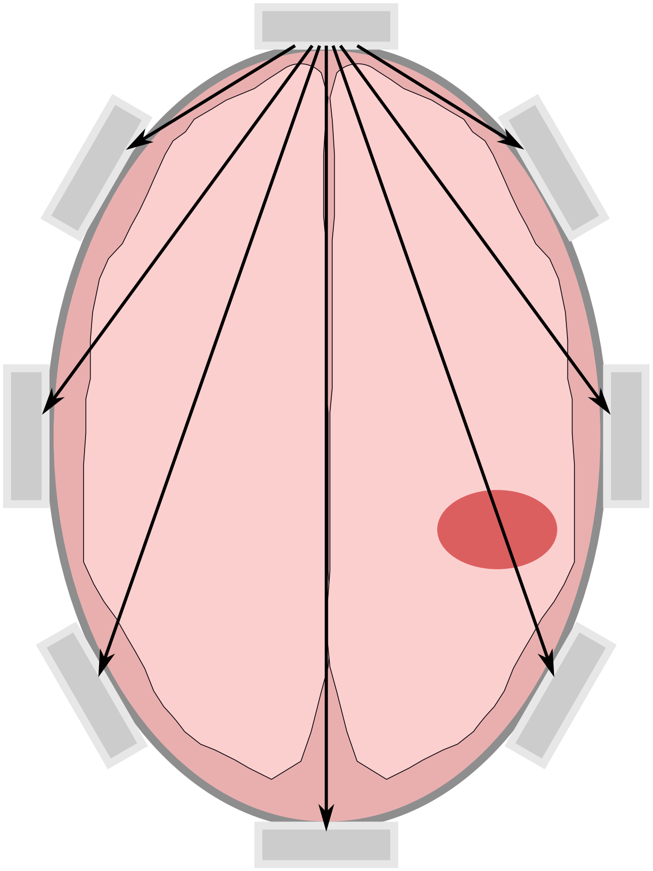
CHALMERS



Machine Learning Algorithms for Stroke Diagnostics

Master's thesis in Biomedical Engineering
EX019/2014

CHRISTOFFER SUNDSTRÖM

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2014

Machine Learning Algorithms for Stroke Diagnostics.
Christoffer Sundström.

Gothenburg, Sweden, 2014

Cover page: Schematic figure of a head surrounded with microwave antennas. The top antenna is sending and the others are receiving. In the right hemisphere, a bleeding is located that alters the wave propagation.

ACKNOWLEDGEMENTS

I may be the author of this thesis, but that does not mean I did it alone. First of all, I want to thank my excellent supervisor at Medfield Diagnostics, Stefan Candefjord. Without your support and the constructive ideas I wouldn't have gotten this far. Another special thanks goes out to my other supervisor and examiner, Professor Tomas McKelvey for the guidance in this jungle of classification and the opportunity to have a meeting every other week for inspiring and valuable discussions.

I have spent almost six months at Medfield Diagnostics and I am kind that I got this opportunity. I am also grateful for having such wonderful co-workers and therefore I want to thank you all; Dag Jungenfelt, Fabian Wenger, Lennart Nilsson, Stefan Kidborg, Antti Stålnacke, Jasila Prabahar, Tobias Olsson and Samuel Lunde. Thanks for all joyful times during lunch and coffe breaks as well as providing a working environment to thrive in.

Six months might seem like a long time, but when I look back I realise that time flies. This is also my fifth and final year at Chalmers University of Technology. In some way this is the culmination of five years of studies that finally comes to an end.

I will also not miss the opportunity to thank all my new friends that I have met during my studies at Chalmers, without you I do not think that I would have even completed my first year.

And last but not least, I want to thank my family for supporting and believing in me throughout the years. I could never have done this without you.

Christoffer

Abstract

Stroke is the most common cause of disability in adults and one of ten leading causes of death in the world. It is estimated that in year 2030, stroke will be one of the four leading causes of death. However, the chances to avoid permanent disability greatly increases when treatment is given quickly after stroke onset.

A stroke occurs when parts of the brain suffer from insufficient oxygen supply due to either a blood clot (ischaemic stroke, IS), or a ruptured vessel causing an intracranial haemorrhage (ICH). Medfield Diagnostics is developing a microwave based technique which can identify the difference in dielectric properties of tissues. There exist dielectric differences between blood and ischaemic tissues and the hypothesis is that healthy brain tissue can be distinguished from blood and ischaemic brain tissue. This technique shows promise for distinguishing ICH from IS patients. The equipment's relatively small size compared to computed tomography (CT) equipment, which is commonly used for diagnosis today, makes it suitable to be used in the ambulance. Thereby it has potential to shorten the time from stroke onset to diagnosis and accordingly decrease the time from onset to treatment.

Presently a subspace based classifier is used for ICH detection, the *inner-product subspace classifier* (ISC). The goal of this master's thesis was to investigate if different classification algorithms could be utilised instead of or together with the ISC to increase the classifier performance. The classification algorithms evaluated were *support vector machines* (SVM) and a metric learning algorithm called *large margin nearest neighbour* (LMNN).

The data used in this study were provided by Medfield Diagnostics. Three measurements are present for each patient and evaluation of the classifier algorithms could therefore be carried out using a *pseudo Monte Carlo procedure* to extend the diversity of the dataset and to get a good statistical foundation. The performance measure for comparison was the area under the receiver operating characteristic (ROC) curve (AUC). For LMNN no ROC-curve could be produced due to its nonparametric nature and therefore it was evaluated by its accuracy, sensitivity and specificity. A bootstrap method was used to derive standard deviation for the performance measures. An amount of 100 Monte Carlo simulations were performed. Standard deviations were derived by randomly selecting 99 Monte Carlo simulations and repeating the procedure 50 times.

