting for chance [40]. Consider a testing set of N examples, the pairwise κ is defined as follows

$$\kappa = \frac{2(N^{11}N^{00} - N^{01}N^{10})}{(N^{11} + N^{10})(N^{10} + N^{00}) + (N^{11} + N^{01})(N^{01} + N^{00})}, \tag{6.1}$$

where N^{11} is the number of testing examples on which both classifiers are correct, where N^{00} is the number on which both classifiers are wrong, N^{10} is the number on which classifier 1 is correct and classifier 2 is wrong, and N^{01} is the number where classifier 1 is wrong and classifier 2 is correct; $N^{00} + N^{11} + N^{01} + N^{10} = N$.

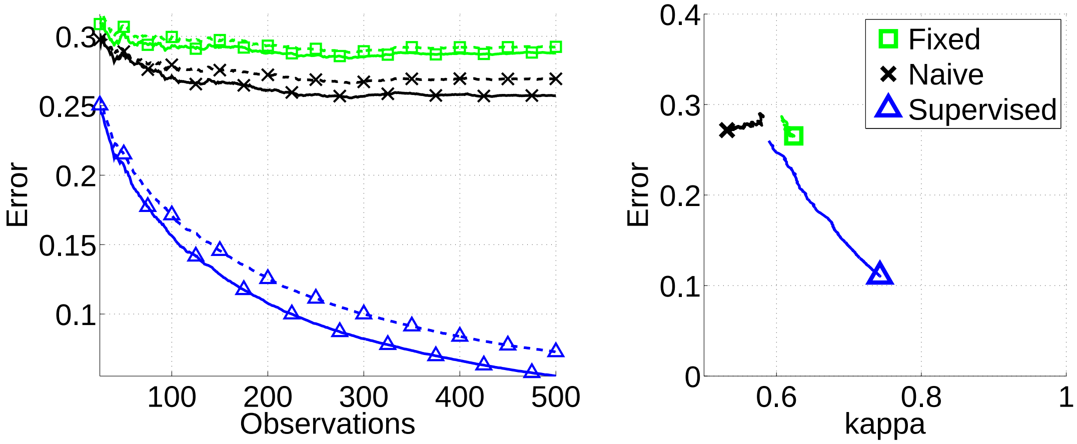
In a kappa-error diagram, the y-axis shows the averaged error rate of the pair. Each ensemble can be plotted as a cloud of points in a kappa-error diagram. Ensembles whose ‘clouds’ of points are situated closer to the bottom left corner of the diagram are usually preferable as they display high pair-wise accuracy and high diversity.

For the experiments considered here, there is one classifier ensemble for every time point t. It is interesting to see how the cloud shape and position of the kappa-error diagram changes with time. For each TR the kappa error and pairwise accuracies are calculated. To demonstrate how the kappa-error diagrams progress with time, the mean of the kappa-error clouds are calculated at each TR. Instead of plotting the entire clouds of points, it was decided to plot the trajectories of the centres.

In order to understand the mechanism of improvement through naive labelling ensembles, the progression of the error over time and the corresponding time-trajectory on the kappa-error diagram are considered. Figure 6.2 (a) shows the error plot for EB2 with $L = 11$, $M = 100$ and $N_T = 100$. The plot is taken from $t = 25$ onwards, as at low t there are large fluctuations in the cumulative error leading to the plot appearing noisy and unstable. If the plot was to be taken from $t = 0$, then the plots for all scenarios would start from one point, as the same offline ensemble is used in each case. The marker and line colour indicate the base classifier, a solid line indicates the classifier ensemble whilst a dashed line indicates the individual classifier.

The error rate of the fixed classifier is expected to remain constant over time. The error rate of the supervised classifier will drop as the classifier sees more data. It is hoped that the naive classifier follow the same pattern as the supervised classifier, in that the error will drop as t increases, thus showing the naive labelling strategy to be beneficial.

The figure shows the dashed lines, representing the individual classifiers, above



(a)

(b)

Figure 6.2: Figures for EB data taken with ensemble size $L = 11$, feature set cardinality $M = 100$ and training data set cardinality $N_T = 100$.
(a) Error progression. Solid line indicates classifier ensembles. Dotted line indicates individual classifiers. Plot illustrates changes in error over time.
(b) Kappa-error progression. Kappa-error progression plots the changes in pair-wise accuracy and diversity as the classifier ensembles learn over time.

the corresponding solid lines. This indicates that the classifier ensembles outperform the individual classifiers. The error of the supervised ensemble is seen to drop over time, whilst the error of the fixed classifier ensemble remains constant. The naive ensemble is seen to improve over time, with significantly better results than the fixed ensemble.

Figure 6.2 (b) shows the kappa error trajectories corresponding to EB2 with $L = 11$, $M = 100$ and $N_T = 100$. The cloud for the fixed ensemble is expected to float about the initial point, as the only difference from one time point to the next will be the estimate of kappa and the individual errors. The classifier and the ensemble parameters do not change, hence the movement will be only a small fluctuation. The supervised ensemble, on the other hand is expected to drop down the plot, indicating that the individual accuracies improve with more data being seen. It is curious how the diversity of the ensemble progresses, i.e., whether the cloud will move to the left (larger diversity) or right. The endpoint of each trajectory is indicated with a marker.

A good classifier ensemble will be both accurate and diverse, and thus appear near the bottom left hand corner of the diagram. As the trajectories of the ensembles are plotted over time, if the ensemble improves, it is expected that the trajectory will progress towards the bottom left corner.

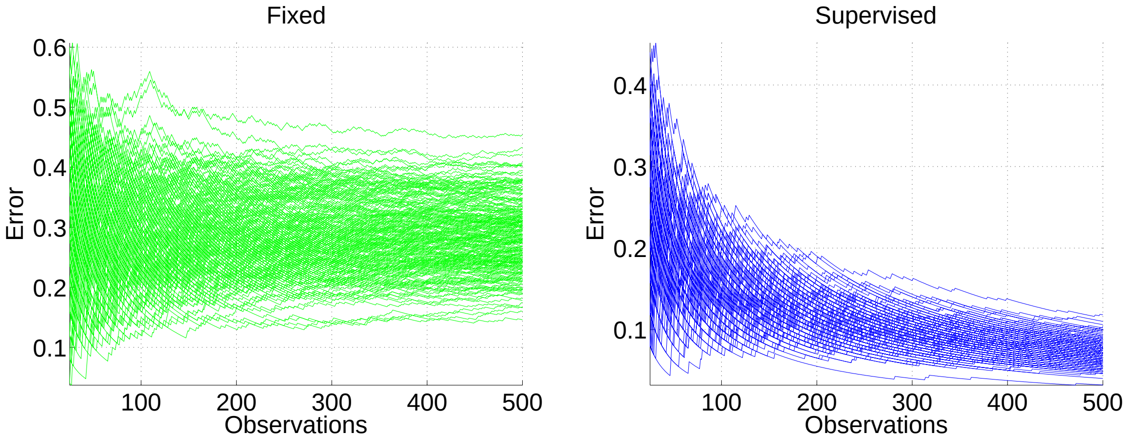
The trajectory of the supervised ensemble tracks down as accuracy increases. The diversity decreases slightly, this suggests that all classifiers within the ensemble are being driven towards the optimal classifier, thus are becoming more similar. The fixed ensemble, as expected, shows very little progression. The naive ensemble shows an increase in both accuracy and diversity, both of which are desirable characteristics.

To better see what is happening the individual progressions of the classifiers are plotted. Figures 6.3 (a) and (b) show the typical patterns of the individual classifiers for the fixed and supervised ensembles respectively: the error of the fixed classifiers remains constant while the error of the supervised classifiers drops over time.

For the naive classifiers, it is desirable to see a similar shape to that of the supervised classifiers. Figures 6.4 (a) and (b) show two cases of the patterns of the individual naive classifiers. In both cases the naive ensemble performed significantly better than the fixed ensemble. Figure 6.4 (a) is a case where the naive classifiers show a desirable learning pattern, improving over time. In Figure 6.4 (b) some classifiers are seen to display runaway behaviour. What is interesting in this case is that the naive ensemble still performs better than the fixed ensemble, indicating that the ensemble environment counteracts the runaway behaviour.

6.2.2 Individual vs Ensemble

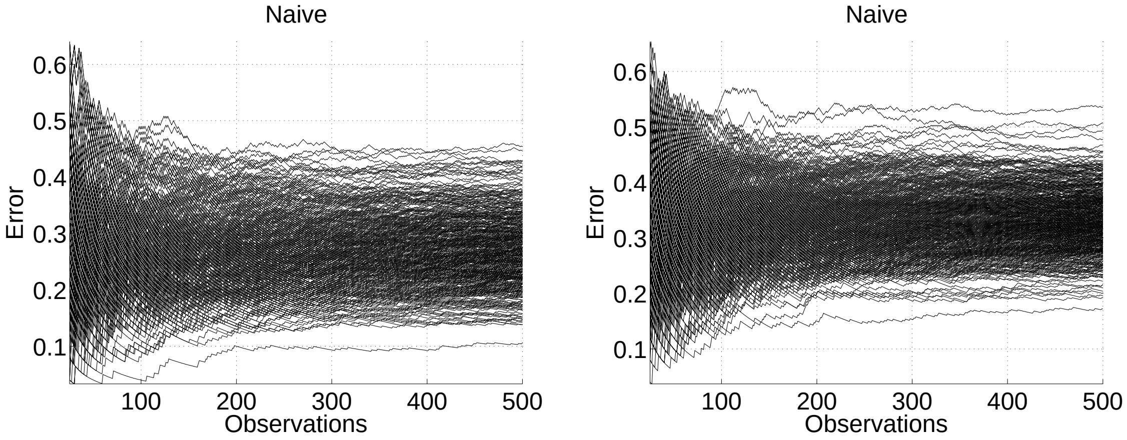
From Figure 6.1 the error of the individual classifiers is compared with the error of the ensembles. The error rate for the ensembles can be seen to be lower. This supports the hypothesis that a classifier ensemble is more accurate than an individual classifier. In Figure 6.2 (a) the error progression of the individual classifiers can be directly



(a) Fixed classifiers.

(b) Supervised classifiers.

Figure 6.3: Typical error progression of individual classifiers (data set: EN1, size of ensemble: $L = 5$, cardinality of feature subsets: $M = 100$, cardinality of training data set: $N_T = 20$).



(a) EN1: $L = 9$, $M = 100$, $N_T = 40$.

(b) EN2: $L = 11$, $M = 50$, $N_T = 20$.

Figure 6.4: Comparison of different individual classifier progressions for the naive ensemble. Improvement over the fixed ensemble was obtained with both ensembles.

compared with the classifier ensemble. The classifier ensembles are seen to be more accurate than their individual counterparts.

6.2.3 Fixed vs Unsupervised Updates

From Table 6.2 the results from the fixed classifier can be directly compared with those of the naive ensemble. For the correct parameters, the naive ensemble with unsupervised updates is beneficial to the ensemble. Specifically, when the cardinality of the feature set is $M \geq 50$ the naive ensemble performs better than the fixed ensemble. The method is tested on two 2-class data sets and one 3-class data set. The method performs best for the 2-class data sets, when the accuracy of the initial fixed ensemble is higher.

6.3 Discussion

It has been shown that classifier ensembles are more accurate than individual classifiers for streaming i.i.d. fMRI data. The experiments also show that given an appropriate choice of parameters, classifiers updating using the naive labelling strategy perform well within an ensemble framework. It was shown that for sufficient training data, a naive classifier ensemble performs significantly better than a fixed, pre-trained classifier ensemble.

During a real time fMRI experiment, there is the potential for concept drift. An online classifier working in this environment is required to be capable of updating and adapting during the course of the experiment. Naive labelling offers an intuitive solution to this problem.

In these experiments the data is treated as being i.i.d., which is not strictly the case for fMRI in general. This approach serves as a first step towards semi-supervised learning for streaming fMRI data. The non-i.i.d. case raises new questions. Autocorrelations and the non-stationary nature of streaming fMRI data may weight and ‘pull’ an online classifier in a certain direction, encouraging runaway traits. In the next chapter, it is shown how naive labelling combined with the random subspace ensemble may be applied to streaming fMRI data in order to simulate a real-time scenario.

Chapter 7

Semi-supervised Classification of Non-i.i.d. fMRI Streaming Data

7.1 Introduction

Up to this point, online classifiers for fMRI have been considered in cases where data was simulated as i.i.d. data, or where true class labels are known. During fMRI experiments there may be concept drift. This can be attributed to head motion, physiological changes or low-frequency scanner drift. Recall that, as stated by LaConte, [76], future applications of fMRI analysis may consider cases where changes in patterns are *expected*. Experiments involving performance enhancement, rehabilitation or therapy expect the brain response to change over time, with trials being conducted weeks, months or even years apart. In these cases, pre-trained supervised classifiers will become less relevant and there is a need for a classifier which adapts with the data as it trains over time. This is identified as one of the current challenges in fMRI classification.

It has already been shown that with the correct parameter tuning, an ensemble framework may constrain the potential negative behaviour of a classifier with naive updates. As a next step, the theory is applied to streaming data, and options are considered for handling changes within the data, that is, when there is both concept drift *and* unlabelled data.

The situation becomes a catch twenty two. A fixed pre-trained classifier may be used, which is known to be inaccurate due to concept drift, or there is the gamble with naive labelling, which may improve the classifier, but runs the risk of making it worse. Here an alternative method is proposed for using naive labelling within an ensemble framework. Consider *guided updates*: the ensemble prediction is taken to be the 'true' label and is used to update each member classifier instead of its own predicted label. In a related work, [82] also use ensemble labels to boost accuracy in semi-supervised learning, in an offline co-training approach.

7.2 Guided Update Strategy

Intuitively, the predicted label from a classifier ensemble is likely to be more accurate than the predicted label from an individual classifier within the ensemble. It is hypothesised that by using the ensemble decision to update the individual ensemble members, the likelihood of runaway classifiers is reduced. The ensemble with 'guided' updates is expected to perform better than an ensemble where its members are updated using their individual predicted labels.

7.2.1 Theory and Illustrations

Before applying the guided update strategy to streaming fMRI data, the hypothesis is first tested theoretically and with a simulated i.i.d. case.

Consider an ensemble of L classifiers. The ensemble receives a sequence of N i.i.d. data points whose class labels are unknown. If the classifiers in the ensemble are not updated throughout the online run, the ensemble at data point N will be equally accurate as the starting ensemble. Updating the classifiers can improve ensemble accuracy.

Two update strategies can be employed, both within the naive labelling approach.

Individual Update. The member classifiers are re-trained by augmenting the training data with the current observation and the label proposed by *the classifier* as the true label.

Ensemble Update. The member classifiers are re-trained by augmenting the training data with the current observation and the label proposed by *the ensemble* as the true label.

The individual update can be regarded as a Markov chain where each processed data point is a step in the chain. Denote the initial accuracy of the classifier by p. Assume that, if a correct label is used for the update, the accuracy increases to $p+\epsilon$, and if incorrect label is used, the accuracy decreases to $p-\epsilon$, where ϵ is a small positive constant. The transition matrix for the update step is

	After the update	
Before the update	wrong	correct
wrong	$1-p_t+\epsilon$	$p_t-\epsilon$
correct	$1-p_t-\epsilon$	$p_t+\epsilon$

Note that the accuracy is tagged by t, the time step. The transition matrix contains the current accuracy p_t which varies from one step to the next. Thus the Markov chain is non-homogeneous, and asymptotic distributions are not readily available.

The probability for correct classification at step $t+1$ can be calculated from the transition matrix

$$p_{t+1} = p_t(p_t+\epsilon) + (1-p_t)(p_t-\epsilon) = p_t + \epsilon(2p_t-1). \tag{7.1}$$

If the classifier is better than chance at the start ($p > 0.5$), the accuracy is expected to increase progressively with t. For the individual update method, the majority vote accuracy does not play a role in the update. Assuming independent classifiers, the majority vote accuracy will increase with the increase of p.

For the guided update, the probability for correct classification *of the individual classifier* at step $t+1$ depends on the ensemble accuracy, P_{ens} in addition to p_t

$$p_{t+1} = P_{\text{ens}}(p_t + \epsilon) + (1 - P_{\text{ens}})(p_t - \epsilon) = p_t + \epsilon(2P_{\text{ens}} - 1). \tag{7.2}$$

Since for independent individual classifiers $P_{\text{ens}} > p_t$, the improvement in the individual accuracy will be better for the ensemble updates.

Monte Carlo simulations were carried out to illustrate the behaviour of the two types of updates compared to the fixed (not updated) classifier. 1000 independent runs were performed for each of the two update strategies, and for the fixed ensemble (no updates). The following protocol was used:

- L random numbers between 0.5 and 0.6 were generated as the initial accuracies of the classifiers in the ensemble.
- 500 steps of online update were performed.
- At each step, L random numbers were sampled uniformly from the interval $[0, 1]$ and compared with the current classification accuracies to obtain 'correct' and 'wrong' classifications for each classifier.
- Majority vote accuracy of the ensemble was calculated and stored.
- The accuracy of each classifier was updated according to the respective strategy
    - $p_{t+1} = p_t$ for the fixed ensemble strategy.
    - $p_{t+1} = p_t + \epsilon$ if the classifier was 'correct' at step t and $p_{t+1} = p_t - \epsilon$ if the classifier was 'incorrect', for the individual update strategy ($\epsilon = 0.001$).
    - $p_{t+1} = p_t + \epsilon$ if the *ensemble* was 'correct' at step t and $p_{t+1} = p_t - \epsilon$ if the *ensemble* was 'incorrect', for the guided update strategy ($\epsilon = 0.001$).

The ensemble accuracies were averaged across the 1000 runs, producing curves with 500 consecutive online accuracies. Figure 7.1 plots the three curves together with the predicted majority vote curves. The majority vote accuracy at step t was calculated under the assumption of independent classifiers using

$$P_{\text{ens}} = \sum_{i=\lceil L/2 \rceil}^{L} p_t^i \, (1 - p_t)^{L-i}.$$

The individual accuracies p_t were calculated iteratively, starting from $p_0 \;=\; 0.55$ (the expected value of the initial accuracies), and using updates as in equations 7.1 and 7.2.

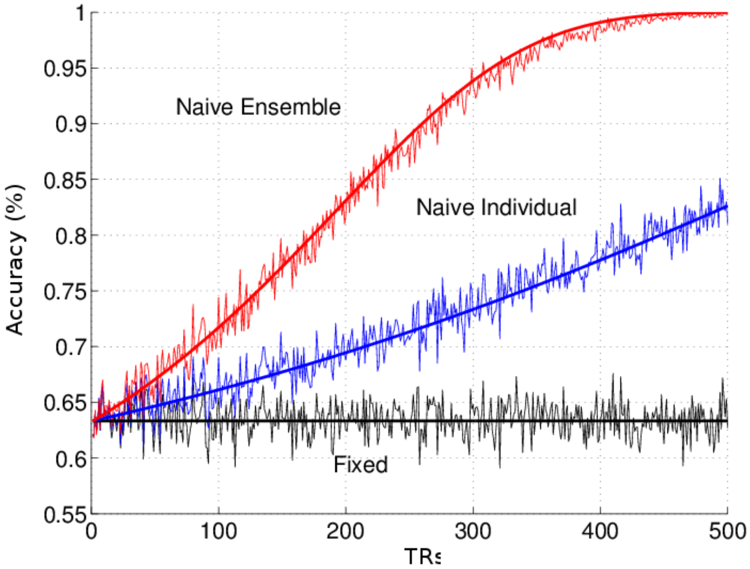


Figure 7.1: Simulation and theoretical results for the fixed ensemble and the two update strategies.

The figure shows that the ensemble-update leads to the best results, followed by the single update. The assumptions that the data is i.i.d, the classifiers are independent, and the updates lead to improvement (however small) if the correct label is used, cannot be guaranteed in practice. Intuitive as they are, caution should be exercised. Naive labelling has been shown to have mixed effect on the classification accuracy depending on the classifier model, and even on the initial parameter guesses [68].

Given the non-i.i.d. nature of fMRI data, a lesser difference will be seen between the two update strategies, however this simulation serves as proof of concept and supports the hypothesis. Having shown that the ensemble update strategy is capable of improving upon the individual update strategy for i.i.d. data, the concept is applied to streaming fMRI data.

7.2.2 Protocol

The experiment is carried out on data set EN1. Class labels were assigned using the box-car approach, shifted by a single TR. The data sets are prepared by taking, as an offline training sample, T, the first 17 instances of the data set. These instances correspond to the first blocks of stimuli presented from each class. Volume masks are derived and applied to each data set. This is the same approach as in Section 6.1, and the data set summary in Table 6.1 applies here also. An ANOVA test is used on T to pre-select a fixed amount, $K = 2000$, of voxels. The remaining 187 data points, in sequence, make up the online training data, S. The data sets T and S are then normalised, using the means and standard deviations calculated for T. Note that this is different to the previous study on naive labelling, where the streaming data set was shuffled and oversampled to form i.i.d. data.

An offline RS ensemble of O-LDC classifiers is trained on T. The same base ensemble is used for each of the update strategies. Data points from S are presented sequentially, with the following procedure being applied:

Fixed strategy Ensemble accuracy is tested on the new data point. No ensemble update is carried out.

Naive strategy Ensemble accuracy is tested on the new data point. Each ensemble member is updated using its individual predicted label.

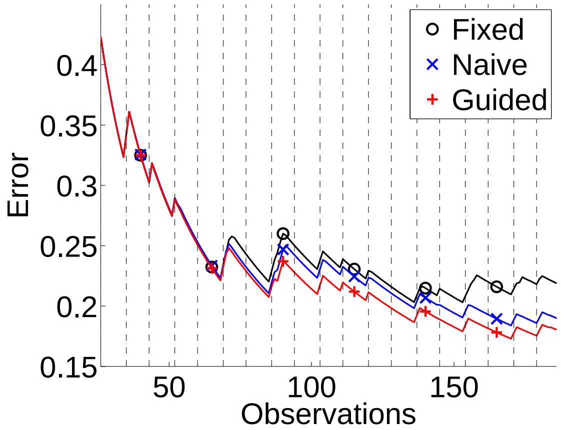


Figure 7.2: Cumulative error progression comparing the three strategies for $L = 13$, $M = 20$. Vertical lines represent class boundaries.