For the ISC, SVM with a linear kernel and SVM with a (non-linear) RBF-kernel, the AUC was found to be 0.86, 0.67 and 0.87 respectively. The accuracy, sensitivity and specificity for the LMNN algorithm was found to be 0.68, 0.36 and 0.89. The standard deviation for all measures was lower than 0.05. It shall be taken into consideration that the frequency intervals differ for the methods and the results are valid only within distinct frequency intervals. The outcome from the SVM needs to be further investigated to verify that there is no overtraining in the SVM algorithms.

In order to further increase the classification performance we propose to investigate impact of different preprocessing procedures and existing reference measurements. Combining the methods of SVM and metric learning has in recent studies shown to be effective. We therefore propose to investigate the *support vector metric learning* (SVML) algorithm.

Keywords: stroke, binary classification, microwave technology, biomedical engineering, support vector machines, SVM, large margin nearest neighbour, LMNN

Nomenclature

AUC Area under the receiver operating characteristic curve. Measurement on performance of a binary classifier.

CT Computed tomography. A type of X-ray imaging technique.

DALY *Disability-adjusted life years.* A measure of the personal suffering due to disability.

EM Electromagnetic. Mostly in the context of EM waves.

FN False negative.

FP False positive.

ICH *Intracerebral haemorrhagic stroke*, caused by a bleeding inside the brain.

IS *Ischaemic stroke*, caused by a blood clot in the brain.

ISC Inner-product subspace classifier. A classifier based on a LSM.

KFCV *k-fold cross-validation.*

LMNN *Large margin nearest neighbour*, a type of kNN classification algorithm.

LOOCV *Leave one out cross-validation.*

LPOCV *Leave pair out cross-validation.*

LSM Linear subspace model. A mathematical model upon which the ISC is based.

LSVM *Linear* SVM.

MC Denotes Monte Carlo, a concept in statistical analysis.

MHMS Microwave head measurement system.

MRI Magnetic resonance imaging. Imaging technique.

MWT Microwave technology.

RBFSVM SVM using the *Radial Basis Function* kernel.

ROC Receiver operating characteristic. Performance curve of a binary classifier.

rtPA Recombinant tissue plasminogen activator. Clot dissolver given intravenously.

SVM *Support vector machines*, a classification concept.

TN True negative.

TP True positive.

TT Thrombolytic treatment. Usage of ”clot busters” such as rtPA.

VNA Vector network analyser. Hardware that measures S-parameters.

WHO World Health Organization.

Table of Contents

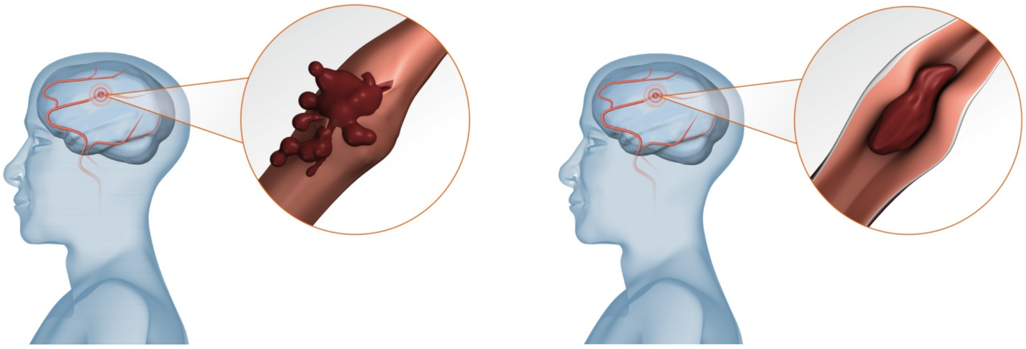
Chapter 1

Introduction

Cerebrovascular disease (stroke) was in 2004 ranked as one of the ten leading causes of death according to the World Health Organization (WHO) and is projected to be one of the four leading causes of death in 2030 [1], [2]. Stroke is also the most common cause of permanent disability in human adults. In the United States, every 45 seconds someone suffer a stroke and every three minutes someone dies due to a stroke [3].

1.1 What is stroke?

The brain relies upon the unobstructed flow of blood to provide glucose and oxygen as well as to remove waste products. During a stroke, the blood flow to the brain is disturbed which leads to inadequate blood supply [4]. A stroke can be either haemorrhagic or ischaemic, the former meaning a rupture of a blood vessel causing blood to enter the intracranial cavity (intracranial haemorrhage, ICH) and the latter meaning obstruction by a blood clot (ichaemic stroke, IS) in the intracranial arteries [5], see Figure 1.1.



(a) Intracranial haemorrhage due to a burst vessel.

(b) Blood clot hindering blood flow.

Figure 1.1: Intracranial haemorrhage compared to ischaemic stroke. Image used with kind permission from Medfield Diagnostics.

Ichaemic stroke accounts for a majority of all strokes [4], approximately 85 %. Despite ICH not being the most common form of stroke, it accounts for 51.7 % of the stroke-related deaths [6].

In recent years the knowledge of stroke has increased and revealed that neuronal death in stroke is time-dependent [5]. For strokes caused by a blood clot one effective treatment is to use medication that dissolves the clot [7]. With the introduction of the ”clot dissolver” *recombinant tissue plasminogen activator*, rtPA, a concept called *time is brain* emerged and stroke care is therefore to be more focused on prehospital care. Common today is a chain of events as illustrated in Figure 1.2 in which stroke onset is followed by an emergency call, transport to the hospital and investigation using CT. The diagnosis is made by a doctor by looking at the acquired images.