Guided strategy Ensemble accuracy is tested on the new data point. Ensemble members are updated using the *ensemble* decision.

The experiment is repeated for parameters $L = [5, 9, 13]$, $M = [20, 50, 100, 250]$.

7.2.3 Results

For each time-step t the cumulative error is calculated using Equation 3.8. Figure 7.2 plots of the cumulative error scores over time for parameters $L = 13$ and $M = 20$. The vertical lines indicate the class boundaries. The presence of a sequence of multiple instances from the same class (due to the block design of the experiment) can be seen to affect the classifier in that with every class change a small peak is seen in the error level. This peak arises where the classifier sees data points from the 'transition' period, where the true state of the brain is uncertain. The transition is the period when the classifier is expected to make most mistakes. Overall, the trend of the plot is that the error level declines over time, showing that the classifiers learn and adapt with the data. The ensembles follow a similar learning pattern for the first two thirds of the trial. For the last part, the naive and guided ensembles continue to gradually improve, whilst the error of the fixed ensemble stabilises at around 22.5%.

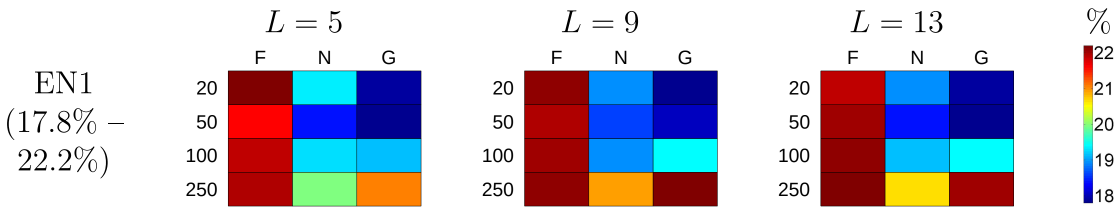


Figure 7.3: Final cumulative error scores (%) taken at $t = 187$. 'F', 'N' and 'G' correspond to fixed, naive, and guided strategies,respectively. The range of the final error scores is 17.8% – 22.2%. Error scores are coloured from blue through to red representing low and high error respectively.

The 'final' error scores, taken at $t = 187$, are illustrated in Figure 7.3, and are available numerically in Table A.4. Each 'box' corresponds to a value of L, whilst each row corresponds to a value of M. Strategies can be compared for a given L, M combination by looking at a row within a box. The range of final error scores across the three strategies was 17.8% – 22.2%.

The results from the strategies are compared for all parameters using a paired t-test with significance $\alpha = 0.05$. Both the final error scores and the average error are compared. For a given strategy, the average error corresponds to the area under the error progression curve for that strategy and thus gives an indication of learning capability. As there are 12 parameter sets and 3 strategies, a total of 36 pairwise comparisons are made. The numbers of wins vs losses are plotted in Figure 7.4. The best point is at 24 wins and no losses, the worst point at 0 wins and 24 losses.

Direct comparison of the naive and guided strategies with the fixed ensemble is offered in Table 7.1. Strategies which perform better than the fixed classifier are indicated by a '+', strategies which perform worse are indicated by a '−'. Significant results (calculated using a paired t-test at $\alpha = 0.05$) are indicated by $\oplus$ and $\ominus$ respectively. From Table 7.1 it can be seen that both the naive and guided ensembles perform significantly better than the fixed ensemble for the vast majority of parameters. The guided ensemble, overall, has a higher number of 'wins' than the naive

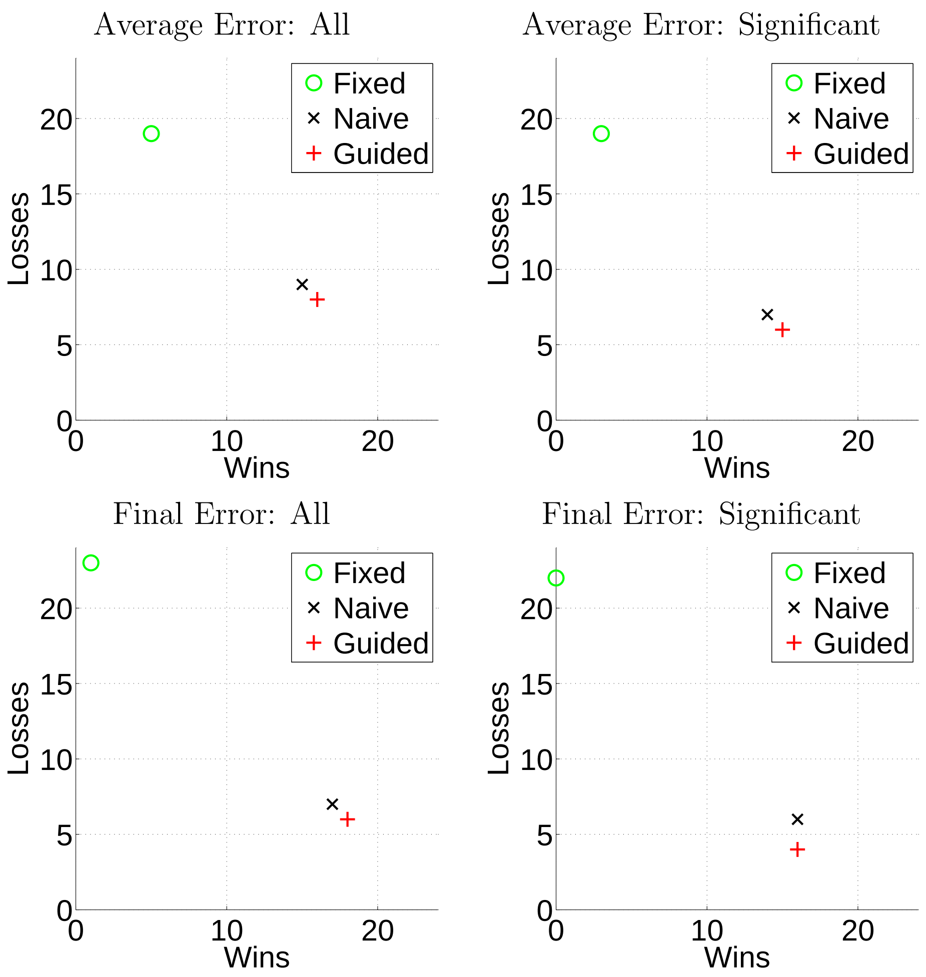


Figure 7.4: Pairwise wins vs losses. Significance calculated at $\alpha = 0.05$.

Table 7.1: Direct comparison of the naive and guided strategies with the fixed ensemble. Strategies which perform better than the fixed classifier are indicated by a '+', strategies which perform worse are indicated by a '−'. Significant results are indicated by $\oplus$ and $\ominus$ respectively.

	$L = 5$		$L = 9$		$L = 13$	
	Naive	Guided	Naive	Guided	Naive	Guided
$M = 20$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$M = 50$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$M = 100$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
$M = 250$	$\oplus$	$\oplus$	$\oplus$	-	$\oplus$	$+$

ensemble across these parameter settings. This indicates that using the ensemble decision to update the classifiers is beneficial. Making use of the higher accuracy of the ensemble decision constrains potential runaway behaviour in individual ensemble members, which in turn leads to a more accurate ensemble.

7.2.4 Discussion

Real-time fMRI classification faces the challenge of unlabelled data and concept drift. This study proposes a solution in the form of classifier ensembles. The solutions have been tested and illustrated on streaming fMRI data. The experiments show that the ensembles benefit from updating during the online phase. Both update strategies perform significantly better than the fixed strategy across a variety of parameters.

The guided update strategy offers a possible solution to the combination of unlabelled data and concept drift. Results from the update strategy compare well with the standard naive classifier ensemble, and perform significantly better than the fixed classifier ensemble. The guided update strategy is shown to have a lower error score than either the fixed or naive update strategies in seven of the twelve parameter combinations tested. From the cumulative plot it can be seen that the majority of errors occur during transition periods. In order to prevent the classifiers learning ‘the wrong thing’ during these periods, introduced below are further criteria, offering different scenarios under which the ensemble may be updated.

7.3 Error-driven and Confidence-driven Updates

Two update criteria are considered: The *confidence* and *error*. By using confidence and error to regulate the updates, three further guided update strategies are generated:

Error Driven Classifiers within the ensemble whose individual predicted label does not agree with the ensemble decision are updated using the ensemble decision.

Confidence Driven The confidence with which the ensemble has made its prediction is calculated. Denote by y the outputted (predicted) label of the ensemble, and by y_i the predicted label of the component classifiers. Confidence is calculated as

$$\text{confidence} = \frac{\sum_{i=1}^{L}\{y_i = y\}}{L}$$

Classifiers within the ensemble are updated using the ensemble decision, when the confidence in the predicted label is above a threshold. For data sets of two classes this threshold may be between 50% and 100%, here 75% is chosen, as the mid-point of the available range.

Error and Confidence Driven Classifiers within the ensemble whose predicted label does not agree with the ensemble decision are updated using the ensemble decision, when the confidence in the predicted label is above a threshold.

7.3.1 Experimental Protocol

Experiments are carried out for the two single run, two-class 'emotion detection' fMRI data sets, EN1 and EN2, using the same protocol as in Section 7.2.2. For this experiment, the parameters of the RS ensemble are chosen as $L = \{5, 9, 11\}$ and $M = \{20, 50, 100, 250\}$.

In the online phase of the experiment, the following strategies are considered:

Naive strategy Ensemble accuracy is tested on the data point. Classifiers are updated using predicted labels from the *individual* classifiers.

Guided strategy Ensemble accuracy is tested on the data point. Classifiers are updated using the *ensemble* decision.

Error Driven Ensemble accuracy is tested on the data point. Classifiers whose individual predicted label does not agree with the ensemble decision are updated using the *ensemble* decision.

Confidence Driven Ensemble accuracy is tested on the data point. Confidence is calculated. Classifiers are updated when the confidence in the predicted label is above the threshold.

Error and Confidence Driven Ensemble accuracy is tested on the data point. Confidence is calculated. Classifiers whose individual predicted label does not agree with the ensemble decision are updated when the confidence in the predicted label is above the threshold.

7.3.2 Results

The cumulative error scores are calculated for each parameter, data set and method combination using Equation 3.8. For each data set the final error scores (taken at $t = 187$) are compared for each of the parameters and methods. The results are illustrated for the two data sets in Figure 7.5. Numerical values for the final error are available in Table A.5.

Table 7.2 summarises the number of wins per strategy. A winning ensemble is the ensemble with the lowest error at the final data point, $t = 187$. As there are twelve parameter combinations for each data set, there are 24 comparisons. Where there was a tie between methods for one parameter set, both methods were awarded a 'win' as either is acceptable when choosing an optimal strategy, hence the total sums to 25.

Table 7.3 directly compares each of the methods with the naive classifier. Methods which perform better than the naive classifier are indicated by a '+', methods which perform worse are indicated by a '−'. Significantly better/worse results (at $\alpha = 0.5$, tested using a paired t-test) are indicated by $\oplus$ and $\ominus$ respectively. The numbers of each 'result' are summarised in Table 7.4.

In order to see if the new strategies improve upon the performance of the guided ensemble, the results are directly compared. These are presented in Table 7.5, with

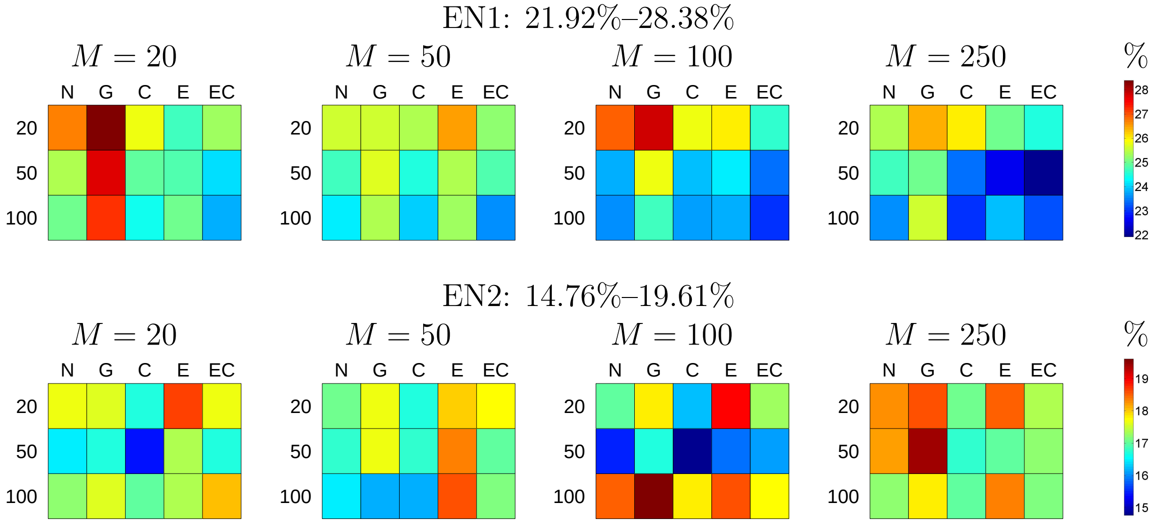


Figure 7.5: Final cumulative error scores. The range of final error scores for each data set is indicated in the title . Error scores are coloured from blue through to red representing low and high error respectively. L is the ensemble size, ($L = \{5, 9, 11\}$ represent the rows of each coloured grid). M is the cardinality of feature subsets. 'N', 'G', 'E', 'C' and 'EC' correspond to the naive, guided, error-driven, confidence-driven and error plus confidence-driven ensembles respectively.

Table 7.2: Number of wins per strategy, taken by comparing ensemble errors at the final data point. Total adds up to 25 due to a tie between methods for one parameter set.

Strategy	Number of wins
Naive	1
Guided	0
Error-driven	1
Confidence-driven	12
Error plus confidence-driven	11

Table 7.3: Comparison with naive classifier ensemble. 'G', 'E', 'C' and 'EC' correspond to the guided, error-driven, confidence-driven and error plus confidence-driven ensembles respectively. Methods which perform better than the naive ensemble are indicated by a '+', methods which perform worse are indicated by a '$-$'. Significantly better/worse results (at $\alpha = 0.5$, tested using a paired t-test) are indicated by $\oplus$ and $\ominus$.

		$M = 20$				$M = 50$				$M = 100$				$M = 250$			
		G	E	C	EC	G	E	C	EC	G	E	C	EC	G	E	C	EC
EN1	$L = 5$	$\ominus$	$\oplus$	+	$\oplus$	+	-	+	+	-	+	+	$\oplus$	-	+	-	+
	$L = 9$	$\ominus$	+	+	$\oplus$	-	-	+	-	$\ominus$	-	-	+	-	$\oplus$	+	$\oplus$
	$L = 11$	$\ominus$	+	+	$\oplus$	-	-	+	+	-	-	-	+	$\ominus$	-	+	+
EN2	$L = 5$	+	-	+	+	-	-	+	-	-	$\ominus$	+	-	-	-	+	+
	$L = 9$	-	-	$\oplus$	-	-	$\ominus$	+	-	$\ominus$	-	+	-	$\ominus$	+	$\oplus$	+
	$L = 11$	-	-	+	-	+	$\ominus$	+	-	-	-	+	+	-	-	+	+

Table 7.4: Summary of Table 7.3 showing number of each 'result' per strategy

Method	$\oplus$	$+$	$-$	$\ominus$
Guided	0	3	14	7
Error driven	2	5	14	3
Confidence driven	2	19	3	0
Confidence plus error	5	11	8	0

Table 7.5: Comparison with guided classifier ensemble. ‘E’, ‘C’ and ‘CE’ correspond to the error-driven, confidence-driven and error plus confidence-driven ensembles respectively. Methods which perform better than the guided ensemble are indicated by a ‘+’, methods which perform worse are indicated by a ‘$-$’. Significantly better/worse results (at $\alpha = 0.5$, tested using a paired t-test) are indicated by $\oplus$ and $\ominus$.

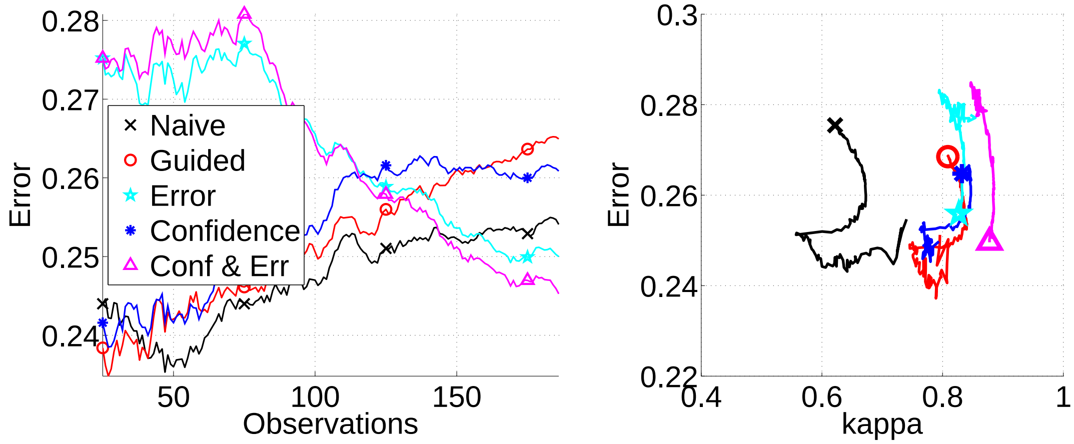
		$M = 20$			$M = 50$			$M = 100$			$M = 250$		
		E	C	EC	E	C	EC	E	C	EC	E	C	EC
EN1	$L = 5$	$\oplus$	$\oplus$	$\oplus$	-	+	+	+	$\oplus$	$\oplus$	+	+	$\oplus$
	$L = 9$	$\oplus$	$\oplus$	$\oplus$	-	+	+	+	$\oplus$	$\oplus$	+	+	$\oplus$
	$L = 11$	$\oplus$	$\oplus$	$\oplus$	+	+	+	+	$\oplus$	$\oplus$	$\oplus$	+	$\oplus$
EN2	$L = 5$	-	+	-	-	+	-	-	+	+	+	+	+
	$L = 9$	-	$\oplus$	+	-	+	+	+	$\oplus$	+	$\oplus$	$\oplus$	$\oplus$
	$L = 11$	+	+	-	$\ominus$	+	-	+	$\oplus$	+	-	+	+

Table 7.6: Summary of Table 7.5 showing number of each ‘result’ per strategy

Method	$\oplus$	$+$	$-$	$\ominus$
Error driven	5	10	8	1
Confidence driven	10	14	0	0
Confidence plus error	10	10	4	0

a summary in provided in Table 7.6.

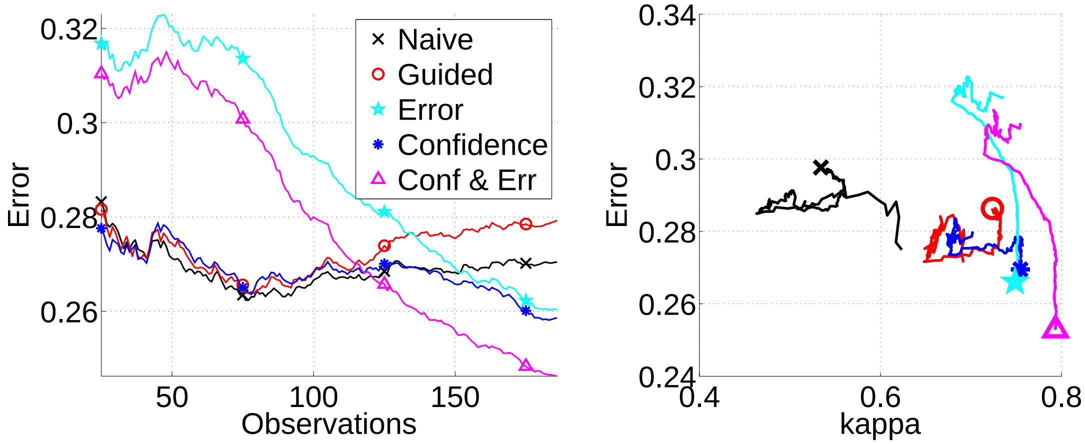
All three adaptations of the guided ensemble are shown to improve upon the guided ensemble. The methods are shown to perform well across a variety of parameters. The results from Tables 7.2, 7.3, 7.4, 7.5 and 7.6 suggest that of those tested, the best method for streaming fMRI data is the confidence driven method. The method compares well against both the naive and guided ensembles. In direct comparison with the guided ensemble, the confidence-driven ensemble is shown to perform better for all parameters. The methods appear to perform better for EN1 than EN2. To better see what is happening, the cumulative error plots and kappa error trajectories are considered. Figures 7.6, 7.7, 7.8 and 7.9 show the pairs of cumulative error diagrams and kappa error trajectories for a selection of parameter combinations and data sets.