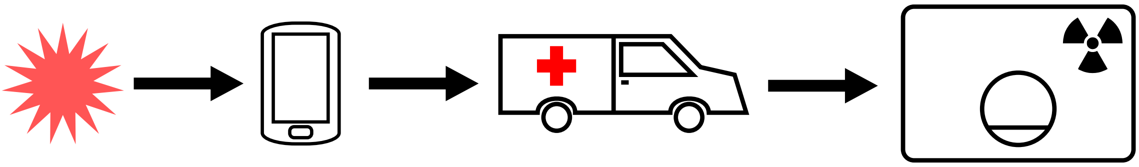


Figure 1.2: Sequence of events during a stroke. Stroke onset is generally followed by an emergency call. The ill person is taken with the ambulance to the hospital in which CT examination is performed. The CT images are investigated by a doctor who issues the treatment.

It is known that thrombolytic treatment (TT) administered within a specific therapeutic window is an effective therapy for ischaemic stroke. However, due to prehospital delay no more than 1–8 % of all patients in need of thrombolytic treatment obtain it in time [7].

It comes naturally that in order to begin treatment faster, the diagnosis must be made earlier.

1.2 Impact on personal life and society

The impact of a stroke (or any other injury) in a person's life is measured by the WHO using *disability-adjusted life years*, DALYs, and this measure accounts for healthy life years lost due to living with disability as well as premature mortality [8] Stroke is the third leading cause of DALYs worldwide and the global burden due to it is increasing [6].

Apart from the individual suffering, the costs for society are huge [7]. A study concerning the region of Västra Götaland with 1.5 million inhabitants in 2008 found that there were 3074 people having a first-ever stroke [9]. The excess costs due to this was estimated to 629 million SEK.

1.3 Stroke diagnosis and treatment today

IS and ICH cannot be distinguished based on its symptoms, as they are very similar [10] and as of today some type of brain imaging is required to do this. *Computed tomography* (CT) or *magnetic resonance imaging* (MRI) is used to discriminate between ischaemic and haemorrhagic stroke [11] and of those, CT is the most common [12]. Both CT and MRI provide useful information on tissue properties that might be related to malignancies [3]. However, they have drawbacks such as exposure to ionizing radiation for CT and for MRI where the examination time is long. In MRI it is sometimes also an issue with claustrophobia, even if it is not frequent [13].

A common drawback related to stroke detection is the cost-efficiency and their absence in portability due to their apparatus sizes. However, there have been trials with custom made ambulances equipped with CT-scanners for the purpose of stroke detection [4], but this does not solve the issues of cost-efficiency or the ionising radiation. The ionising radiation furthermore makes the technique unsuitable for continuous monitoring.

For patients suffering from ICH there is no universal treatment today [10]. For IS, on the other hand, thrombolytic treatment is used. However, TT have a negative effect on ICH patient due to risk of increased bleeding and ICH must therefore be ruled out before TT is administered.

1.4 Medfield Diagnostics' proposed solution

Medfield Diagnostics' proposed solution is to use microwave technology, MWT, to perform stroke diagnosis due to its possibility in prehospital diagnosis in e.g. ambulances unlike CT or MRI. As the apparatus size of the proposed microwave system is significantly smaller than CT and MRI machines, the portability is increased. The cost of a complete system will also be much lower in comparison. A schematic view of the measurement concept is shown in Figure 1.3.

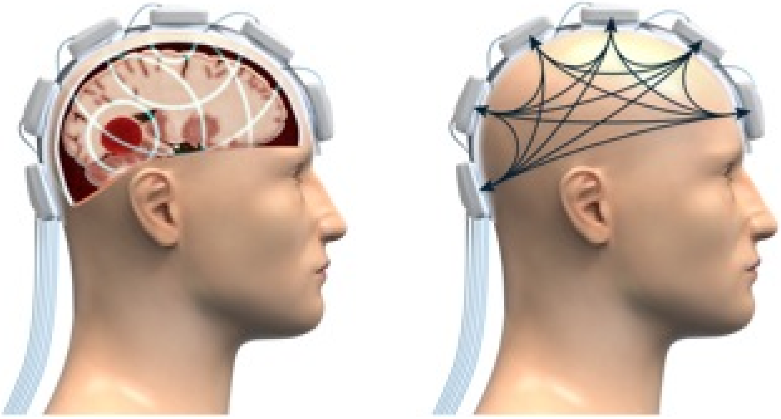


Figure 1.3: The concept of Medfield Diagnostics' solution. Microwaves are sent and received using antennas placed around the head. Image used with kind permission from Chalmers University of Technology.

Due to the portability and lower cost, the aim is that usage of this technique in ambulances will be able to lower the prehospital delay.

MWT has been demonstrated to be applicable for brain imaging [3], [4], which might indicate that the technology proposed by Medfield Diagnostics is on the right track. It shall though be pointed out that the proposed solution is not to use microwave as a tomographic technique[1], but the theoretical concepts still apply.

The underlying postulate for using microwave technology to discriminate a haemorrhagic from an ichaemic stroke is the difference in conductivity and permittivity of different tissues. It is known that different tissues have different dielectric properties [14], [15]. These properties determine how the electromagnetic wave propagate through them [16]. Generally, by sending a microwave through tissue, some information of the wave propagation can be gathered, giving information about the tissue it has propagated through.

Microwave measurements are performed over a defined range of frequencies. The whole frequency span of interest is discretised into smaller frequency spans to be able to identify frequency regions that yield better classification performance.

The clinical data used in this project were provided by Medfield Diagnostics.

[1] I.e. the technique is not used to produce an image.

1.5 Classification

Classification or *pattern recognition* is the process of distinguishing objects or items from one another. At first glance, this might not seem like an extensive task, however finding complex patterns in data is easier said than done. The concept of classification used throughout this thesis is shown in Figure 1.4.

Input data
Classifier
Predicted class

Figure 1.4: The concept of classification.

Classification is hence the concept of mapping input data to a specific class. The classifier does this mapping by a set of rules and in this thesis the rules are described using *training* data. Training data is data associated with a specific class and the classifier algorithm uses information acquired from this data to predict the class of future data.

Evaluation of a classifier can be done in various ways. In this thesis, the classifier is trained with one set of data and is then to predict outcome of *validation* data. Validation data in this case means data that belongs to a known class, but is unknown to the classification algorithm. The predicted class of the classifier can then be compared to the true class of the data.

1.5.1 The curse of dimensionality

The dimensionality can be thought of as the number of variables that describe an object. For example, in the digital world, colors are represented by fractions of the colours red, green and blue and thus can be thought of as three dimensional. In our case, the dimensionality is determined by how many variables that defines our measurement and is the number of discrete frequency intervals we consider. The overall dimensionality of one measurement is acquired by the number of frequency intervals considered and the number of antenna combinations considered.

There are problems associated with data of high dimensions given a small sample size, the problem referred to as HDLSS. High dimensions and small sample sizes are relative notions and commonly HDLSS is the case when the number of dimensions in the input data is greater than the sample size [17], [18]. In this case, the dimensionality is in the order of thousands, while the size of the datasets includes around 100 patients at most and therefore is considered a HDLSS problem.

The goal with the ISC is to extract features, characteristic patterns in the data, that can be used to discriminate between intracranial haemorrhage and ischaemic stroke. This is done according to a *linear subspace model*[2] and in general it can be thought of as first deriving subspaces corresponding to ICH and IS respectively. Then, a measurement is compared to each of the subspaces yielding a size within the subspace that can be thought of as a similarity measure where the similarity is better the larger the size. The numerous dimensions has then been reduced to use only two dimensions, the size within each of the

[2] Linear subspaces are mathematical constructions defined by vectors. In this case, the two subspaces are created based on the microwave measurements. Similar measurements are generally larger in the subspace to which they belong.

subspaces. In the ISC, classification is done by comparing the sizes within the subspaces and assign the measurement the label of which subspace it is most similar to. This is illustrated in Figure 1.5.

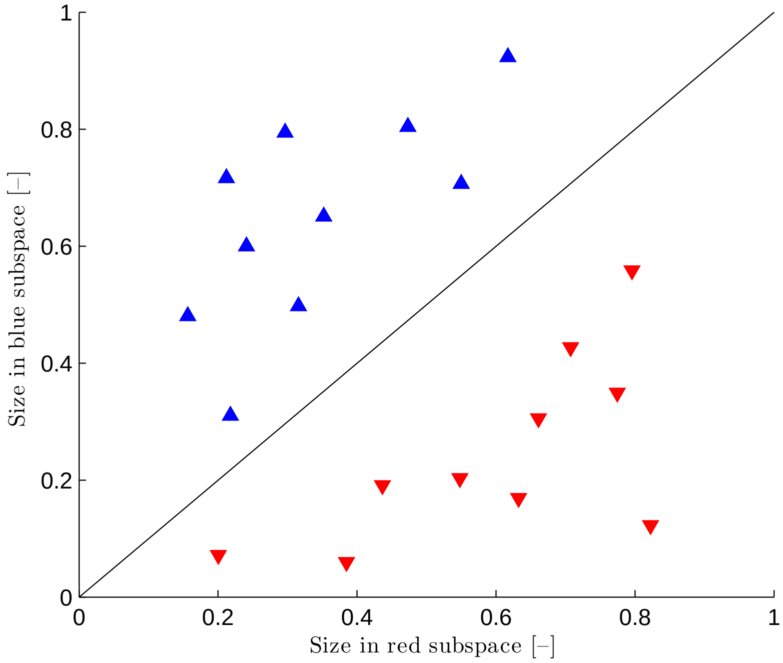


Figure 1.5: Classification using most similar subspace. Above the solid line, the decision boundary, data belongs to the blue class and below, data belongs to the red class.

1.5.2 Support vector machines

Support vector machines, SVM, is a technique in which a decision boundary (or more strictly, a hyperplane) that optimally separates data of two classes is acquired. In this sense, the *optimal* separating hyperplane is the hyperplane that acquires maximum separation between the two classes. It might be the case that the two classes differ in the similarity measurements, but that the most similar subspace is not necessarily a good discriminator. An example is shown in Figure 1.6.

The idea is to investigate if using the SVM algorithm can improve the classification outcome. For example, in Figure 1.6, linear SVM perfectly separates the red class from the blue class with the decision boundary. Everything that falls on the right side of this hypothetical line would be classified as red, while everything that falls on the left side will be classified as blue.

1.5.3 Large margin nearest neighbour

Large margin nearest neighbour, or LMNN, is a classification algorithm based on the k-nearest neighbour, kNN, classification algorithm [19]. The kNN classification algorithm simply assigns a label to an unclassified data point based on the k closest neighbouring data points' class labels [20], as seen in Figure 1.7.