(a) Cumulative error plot

(b) Kappa error trajectory

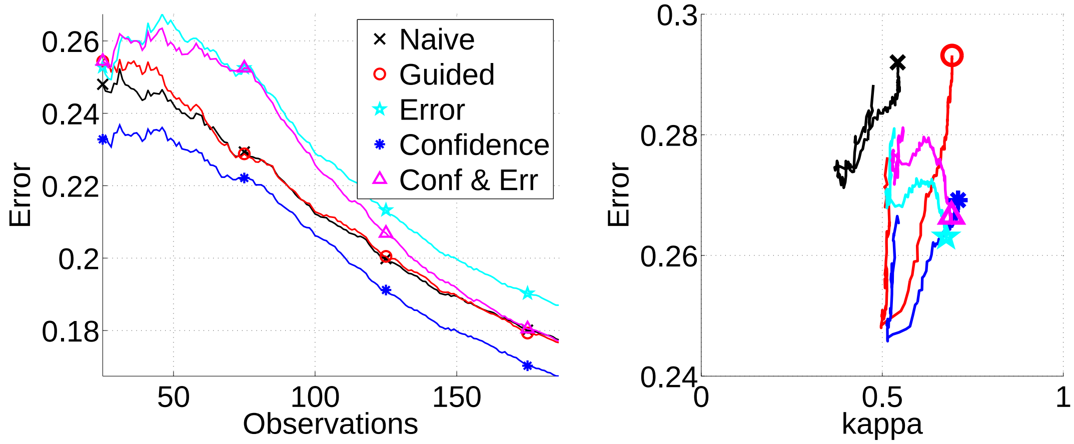
Figure 7.6: Data set EN1, $L = 5$, $M = 250$



(a) Cumulative error plot

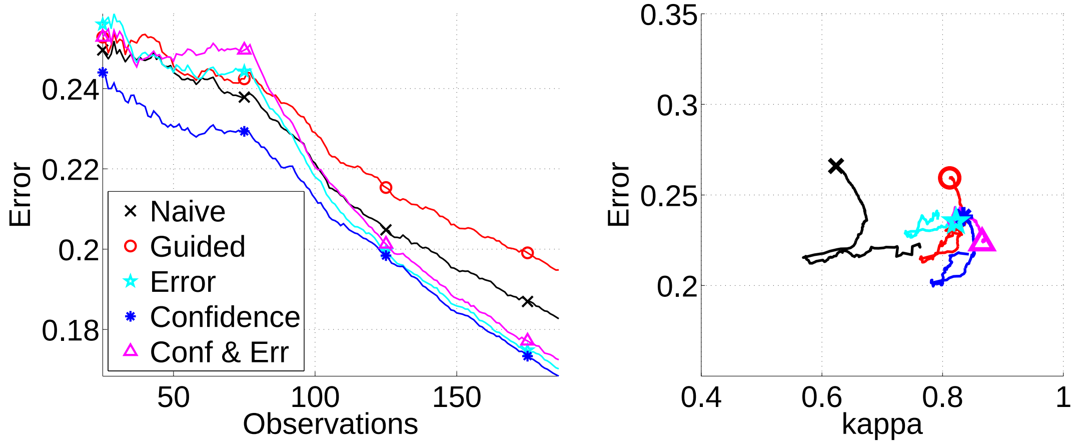
(b) Kappa error trajectory

Figure 7.7: Data set EN1, $L = 5$, $M = 100$



(a) Cumulative error plot (b) Kappa error trajectory

Figure 7.8: Data set EN2, $L = 5$, $M = 20$



(a) Cumulative error plot (b) Kappa error trajectory

Figure 7.9: Data set EN2, $L = 9$, $M = 250$

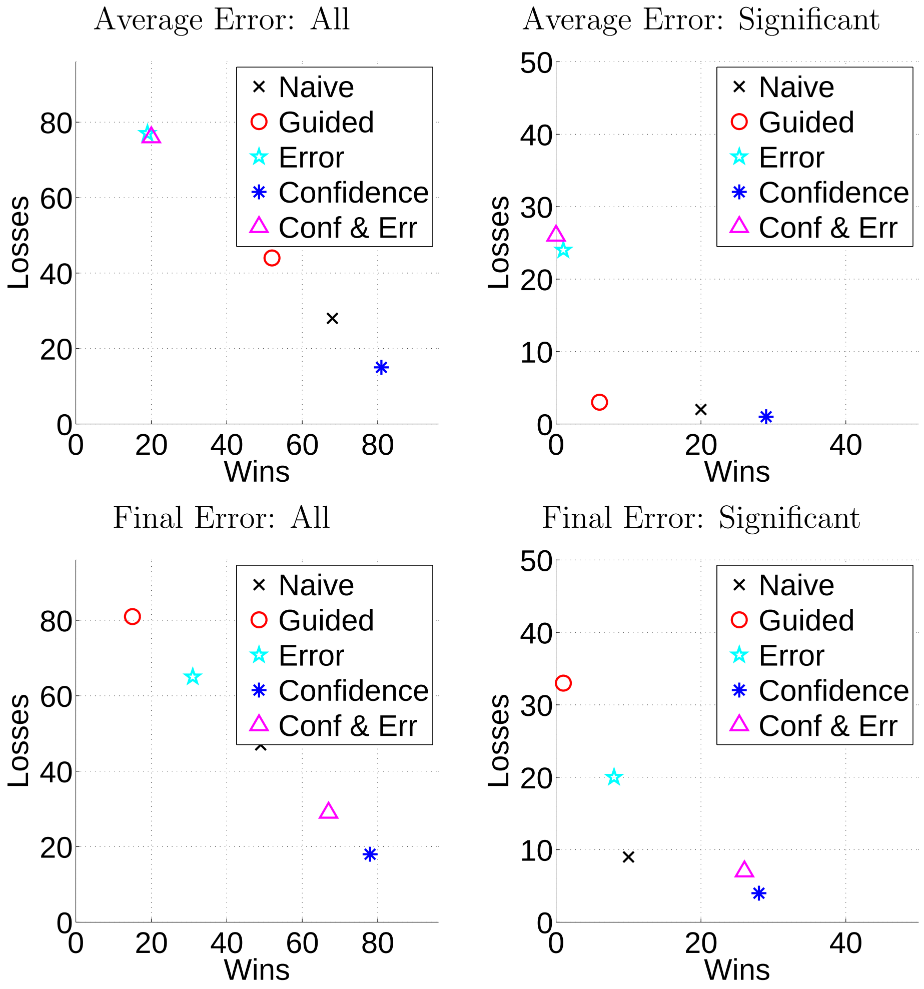


Figure 7.10: Pairwise wins vs losses. Significance calculated at $\alpha = 0.05$.

Again, pairwise comparisons are performed using a t-test. With 5 methods tested across 12 parameter sets and 2 data sets, a total of 240 pairwise comparisons are made. The numbers of wins vs losses are plotted in Figure 7.10. The best point is at 96 wins and no losses, the worst point at 0 wins and 96 losses.

For EN1, in both cases, initially the error rates of the error-driven ensemble and confidence plus error-driven ensemble are higher than these for the other three ensembles. As more streaming data points are processed the error rate for these two ensembles continues to drop, whilst the error rate of the naive ensemble and guided ensemble rises. These error changes are reflected in the kappa error trajectory diagrams. The interesting point to note from these, is that the higher kappa value (and thus *lower* diversity) is associated with the most accurate ensembles. It may be that as the individual classifiers within the ensemble become more accurate they are driven towards the optimal classifier, and thus as they become more accurate, they become more similar and the diversity decreases.

For EN2, the cumulative error rate of all ensembles follows a similar pattern. From the kappa-error trajectory diagrams the pairwise accuracy of the naive, guided and confidence-driven ensembles can be seen to initially decrease, and then increase. This is matched by an initial increase then decrease in diversity. The 'error-driven' and 'confidence plus error-driven' ensembles are seen to follow the opposite pattern, resulting in a higher pairwise accuracy and lower diversity than the other three ensembles.

Whilst the ensembles follow different error patterns for the two data sets, the results for the confidence-driven and confidence plus error-driven ensembles are consistently better than, or on a par with the guided ensemble. Both strategies also show good results compared with the naive ensemble.

7.3.3 Discussion

Real-time fMRI classification faces the challenge of unlabelled data and concept drift. The naive and guided ensembles have previously been shown to handle streaming data better than a fixed classifier. This study proposes several extensions to the guided update strategy, in an attempt to maintain ensemble diversity and accuracy, and constrain the possibility of runaway classifiers given the potential changing environment. The solutions have been tested on streaming fMRI data.

When compared with the naive and guided ensembles, for two two-class data sets, ensembles updating using the confidence criterion, or confidence combined with error, were shown to perform best. These methods performed well consistently, compared with both the naive and guided ensembles. Experiments were carried out with a simple O-LDC classifier for its simplicity and speed. Whilst the experiments are not carried out in real time, there is no reason that the online classification algorithms would not be capable of working in real time. Further work testing the methods with a variety of base classifiers and across a wider variety of data sets may give a deeper understanding of the mechanism of improvement offered by the guided update.

Chapter 8

Conclusion

8.1 Summary of Work

Classification of fMRI data comes with many challenges. Performing the classification of fMRI data in real time only adds to these. By applying online classification to fMRI data, it has been shown that classifier performance improves as more instances are seen. Linear models have been popular for classification of fMRI data as they are sufficiently accurate and are fast enough to work in real time. This work has continued in the same vein, using an online variant of the linear discriminant classifier (O-LDC). Results have been shown to be fast, with updates occuring well within the acceptable time (2s) for 'immediate' feedback in real-time fMRI trials.

The O-LDC has been used in conjunction with a random subspace (RS) ensemble. The ensemble framework allows for fewer features to be used per classifier, and due to the excessive amount of features in fMRI data, is less computationally expensive than a single classifier trained upon the entire feature set. By using an RS ensemble, there is also a lesser risk of over-fitting the classification model.

It was noted that there are cases in fMRI experiments when the true brain state of a participant is unknown. In such cases, it is impossible to update the classifier using a known label. It was hypothesised that using the label predicted by the classifier as the true label, online classification could still be applied. Furthermore, it was hypothesised that by using the ensemble label rather than the individual classifier labels to update the ensemble members, a more accurate result would be achieved.

Chapter 5 introduces streaming fMRI data, simulating a real time scenario. Three base classifiers are tested in an ensemble, the perceptron, winnow and O-LDC. The O-LDC was shown to have a better learning pattern than the other two classifiers, and produced more accurate results.

Chapter 6 presents a study of the naive labelling scenario, using RS ensembles of O-LDC classifiers for i.i.d. (shuffled) fMRI data. The ensemble update strategy was shown to perform better than the individual update strategy, with both update strategies performing better than a fixed, pre-trained classifier.

Combining the results of both studies, a further study, presented in Chapter 7 considers naive labelling for streaming fMRI data. Once again, the ensemble update strategy was shown to prevail. Acknowledging that there is a potential for concept drift in fMRI data, and that there is uncertainty around the transition period between brain states, two update criteria were introduced. These criteria correspond to the certainty of the ensemble decision, and the decision of the member classifiers within the ensemble. Updating the member classifiers when the ensemble was confident in its decision proved to be the best strategy, and meant that classifiers were not updated during the uncertain, transition phases.

8.2 Future Work

The next step would be to apply the methods discussed here in real time. Unfortunately this was not possible for this thesis as there was no direct access to the fMRI scanner. The updates for the O-LDC are fast. Both classifying the incoming data point and updating the classifier ensemble happen within a TR, allowing for immediate feedback.

Perhaps one of the greater challenges with real time fMRI is that of preprocessing the raw data and preparing each data volume for classification. Working with raw data direct from the scanner, whilst the fastest approach, introduces excess noise into the data. Steps such as filtering and motion correction can be computationally expensive, and large head motions may be over compensated, introducing error. Weiskopf et al [129] detail some of the challenges and proposed solutions for preprocessing fMRI data in real time. With computational power ever on the increase, it is not expected that these issues would be of great consequence.