Of course, the closest neighbours depends on how the metric is defined when computing the distance [19]. ”Standard” kNN classification uses the Euclidean metric while the idea with LMNN is to use a different metric but keeps the underlying principle of the

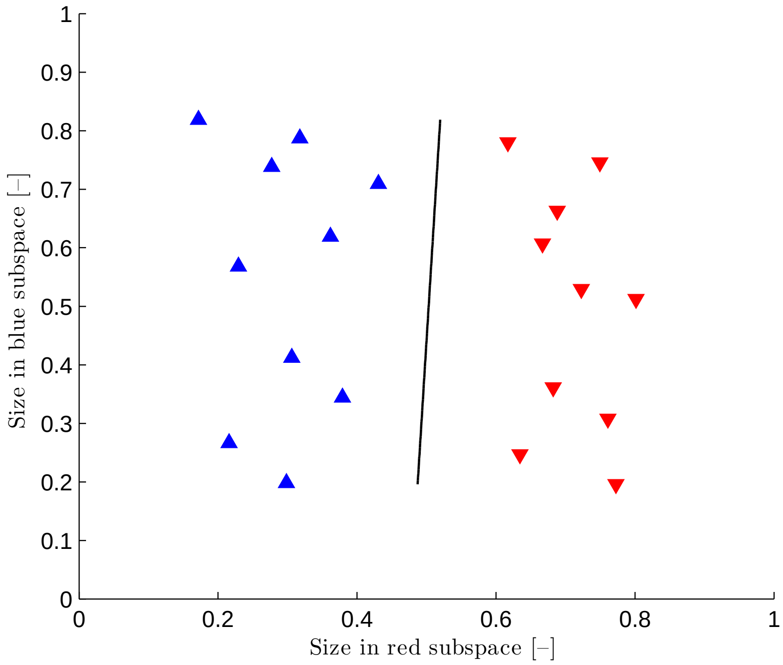


Figure 1.6: Classification with a diagnonal decision boundary (as in Figure 1.5) is not efficient, so there exists a better decision boundary. Linear SVM suggests the solid line as decision boundary.

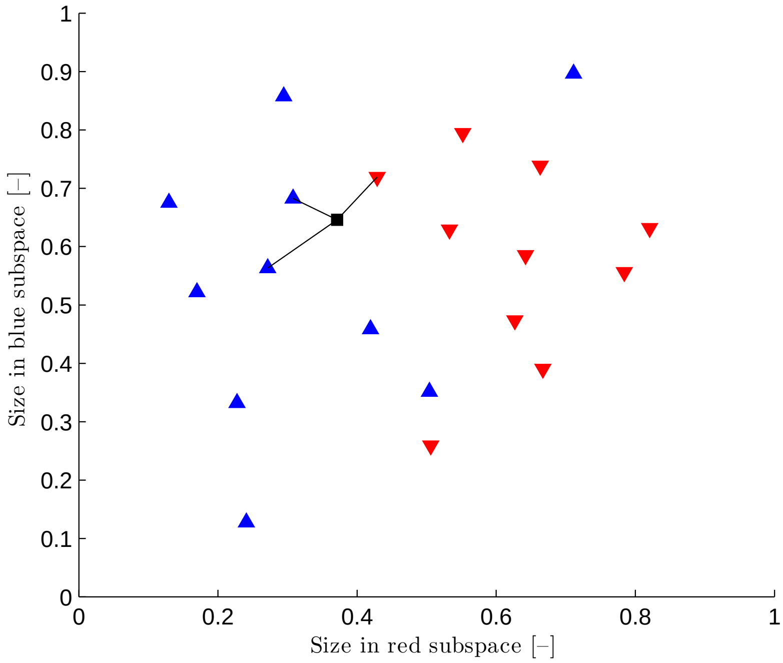


Figure 1.7: Classification using kNN with $k = 3$. The unclassified measurement (black square) is connecting to its three nearest neighbours, of which two belongs to the blue class and one to the red class. The algorithm will therefore predict the class label to be blue.

kNN algorithm. The metric being used is a so-called *Mahalanobis* metric and the goal in LMNN is to identify a metric in which similarly labelled points appear closer than differently labelled points.

1.6 Aims and goals

The aims of this project are to see if the classification performance can be increased by using other classifiers instead of or together with the method currently in use by Medfield Diagnostics. Clinical data provided by Medfield Diagnostics will be used throughout this study. The aims and goals can be summarised as:

- Study different machine learning algorithms for use in stroke diagnostics using microwave measurements:
  - Study performance using ISC
  - Study performance using linear SVM
  - Study performance using SVM with an RBF-kernel
  - Study performance using the metric learning algorithm LMNN.
- Restructure and improve the classification program
  The code should be rewritten to become more modular and to make data analysis easier. Different classifiers should be easy to test without changing the fundamentals of the code. It is also of interest to produce a program that has capability of parallel processing to utilise multi-core processors and also make it suitable for high performance computing clusters.

1.6.1 Scope

The implementations will be made in Matlab, as previous work has been made in this language, but also due to the versatility and speed in these kind of problems.

The underlying feature extraction concept built upon the LSM and *singular value decomposition* will be kept, while the classification algorithms following this step is what is going to be investigated, see Figure 1.4.

Analysis and evaluation of the different machine learning algorithms will be performed using a PC and not using a computer cluster.

Chapter 2

Theory

In this chapter the underlying concepts and theories that are of importance for this project are presented. It starts with a section on microwave technology that gives the background to how the measurements are performed and how they work. This section is then followed by a section dealing with classification.

2.1 Microwaves

Like visible light and radio waves, microwaves are electromagnetic (EM) waves [16]. What differs microwaves from other electromagnetic waves are its frequency and wavelength. The frequency f and wavelength λ is related by the speed of light in the considered media, c, as seen in Equation (2.1).

$$f = c\lambda \tag{2.1}$$

As can be derived from the name, electromagnetic waves consists of an electric and a magnetic component. These components are perpendicular to each other as well as the direction of the wave propagation [16], as can be seen in Figure 2.1.