Another interesting line of future work would be to further examine the mechanisms of using the ensemble decision for classifier updates, in particular, with the update criteria in place. Greater understanding of this process could lead to the development of further techniques, not just for the RS ensemble with O-LDC classifiers, but for a variety of ensemble frameworks and base classifiers, applicable to many types of very high-dimensional streaming data.

8.3 Publications Relating to the Thesis

1. L. I. Kuncheva and C. O. Plumpton. Adaptive learning rate for online linear discriminant classifiers. In *Proc. Joint IAPR International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition S+SSPR*, pages 510-519, Orlando, Florida, USA, 2008.

2. L. I. Kuncheva, J. J. Rodrguez, C. O. Plumpton, D. E. J. Linden and S. J. Johnston. Random subspace ensembles for fMRI classification. *IEEE Transaction on Medical Imaging*, 29:531–42, 2010.

3. L. I. Kuncheva and C. O. Plumpton. Choosing parameters for Random Subspace ensembles for fMRI classification. In *Proc. Multiple Classifier Systems*, 2010

4. C. O. Plumpton, L. I. Kuncheva, D. E. J. Linden, and S. J. Johnston. Online fMRI classification using linear and ensemble classifiers, In *International Conference on Pattern Recognition*, 2010.

5. C. O. Plumpton, L. I. Kuncheva, N. N. Oosterhof and S. J. Johnston. Naive Random Subspace Ensemble with Linear Classifiers for Real-time Classification of fMRI Data, *Pattern Recognition*, In Press, Corrected Proof, 2011.

6. C. O. Plumpton. Online Semi-Supervised Ensemble Updates for fMRI Data, *International Journal of Applied Mathematics and Statistics: Advances in Ensemble Learning and Its Applications* Accepted, 2011.

Appendix A

Results Tables

Table A.1: EN1 Error Table (%)

		$L = 5$			$L = 9$			$L = 11$			Individual		
		F	N	S	F	N	S	F	N	S	F	N	S
$\|T\| = 20$	$M = 20$	30.68	31.12	14.26	29.42	29.94	13.41	30.00	29.97	12.83	33.28	34.61	19.76
	$M = 50$	29.52	28.75	7.84	27.86	28.48	7.30	27.33	26.66	7.28	29.93	31.46	11.45
	$M = 100$	27.45	27.52	5.18	28.99	27.99	4.88	26.56	25.08	4.71	28.63	30.18	7.06
$N_T = 40$	$M = 20$	24.88	25.66	13.68	21.75	22.33	11.23	20.62	21.03	11.69	32.43	32.97	19.49
	$M = 50$	25.43	24.09	8.08	25.45	24.39	7.54	24.21	23.19	7.19	27.04	27.00	11.14
	$M = 100$	24.74	24.16	5.36	26.59	24.77	5.18	24.86	23.44	4.76	26.33	26.14	6.86
$N_T = 100$	$M = 20$	18.80	19.82	12.24	17.37	18.06	11.14	16.40	17.39	11.02	25.21	25.96	17.83
	$M = 50$	17.43	16.85	5.24	13.42	14.57	4.34	15.04	15.27	4.25	24.71	25.06	9.16
	$M = 100$	23.64	22.24	5.07	23.44	21.74	4.86	24.47	23.17	4.95	24.75	23.44	6.18

Table A.2: EN2 Error Table (%)

		$L = 5$			$L = 9$			$L = 11$			Individual		
		F	N	S	F	N	S	F	N	S	F	N	S
$N_T = 20$	$M = 20$	32.30	32.39	17.78	31.05	31.45	16.55	30.69	30.32	16.23	33.25	34.79	23.29
	$M = 50$	29.80	29.15	10.23	30.11	29.63	9.86	30.54	29.43	9.18	31.09	32.79	15.15
	$M = 100$	30.14	30.43	7.08	30.11	28.94	6.76	30.10	30.04	6.56	30.66	32.62	9.64
$N_T = 40$	$M = 20$	29.16	29.49	16.40	27.05	27.73	14.80	25.36	25.92	14.70	35.69	36.19	22.80
	$M = 50$	30.45	28.49	10.12	30.91	29.30	9.75	29.43	27.73	9.12	30.93	30.83	14.59
	$M = 100$	29.87	28.82	6.86	28.96	26.83	6.16	28.64	26.87	6.17	29.57	29.50	8.92
$N_T = 100$	$M = 20$	23.96	24.16	15.36	22.00	22.80	13.12	21.18	21.46	12.50	29.52	30.36	20.90
	$M = 50$	21.94	23.32	6.82	20.56	20.62	5.50	18.38	18.14	5.25	29.43	30.10	11.88
	$M = 100$	28.90	26.90	5.90	29.42	26.01	5.84	28.80	25.73	5.54	29.44	27.30	7.41

Table A.3: EB Error Table (%)

		$L = 5$			$L = 9$			$L = 11$			Individual		
		F	N	S	F	N	S	F	N	S	F	N	S
$N_T = 20$	$M = 20$	63.00	62.93	30.73	62.52	63.54	27.10	64.71	63.44	26.04	64.12	64.68	40.33
	$M = 50$	62.03	62.73	15.72	63.09	62.25	14.20	62.22	63.11	13.56	62.90	63.84	22.24
	$M = 100$	62.49	61.59	10.42	61.78	62.80	10.31	63.45	63.80	10.16	62.98	63.60	11.81
$N_T = 40$	$M = 20$	62.20	62.46	26.08	61.51	62.79	22.11	60.21	60.83	21.87	64.03	64.09	36.98
	$M = 50$	61.74	61.36	13.00	61.65	61.44	11.67	60.99	61.87	11.14	62.46	62.64	18.30
	$M = 100$	62.23	62.17	8.71	60.90	60.16	8.33	61.27	61.02	8.17	62.04	62.10	9.67
$N_T = 100$	$M = 20$	56.42	58.01	1.82	58.76	59.25	1.60	60.91	60.26	1.56	62.00	62.14	2.81
	$M = 50$	57.19	56.85	1.11	54.21	55.23	1.08	58.97	57.69	1.15	60.64	60.72	1.48
	$M = 100$	59.12	58.76	1.25	59.10	60.31	1.31	58.11	59.54	1.33	60.52	60.64	1.38

Table A.4: EN1 Error (%)

	$M = 20$			$M = 50$			$M = 100$			$M = 250$		
	F	N	S	F	N	S	F	N	S	F	N	S
$L = 5$	22.18	19.61	18.35	21.71	18.78	18.24	21.96	19.56	19.47	22.03	20.24	21.22
$L = 9$	22.02	18.92	17.82	21.91	18.61	18.09	22.00	18.92	19.39	22.03	20.92	22.09
$L = 13$	21.76	18.90	17.95	21.90	18.36	17.83	21.99	19.14	19.38	22.00	20.59	21.87

Table A.5: Error Table (%)

		EN1			EN2		
		$L = 5$	$L = 9$	$L = 11$	$L = 5$	$L = 9$	$L = 11$
$M = 20$	N	26.82	25.45	25.05	17.74	16.46	17.25
	G	28.38	27.80	27.27	17.67	16.72	17.61
	C	25.89	24.98	24.45	16.74	15.40	17.04
	E	24.75	24.86	25.02	18.70	17.43	17.43
	EC	25.37	24.17	23.82	17.71	16.70	18.14
$M = 50$	N	25.67	24.76	24.19	17.14	16.82	16.46
	G	25.65	25.74	25.49	17.73	17.71	16.22
	C	25.44	24.56	24.02	16.74	16.82	16.17
	E	26.54	25.41	25.34	18.01	18.41	18.66
	EC	25.25	24.86	23.60	17.78	17.06	17.20
$M = 100$	N	27.04	23.86	23.66	17.01	15.47	18.54
	G	27.92	25.87	24.74	17.87	16.75	19.61
	C	25.86	23.89	23.67	16.26	14.76	17.87
	E	26.04	24.24	23.81	19.06	15.90	18.67
	EC	24.62	23.40	23.01	17.38	16.13	17.80
$M = 250$	N	25.41	24.77	23.66	18.37	18.27	17.24
	G	26.49	25.10	25.66	18.61	19.48	17.90
	C	26.09	23.37	23.02	17.09	16.84	17.01
	E	25.00	22.59	23.87	18.57	17.02	18.44
	EC	24.53	21.92	23.25	17.40	17.25	17.17

Bibliography

[1] M. B. Aberg, K. Rylander, and J. Wessberg. An evolutionary approach to the identification of informative voxel clusters for brain state discrimination. *IEEE Journal of Selected Topics in Signal Processing*, 2(6):919 –928, December 2008.

[2] J. R. Anderson, S. Betts, J. L. Ferris, and J. M. Fincham. Neural imaging to track mental states while using an intelligent tutoring system. *Proc Natl Acad Sci U S A. Neuroscience, Psychological and Cognitive Sciences*, 107(15):7018–7023, 2010.

[3] R. Avnimelech and N. Intrator. Boosted mixture of experts: An ensemble learning scheme. *Neural Computation*, 11(2):483–497, 1999.

[4] S. Baron-Cohen, H. A. Ring, E. T. Bullmore, C. Ashwin S. Wheelwright, and S. C. R. Williams. The amygdala theory of autism. *Neuroscience and Biobehavioral Reviews*, 24:355–364, 2000.

[5] C. M. Bennett, A. A. Baird, M. B. Miller, and G. L. Wolford. Neural correlates of interspecies perspective taking in the post-mortem atlantic salmon: An argument for multiple comparisons correction. In *Organization for Human Brain Mapping Abstracts*, 2009.

[6] C. M. Bennett and M. B. Miller. How reliable are the results from functional magnetic resonance imaging? *Annals of the New York Academy of Sciences*, 1191:133–155, 2010.

[7] A. Bertoni, R. Folgieri, and G. Valentini. Feature selection combined with random subspace ensemble for gene expression based diagnosis of malignancies. In B.Apolloni, M.Marinaro, and R. Tagliaferri, editors, *Biological and Artificial Intelligence Environments*, pages 29–36. Springer, 2005.

[8] A. Bertoni, R. Folgieri, and G. Valentini. Random subspace ensembles for the biomolecular diagnosis of tumors. *Neurocomputing*, 63C:535–539, 2005.

[9] M. Bjornsdotter, K. Rylander, and J. Wessberg. A monte carlo method for locally multivariate brain mapping. *NeuroImage*, 56(2):508–516, May 2011.

[10] M. Bjornsdotter and J. Wessberg. A memetic algorithm for selection of 3d clustered features with applications in neuroscience. In *International Conference on Pattern Recognition*, 2010.

[11] B. Blankertz, G. Curio, and K. Muller. Classifying single trial EEG: Towards brain computer interfacing. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pages 157–164. MIT Press, 2002.

[12] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *in COLT 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[13] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, pages 144–152, 1992.

[14] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[15] G. Brown. Ensemble learning. In Claude Sammut and Geoffrey Webb, editors, *In Encyclopedia of Machine Learning*. Springer Verlag, 2009.

[16] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005.

[17] A. Buttfield and J. del R. Millán. Online classifier adaptation in brain-computer interfaces, 0 2006.

[18] N. R. Carlson. *Physiology of Behaviour, 4th Edition*. Allyn & Bacon, 1991.

[19] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, 14:67–74, November 1999.

[20] N. V. Chawla and K. W. Bowyer. Random subspaces and subsampling for 2-d face recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:582–589, 2005.

[21] M. Ciaramita, V. Murdock, and V. Plachouras. Online learning from click data for sponsored search. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 227–236, 2008.

[22] M. S. Cohen. Real-time functional magnetic resonance imaging. *Methods*, 25:201–220, 2001.

[23] D. D. Cox and R. L. Savoy. Functional magnetic resonance imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, 19(2 Pt 1):261–270, June 2003.

[24] R. W. Cox. AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research*, 29:162–173, 1996.

[25] R. W. Cox, A. Jesmanowicz, and J. S. Hyde. Real-time functional magnetic resonance imaging. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 33(2):230–6, February 1995.

[26] F. Cozman, I. Cohen, and M. Cirelo. Semi-supervised learning of mixture models and bayesian networks. In *Proceedings of the 20th International Conference on Machine Learning*, pages 99–106, 2003.

[27] F. G. Cozman and I. Cohen. Unlabeled data can degrade classification performance of generative classifiers. In *Proceedings of the 15th International FLAIR Conference*, pages 327–331, 2002.

[28] R. J. Davidson, C. E. Schaffer, and C. Saron. Effects of lateralized presentations of faces on self-reports of emotion and EEG asymmetry in depressed and non-depressed subjects. *Psychophysiology*, 22(3):353–364, 1985.

[29] F. De Martino, F. Gentile, F. Esposito, M. Balsi, F. Di Salle, R. Goebel, and E. Formisano. Classification of fMRI independent components using ic-fingerprints and support vector machine classifiers. *NeuroImage*, 34:177–194, 2007.

[30] F. De Martino, G. Valente, N. Staeren, J. Ashburner, R. Goebel, and E. Formisano. Combining multivariate voxel selection and support vector machines for mapping and classification of fMRI spatial patterns. *NeuroImage*, 43(1):44–58, 2008.

[31] R. C. DeCharms. Applications of real-time fMRI. *Nature Reviews Neuroscience*, 9(9):720–9, 2008.

[32] R. C. DeCharms, F. Maeda, G. H. Glover, D. Ludlow, J. M. Pauly, D. Soneji, J. D. E. Gabrieli, and S. C. Mackey. Control over brain activation and pain learned by using real-time functional MRI. In *Proc Natl Acad Sci USA*, volume 102(51), pages 18626–31, 2005.

[33] P. Domingos and G. Hulten. Mining high-speed data streams. In *in Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery anddata mining*, pages 71–80, 2000.

[34] P. Domingos and G. Hulten. A general framework for mining massive data stream. *Journal of Computational and Graphical Statistics*, 12(4):945–949, 2003.

[35] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley & Sons, NY, second edition, 2001.

[36] A. Eklund, M. Andersson, H. Ohlsson, A. Ynnerman, and H. Knutsson. A brain computer interface for communication using real-time fmri. In *International Conference on Pattern Recognition*, 2010.

[37] A. Eklund, H. Ohlsson, M. Andersson, J. Rydell, A. Ynnerman, and H. Knutsson. Using real-time fMRI to control a dynamical system by brain activity classification. In *Proc. MICCAI'09*, London, UK, 2009. Springer.

[38] G. Esposito, B. S. Kirkby, J. D. Van. Horn, T. M. Ellmore, and K. F. Berman. Context-dependent, neural system-specific neurophysiological concomitants of ageing: mapping PET correlates during cognitive activation. *Brain*, 122:963 – 979, 1999.

[39] M. W. Eysenck and C. Flanagan. *Psychology for A2 level.* Psychology Press Ltd, 2001.

[40] J. L. Fleiss. *Statistical Methods for Rates and Proportions.* John Wiley & Sons, 1981.

[41] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55:119–139, 1997.

[42] K. Friston, J. Ashburner, S. Kiebel, T. Nichols, and W. Penny (Eds.). *Statistical Parametric Mapping: The Analysis of Functional Brain Images,*. Academic Press, 2007.

[43] A. Fuchs, V. K. Jirsa, and J. A. S. Kelso. Theory of the relation between human brain activity (MEG) and hand movements. *NeuroImage*, 11(5):359 – 369, 2000.

[44] B. Gabrys and L. Petrakieva. Combining labelled and unlabelled data in the design of pattern classification systems. *International Journal of Approximate Reasoning*, 35:251–273, 2004.

[45] H. Garavan. Insula and drug cravings. *Brain Structure and Function*, 214:593–601, 2010.

[46] A. Geissler, A. Gartus, T. Foki, A. R. Tahamtan, R. Beisteiner, and M. Barth. Contrast-to-Noise Ratio (CNR) as a Quality Parameter in fMRI. *Journal of Magnetic Resonance Imaging*, 25:1263–1270, 2007.

[47] R. Goebel, F. Espositio, and E. Formisano. Analysis of functional image analysis contest (FIAC) data with Brainvoyager QX: From single-subject to cortically aligned group general linear model analysis and self-organizing group independent component analysis. *Human Brain Mapping*, 27:392–401, 2006.

[48] L. Grosenick, S. Greer, and B. Knutson. Interpretable classifiers for fMRI improve prediction of purchases. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(6):539–548, 2008.

[49] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[50] D. A. Handwerker, J. M. Ollinger, and M. D'Esposito. Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. *Neuroimage*, 21(4):1639–1651, 2004.

[51] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

[52] T. K. Ho. The random space method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (8):832 – 844, 1998.

[53] M. Hollmann, T. Mönch, S. Mulla-Osman, C. Tempelmann, J. Stadler, and J. Bernarding. A new concept of a unified parameter management, experiment control, and data analysis in fMRI: Application to real-time fMRI at 3T and 7T. *Journal of Neuroscience Methods*, 175:154–162, 2008.

[54] M. Hollmann, T. Mönch, C. Muller, and J. Bernarding. Predicting human decisions in socioeconomic interaction using real-time functional magnetic resonance (rtfMRI). In *SPIE-Medical Imaging*, 2009.

[55] M. Hollmann, T. Mönch, C. Tempelmann, and J. Bernarding. An unified approach for fMRI-measurements used by a new real-time fMRI analysis system. In *Bildverarbeitung für die Medizin*, 2007.

[56] A. J. Holmes, A. MacDonald, C. S. Carter, D. M. Barch, V. A. Stenger, and J. D. Cohen. Prefrontal functioning during context processing in schizophrenia and major depression: An event-related fMRI study. *Schizophrenia Research*, 76:199–206, 2005.

[57] K. Hugdah, B. R. Rund, A. Lund, A. Asbjornsen, J. Egeland, L. Ersland, N. I. Landro, A. Roness, K. I. Stordal, K. Sundet, and T. Thomsen. Brain activation measured with fMRI during a mental arithmetic task in schizophrenia and major depression. *American Journal of Psychiatry*, 161:286–293, 2004.

[58] S. J. Johnston, S. G. Boehm, D. Healy, R. Goebel, and D. E. J. Linden. Neurofeedback: A promising tool for the self-regulation of emotion networks. *Neuroimage*, 49(1):1066–72, 2010.

[59] T. M. Mitchell K. Nigam, A. McCallum. *Semi-Supervised Text Classification Using EM.* MIT Press, 2006.

[60] L. Kauhanen, T. Nykopp, J. Lehtonen, P. Jylanki, J. Heikkonen, P. Rantanen, H. Alaranta, and M. Sams. EEG and MEG brain computer interface for tetraplegic patients. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):190 –193, 2006.

[61] K. M. Kelly, D. S. Shiau, R. T. Kern, J. H. Chien, M. C. K. Yang, K. A. Yandora, J. P. Valeriano, J. J. Halford, and J. C. Sackellares. Assessment of a scalp EEG-based automated seizure detection system. *Clinical Neurophysiology*, 121(11):1832 – 1843, 2010.

[62] J. Kong, N. S. White, K. K. Kwong, M. G. Vangel, I. S. Rosman, R. H. Gracely, and R. L. Gollub. Using fMRI to dissociate sensory encoding from cognitive evaluation of heat pain intensity. *Human Brain Mapping*, 27:715–721, 2006.

[63] I. Koprinska, D. Deng, and F. Feger. Image classification using labelled and unlabelled data. In *in Proceedings of the 14th European Signal and Image Processing Conference (EUSIPCO)*, 2006.

[64] N. Kriegeskorte, R. Goebel, and P. Bandettini. Information-based functional brain mapping. In *Proceedings of the National Academy of Sciences of the United States of America*, 2006.

[65] N. Kriegeskorte, W. K. Simmons, P. SF. Bellgowan, and C. I. Baker. Circular analysis in systems neuroscience: the dangers of double dipping. *Nature Neuroscience*, 12:535–540, 2009.

[66] S. P. Ku, A. Gretton, J. Macke, and N. K. Logothetis. Comparison of pattern recognition methods in classifying high-resolution BOLD signals obtained at high magnetic field in monkeys. *Magnetic Resonance Imaging*, 26(7):1007 – 1014, 2008.

[67] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms.* Wiley-Interscience, 2004.

[68] L. I. Kuncheva and C. O. Plumpton. Adaptive learning rate for online linear discriminant classifiers. In *Proc. Joint IAPR International Workshops on Structural and Syntactic Pattern Recognition and Statistical Techniques in Pattern Recognition S+SSPR*, pages 510–519, Orlando, Florida, USA, 2008.

[69] L. I. Kuncheva and C. O. Plumpton. Choosing parameters for random subspace ensembles for fMRI classification. In *Proc. Multiple Classifier Systems*, 2010.

[70] L. I. Kuncheva and J. J. Rodriguez. Classifier ensembles with a random linear oracle. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):500–508, 2007.

[71] L. I. Kuncheva and J. J. Rodríguez. Classifier ensembles for fMRI data analysis: An experiment. *Magnetic Resonance Imaging*, 28(4):583–593, 2010.

[72] L. I. Kuncheva, J. J. Rodríguez, C. O. Plumpton, D. E. J. Linden, and S. J. Johnston. Random subspace ensembles for fMRI classification. *IEEE Transaction on Medical Imaging*, 29(2):531–42, 2010.

[73] L. I. Kuncheva, C. J. Whitaker, and A. Narasimhamurthy. A case study on naïve labelling for the nearest mean and the linear discriminant classifiers. *Pattern Recognition*, 41:3010–3020, 2008.

[74] L. I. Kuncheva, C. J. Whitaker, C. A. Shipp, and R. P. W. Duin. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications*, 6:22–31, 2003.

[75] S. LaConte, S. Strother, V. Cherkassky, J. Anderson, and X. Hu. Support vector machines for temporal classification of block design fMRI data. *NeuroImage*, 26(2):317 – 329, 2005.

[76] S. M. LaConte. Decoding fmri brain states in real-time. *NeuroImage*, 56(2):440–454, May 2011.

[77] S. M. LaConte, S. J. Peltier, and X. P. Hu. Real-time fMRI using brain-state classification. *Human Brain Mapping*, 28:1033–1044, 2007.

[78] L. Lam and C. Y. Suen. Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Systems, Man and Cybernetics. Part A: Systems and Humans*, 27(5):553 – 568, 1997.