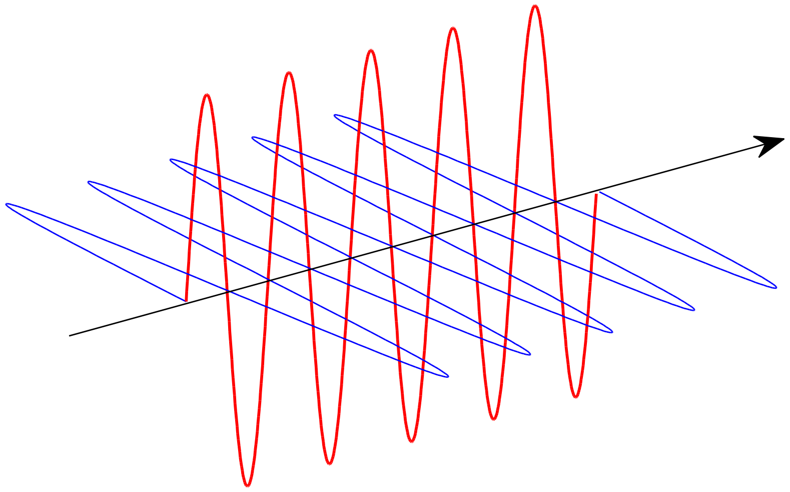


Figure 2.1: Electromagnetic waves propagating along the black arrow. The electric and magnetic components are perpendicular to one another as well to the direction of propagation.

As previously mentioned it is of interest how microwaves propagate through different materials (in this context different tissue types), something that is related to a material's dielectric properties. The dielectric properties and how they relate to EM wave propagation is described by Maxwell's equations published by James Clerk Maxwell in 1873. These equations describe the magnetic and electric phenomena at a macroscopic level and are listed in Equations (2.2a-d) [16].

$$\nabla \times \bar{\mathcal{E}} = \frac{-\partial \bar{\mathcal{B}}}{\partial t} - \bar{\mathcal{M}} \tag{2.2a}$$

$$\nabla \times \bar{\mathcal{H}} = \frac{-\partial \bar{\mathcal{D}}}{\partial t} + \bar{\mathcal{J}} \tag{2.2b}$$

$$\nabla \cdot \bar{\mathcal{D}} = \rho \tag{2.2c}$$

$$\nabla \cdot \bar{\mathcal{B}} = 0 \tag{2.2d}$$

The notation used in these equations are explained in Table 2.1 for clarity. The relation between the flux and field densities in free-space is given according to the constitutive relations described in Equations (2.3a) and (2.3b) [16], where $\mu_0 = 4\pi \times 10^{-7}$ Henry/m and $\epsilon_0 = 8.854 \times 10^{-12}$ farad/m is the permeability and permittivity of free-space respectively.

Table 2.1: Key chart for Maxwell's equations

Symbol	Description	SI-unit
$\bar{\mathcal{E}}$	electric field	V/m
$\bar{\mathcal{H}}$	magnetic field	A/m
$\bar{\mathcal{D}}$	electric flux density	C/m^2
$\bar{\mathcal{B}}$	magnetic flux density	Wb/m^2
$\bar{\mathcal{M}}$	fictitious magnetic current density	V/m^2
$\bar{\mathcal{J}}$	electric current density	A/m^2
ρ	electric charge density	C/m^2
t	time	s

$$\bar{\mathcal{B}} = \mu_0 \bar{\mathcal{H}} \tag{2.3a}$$

$$\bar{\mathcal{D}} = \epsilon_0 \bar{\mathcal{E}} \tag{2.3b}$$

Maxwell's equations presented in Equations (2.2a-d) together with the relations given in Equations (2.3a-b), describes EM waves for arbitrary time dependence travelling through free-space. By assuming sinusoidal time dependence and steady-state, these equations can be written in a more convenient form called *phasor notation* [16]. In this notation, all field quantities is written as complex vectors implied with a $e^{j\omega t}$ time-dependency, where j is the imaginary number, $\omega = 2\pi f$ is the angular frequency and t the time . Applying this phasor notation on the equations in (2.2a-d) yield the equations in (2.4a-d).

$$\nabla \times \bar{E} = -j\omega \bar{B} - \bar{M} \tag{2.4a}$$

$$\nabla \times \bar{H} = -j\omega \bar{D} + \bar{J} \tag{2.4b}$$

$$\nabla \cdot \bar{D} = \rho \tag{2.4c}$$

$$\nabla \cdot \bar{B} = 0 \tag{2.4d}$$

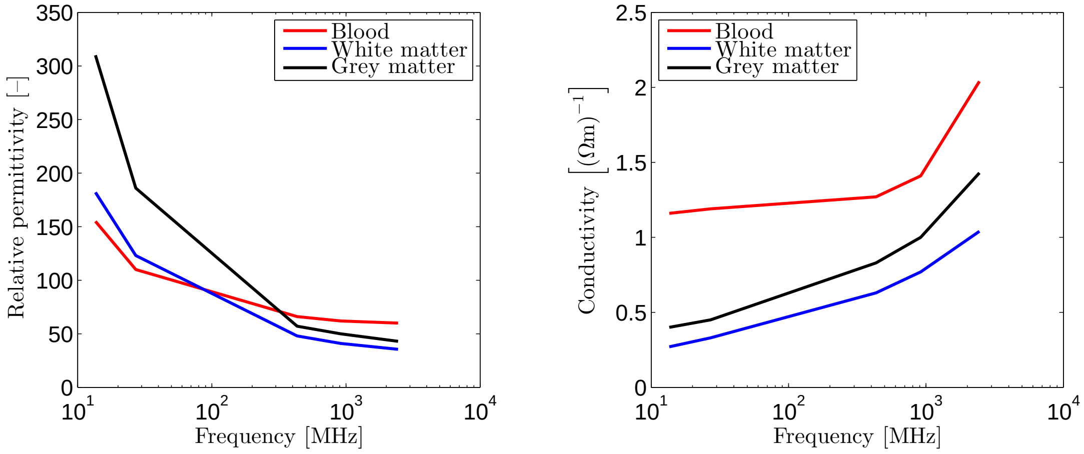
In order to account for fields travelling through media, the relations given in Equations (2.3a-b), has to be slightly modified, yielding Equations (2.5a-c) [16].

$$
\begin{aligned}
\bar{B} &= \mu \bar{H} && \text{(2.5a)}\\
\bar{D} &= \epsilon \bar{E} && \text{(2.5b)}\\
\bar{J} &= \sigma \bar{E} && \text{(2.5c)}
\end{aligned}
$$

The permeability μ and the permittivity ϵ relates to how well a magnetic field and an electric field can propagate in a material respectively. The conductivity σ of the material relates to how well current flows within the material and is no longer zero as for free space.

For biological materials, it has been noted that the permeability μ is close to the permeability of free space [21]. However, for the proposed application it is of interest to study the impact on the electric field $\bar{E}$ which relates to the permittivity ϵ and conductivity σ.

These dielectric properties are known to be frequency dependent, and this fact is illustrated in Figure 2.2 for some interesting tissue types: blood, white matter and gray matter. Gray matter consist of neuronal cell bodies among other brain cell types, while the white matter is primarily composed of myelinated axons that connects the neurons to each other [22].



(a) Permittivity for tissues

(b) Conductivity for tissues

Figure 2.2: Permittivity and conductivity for white matter, grey matter and blood relative to frequency. Figure adapted from Pethig (1984) [15].

As can be seen in Figure 2.2a, permittivity decreases with increasing frequency for the three tissue types shown. The conductivity shown in Figure 2.2b however, increases with increasing frequency.

It has been shown that ischemia (lack of blood) alters the dielectric properties of tissue [23], [24]. As the blood flow during a stroke is altered compared to healthy oxygenated tissue, this will affect the composition of tissues within the head. The underlying principle for using microwave technique to distinguish between ICH and IS is therefore to identify this difference in tissue composition. During an ICH there will be more blood compared to normal tissue, wheras during an IS there will be ischaemic tissue.

2.1.1 S-parameters

When measurements are performed, it is of interest to study how much of the incident voltage is received at another antenna, which will give a measurement of the attenuation and scattering of the wave during its propagation.

A convenient way of relating incident voltage of antenna i to reflected voltage of antenna j for a wave of defined frequency is the scattering parameter, or *S-parameter*, as defined in Equation (2.6), where V_j^+ and V_i^- denotes incident voltage of antenna j and reflecting voltage of antenna i respectively [16].

$$S_{ij} = \left.\frac{V_i^-}{V_j^+}\right|_{V_k^+=0\ ,\ k\neq j} \tag{2.6}$$

Considering a network of N antennas, these relationships can be described using vector notation as seen in Equation (2.7) [16].

$$\begin{bmatrix} V_1^- \\ V_2^- \\ \vdots \\ V_N^- \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1N} \\ S_{21} & & & \\ \vdots & & \ddots & \\ S_{N1} & & & S_{NN} \end{bmatrix} \begin{bmatrix} V_1^+ \\ V_2^+ \\ \vdots \\ V_N^+ \end{bmatrix} \tag{2.7}$$

where the matrix containing S-parameters is denoted the *scattering matrix.*

The S-parameters can be subdivided into two disjoint groups, *reflection* coefficients and *transmission* coefficients. Reflection coefficients are the S-parameters in which the reflecting and transmitting antenna is the same and is equivalent to the diagonal elements S_{ii} in the scattering matrix [16].

The measurements performed by Medfield Diagnostics measures the S-parameters directly using a *vector network analyzer*, or *VNA*, and is the raw data used for distinguishing between ICH and IS. For each frequency, one set of S-parameters is measured. The measurement setup is illustrated in Figure 2.3.

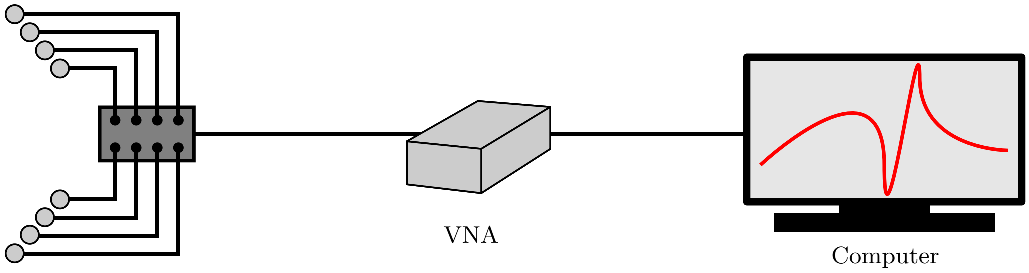


Figure 2.3: Schematic figure of the measurement setup. The eight antennas are connected to a switchboard which in turn is connected to the VNA. The VNA is connected to the computer that operates the VNA as well as stores the measurements for future analysis.

2.2 Classification

In this project, classification is the process of distinguishing a patient's disorder as being an ICH or IS by using S-parameter measurements performed on that very patient.

The goal is to assign a class label to a measurement using a *supervised learning* algorithm. Supervised learning means that the algorithm uses *training data* to acquire a discrimination function to predict (or classify) other data [25]. The data that the machine learning algorithm uses to discriminate objects are commonly called features.

An object is associated with any number of features and can be thought of as a characteristic property that can be used to distinguish one type of object from another. It shall be noted that a feature is not uniquely defined which means that features can be extracted in numerous ways and may differ from application to application. – A feature can simply be any property of an object. In classification it is preferably selected so that it discriminate the classes we want to separate. One can also make use of several features for every object. The procedure of extracting features is commonly called *feature extraction* [26].

The goal of any supervised learning algorithm can therefore be thought of as predicting a class label of an unknown object based on its features.