[79] P. J. Lang, M. M. Bradley, and B. N. Cuthbert. *International Affective Picture System (IAPS): Technical Manual and Affective Ratings.* NIMH Center for the Study of Emotion and Attention, University of Florida, 1997.

[80] N. D. Lawrence and M. I. Jordan. Semi-supervised learning via gaussian processes. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 753–760, 2005.

[81] J. Lehtonen, P. Jylanki, L. Kauhanen, and M. Sams. Online classification of single EEG trials during finger movements. *Biomedical Engineering, IEEE Transactions on*, 55(2):713 –720, feb 2008.

[82] M. Li and Z.-H. Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man and Cybernetics*, A 37(6):1088–1098, 2007.

[83] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, April 1988.

[84] Q. Lu and L. Getoor. Link-based classification using labeled and unlabeled data. In *n ICML Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.

[85] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying suspicious urls: an application of large-scale online learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML 09, pages 681–688, 2009.

[86] D. D. Margineantu and T. G. Dietterich. Pruning adaptive boosting. In *Proc. 14th International Conference on Machine Learning*, pages 378–387, San Francisco, 1997. Morgan Kaufmann.

[87] Matlab. *http://www.mathworks.com/products/matlab/*.

[88] M. J. Minzenberg, J. Fan, A. S. New, C. Y. Tang, and L. J. Siever. Fronto-limbic dysfunction in response to facial emotion in borderline personality disorder: An

event-related fMRI study. *Psychiatry Research: Neuroimaging*, 155(3):231–243, August 2007.

[89] M. Misaki, Y. Kim, P. A. Bandettini, and N. Kriegeskorte. Comparison of multivariate classifiers and response normalizations for pattern-information fMRI. *NeuroImage*, 53(1):103–118, 2010.

[90] T. Mitchell, R. Hutchinson, R. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57((1-2)):145–175, 2004.

[91] T. Moench, M. Hollmann, R. Grzeschik, C. Muller, R. Luetzkendorf, S. Baecke, M. Luchtmann, D. Wagegg, and J. Bernarding. Real-time classification of activated brain areas for fMRI-based human-brain-interfaces. In Anne V. Hu, Xiaoping P.; Clough, editor, *Medical Imaging 2008: Physiology, Function, and Structure from Medical Images*, volume 6916, pages 69161R – 69161R–10, 2008.

[92] M. M. Monti, A. Vanhaudenhuyse, M. R. Coleman, M. Boly, J. D. Pickard, L. Tshibanda, A. M. Owen, and S. Laureys. Willful modulation of brain activity in disorders of consciousness. *New England Journal of Medicine*, 362(7):579–589, 2010.

[93] J. Mourao-Miranda, A. L. Bokde, C. Born, H. Hampel, and M. Stetter. Classifying brain states and determining the discriminating activation patterns: Support Vector Machine on functional MRI data,. *NeuroImage*, 28(4):980 – 995, 2005.

[94] J. Mourao-Miranda, K. J. Friston, and M. Brammer. Dynamic discrimination analysis: A spatial-temporal SVM. *NeuroImage*, 36(1):88 – 99, 2007.

[95] J. Mourao-Miranda, E. Reynaud, F. McGlone, G. Calvert, and M. Brammer. The impact of temporal compression and space selection on SVM analysis of

single-subject and multi- subject fMRI data. *NeuroImage*, 33(4):1055 – 1065, 2006.

[96] G. Nagy. Classifiers that improve with use. Information and Communication Engineers Technical Report 103, Institution of Electronics, 2004.

[97] K. P. Nigam. *Using Unlabeled Data to Improve Text Classification*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, US, 2001.

[98] N. C. Oza and K. Tumer. Classifier ensembles: Select real-world applications. *Inf. Fusion*, 9:4–20, January 2008.

[99] M. Palatucci and A. Carlson. On the chance accuracies of large collections of classifiers. In *Proceedings of the 25th International Conference on Machine Learning*, July 2008.

[100] T. D. Papageorgiou, W. A. Curtis, M. McHenry, and S. M. LaConte. Neurofeedback of two motor functions using supervised learning-based real-time functional magnetic resonance imaging. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 5377–5380, Sept 2009.

[101] F. Pereira, T. Mitchell, and M. Botvinick. Machine learning classifiers and fMRI: a tutorial overview. *NeuroImage*, 45 (1, Supplement 1):S199 – S209, 2009.

[102] A. Pissiota, I. Frans, M. Fernandez, L. von. Knorring, H. Fischer, and M. Fredrikson. Neurofunctional correlates of posttraumatic stress disorder: a PET symptom provocation study. *European Archives of Psychiatry and Clinical Neuroscience*, 252:68–75, 2002.

[103] A. Ploghaus, I. Tracey, J. S.. Gati, S. Clare, R. S. Menon, P. M. Matthews, and J. N. P. Rawlins. Dissociating pain from its anticipation in the human brain. *Science*, 284(5422):1979–1981, June 1999.

[104] C. O. Plumpton. *On-line Linear Discriminant Classifier and its Application to Delayed Labelling*. VDM Verlag, 2009.

[105] R. Polikar, J. DePasquale, H. Syed Mohammed, G. Brown, and L. I. Kuncheva. Learn++.MF: A random subspace approach for the missing feature problem. *Pattern Recogn.*, 43:3817–3832, November 2010.

[106] S. Posse, D. Fitzgerald, K. Gao, U. Habel, D. Rosenberg, G. J. Moore, and F. Schneider. Real-time fMRI of temporolimbic regions detects amygdala activation during single-trial self-induced sadness. *NeuroImage*, 18:760–768, 2003.

[107] H. Qi, H. Xiaoning, Y. Muyun, L. Jun, L. Guohua, H. Zhongyuan, and L. Sheng. Online linear discriminative learning for spam filter. In *Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 2, pages 306–309, oct 2008.

[108] J. Richiardi, H. Eryilmaz, S. Schwartz, P. Vuilleumier, and D. Van De Ville. Brain decoding of fMRI connectivity graphs using decision tree ensembles. In *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, pages 1137 –1140, april 2010.

[109] J. J. Rodriguez and L. I. Kuncheva. Naive bayes ensembles with a random oracle. In *Proc 7th International Workshop on Multiple Classifier Systems*, LCNS 4472, pages 450–458, 2007.

[110] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1619 –1630, October 2006.

[111] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, 1962.

[112] D. Sculley, G. M. Wachman, and C. E. Brodley. Spam filtering using inexact string matching in explicit feature space with on-line linear classifiers. In *In The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*, 2006.

[113] M. Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, 2001.

[114] Y. I. Sheline, D. M. Barch, J. M. Donnelly, J. M. Ollinger, A. Z. Snyder, and M. A. Mintun. Increased amygdala response to masked emotional faces in depressed subjects resolves with antidepressant treatment: An fMRI study. *Biological Psychiatry*, Volume 50(9(1)):651–658, 2001.

[115] R. Sitaram, N. Weiskopf, A. Caria, R. Veit, M. Erb, and N. Birbaumer. fmri brain-computer interfaces. *Signal Processing Magazine, IEEE*, 25(1):95 –106, 2008.

[116] M. Skurichina and R. P. W. Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5:121–135, 2002.

[117] D. Sona and P. Avesani. Feature ratings by random subspace for functional brain mapping. In *LCNS: Brain Informatics: International Conference*, volume 6334, pages 112–123, 2010.

[118] D. Sona and P. Avesani. Multivariate brain mapping by random subspaces. In *International Conference on Pattern Recognition*, 2010.

[119] W. N. Street and Y. Kim. A streaming ensemble algorithm (sea) for largescale classification. In *in Proceedings of the seventh ACM SIGKDD international*

conference on Knowledge discovery and data mining, pages 377–382, New York, NY, USA., 2001. ACM Press.

[120] S. C. Strother. Evaluating fMRI preprocessing pipelines. *Engineering in Medicine and Biology Magazine, IEEE*, 25(2):27–41, 2006.

[121] S. Sun, Y. Lu, and Y. Chen. The stochastic approximation method for adaptive bayesian classifiers: towards online brain-computer interfaces. *Neural Computing and Applications*, 20:31–40, 2011.

[122] S. Sun, C. Zhang, and D. Zhang. An experimental evaluation of ensemble methods for EEG signal classification. *Pattern Recognition Letters*, 28(15):2157 – 2163, 2007.

[123] M. van Gerven, J. Farquhar, R. Schaefer, R. Vlek, J. Geuze, A. Nijholt, N. Ramsay, P. Haselager, L. Vuurpijl, S. Gielen, and P. Desain. The brain-computer interface cycle. *Journal of Neural Engineering*, 6(4):041001, 2009.

[124] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[125] B. Wang, G. Jones, and W. Pan. Using online linear classifiers to filter spam emails. *Pattern Analysis and Applications*, 9:339–351, 2006.

[126] X. Wang and X. Tang. Random sampling LDA for face recognition. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–259 – II–265 Vol.2, 2004.

[127] Z. Wang, A. R. Childress, J. Wang, and J. A. Detre. Support vector machine learning-based fMRI data group analysis. *NeuroImage*, 36(4):1139 – 1151, 2007.

[128] N. Weiskopf, K. Mathiak, S. W. Bock, F. Scharnowski, R. Veit, W. Grodd, R. Goebel, and N. Birbaumer. Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). *IEEE Transactions on Biomedical Engineering*, 51:966–970, 2004.

[129] N. Weiskopf, F. Scharnowski, R. Veit, R. Goebel, N. Birbaumer, and K. Mathiak. Self-regulation of local brain activity using real-time functional magnetic resonance imaging (fMRI). *Journal of Physiology Paris*, 98:357–373, 2004.

[130] N. Weiskopf, R. Sitaram, O. Josephs, R. Veit, F. Scharnowski, R. Goebel, N. Birbaumer, R. Deichmann, and K. Mathiak. Real-time functional magnetic resonance imaging: methods and applications. *Magnetic Resonance Imaging*, 25:989–1003, 2007.

[131] O. Yamashita, M. Sato, T. Yoshioka, F. Tong, and Y. Kamitani. Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns. *NeuroImage*, 42:1414–1429, 2008.

[132] Jian Yang, Ning Zhong, Peipeng Liang, Jue Wang, Yiyu Yao, and Shengfu Lu. Brain activation detection by neighborhood one-class SVM. *Cognitive Systems Research*, 11(1):16–24, March 2010.

[133] S. S. Yoo, T. Fairneny, N. K. Chen, S. E. Choo, L. P. Panych, H. W. Park, S. Y. Lee, and F. A. Jolesz. Brain–computer interface using fMRI: spatial navigation by thoughts. *NeuroReport*, 15(10):1591–1595, 2004.

[134] J. Younger, A. Aron, S. Parke, N. Chatterjee, and S. Mackey. Viewing pictures of a romantic partner reduces experimental pain: Involvement of neural reward systems. *PLos One*, 5(10):e13309, 2010.

[135] X. Zhang and Y. Jia. A linear discriminant analysis framework based on random subspace for face recognition. *Pattern Recognition*, 40:2585–2591, September 2007.

[136] Y. Zhu, J. Liu, and S. Chen. Semi-random subspace method for face recognition. *Image and Vision Computing*, 27(9):1358 – 1370, 2009.