2.2.1 Binary classification

In binary classification there are two discrete groups or classes $\mathcal{C} := \{+1, -1\}$. Suppose there is a function $f(x)$ that discriminates the two classes with labels $c \in \mathcal{C}$, this is mathematically described in Equation (2.8).

$$c = f(x) \tag{2.8}$$

where c is the class label for the measurement x. As the function $f(x)$ is related to some underlying model describing the data [25] and is not perfectly known (in which case a classification algorithm would be unneccesary), the aim of a classifier is to approximate f with a function $\hat{f}$ to acquire a predicted class label $\hat{c}$, as in Equation (2.9).

$$\hat{c} = \hat{f}(x) \tag{2.9}$$

In supervised learning, this approximation is done by training the algorithm with data that has a known class label. How the algorithm deduces the function $\hat{f}(x)$ depends on the implementation of the algorithm and can be done in various ways.

Performance

The performance of a classifier is a measure of how well it performs its task and is commonly described in terms of probabilities, e.g. probability of detection (true positive rate) or probability of false alarm (false positive rate).

The outcome from a classification can be categorised as correct or incorrect, normally denoted *true* or *false* respectively. Consider c being the true label of the measurement x and, as before, $\hat{f}(x)$ being the predicted class label, a true outcome is defined as in Equation (2.10a) and a false outcome as in Equation (2.10b).

$$\hat{f}(x) = \hat{c} = c \tag{2.10a}$$

$$\hat{f}(x) = \hat{c} \neq c \tag{2.10b}$$

For a binary classifier the predicted class can be either positive (+1) or negative (−1). Considering that there are two different outcomes and that the classification can be true (correct) or false (incorrect), this will yield four outcomes in total, which is shown in Table 2.2 [27]. The matrix made up in the same manner as in Table 2.2 where on one axis

Table 2.2: Table over different outcomes of a binary classifier. TP = *true positive*, FP = *false positive*, *FN* = *false negative*, TN = *true negative*.

		True class	
		+1	−1
Predicted class	+1	TP	FP
	−1	FN	TN

is the predicted outcome and the ground truth on the other, is sometimes referred to as a *confusion matrix.* The efficiency of a classification algorithm can be reported in various ways using different measures, such as *accuracy*, *sensitivity* and *specificity.* All of these measures relate to the confusion matrix.

Accuracy

The *accuracy* measure of a classification outcome is the amount of correctly classified over the total amount of objects classified [28], or mathematically as in Equation (2.11).

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.11}$$

This measure works well for classifiers outputting a score from which the predicted class label is determined or simply just a predicted class label. The accuracy measure can be thought of as the empirical probability of getting a correct classification outcome. It shall though be noted that the accuracy measure must be interpreted with the class distribution in mind. Classifying everything as one class, will essentially give an accuracy equal to the distribution of this class. For example, out of 100 measurements there are 68 of class A and 32 of class B. Classifying everything as class A will yield an accuracy of 68 %.

Sensitivity and specificity

Classifier performance in often reported as *sensitivity* and *specificity*, which relates to the classifier's ability to identify positive and negative results respectively [25]. Sensitivity is sometimes referred to as *true positive rate* or *probability of detection* (*TPR*) and specificity as *true negative rate* (*TNR*) [27]. Mathematically, these properties are defined as in Equations (2.12a) and (2.12b).

$$\text{Sensitivity} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.12a}$$

$$\text{Specificity} = \text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2.12b}$$

Some classifiers have a tuning parameter that alters the outcome of the classification which can be used to trade sensitivity for specificity. This concept is illustrated in Figure 2.4 in which the two classes are distributed as two bell curves. By selecting the tuning parameter to the leftmost position, everything will be classified positive yielding 100 % sensitivity and 0 % specificity, while at the rightmost position it will be the complete opposite, 0 % sensitivity and 100 % specificity. Thus, sensitivity and specificity must be interpreted as a pair in order to say anything about the overall performance of the classifier.

Receiver operating characteristic and area under the curve

A common way of investigating the performance of a binary classifier outputting a score rather than just the predicted class label is to generate an *receiver operating characteristic* (ROC) curve. In a binary classifier, the predicted class is acquired from the classifier score by its sign and the ROC curve is acquired by ranking the classifier output and the true class labels in ascending order with respect to the classifier score. Then a tuning parameter is introduced that bias the classifier outcome (changing the decision boundary

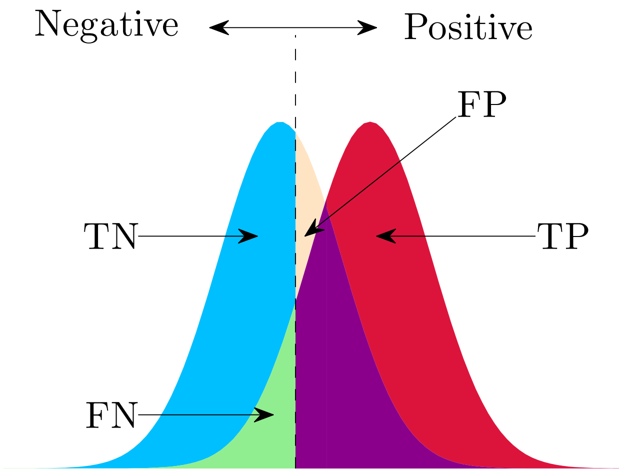


Figure 2.4: Two classes distributed as bell curves with the decision boundary being the dashed line. Everything left of this line is classified negative, while everything to the right of this line is classified positive.

in Figure 2.4) to predict everything into one class, yielding 100 % specificity and 0 % sensitivity. The tuning parameter is then incremented while at every step calculating sensitivity and specificity. This process is iterated until the classifier output is biased to yield 0 % specificity and 100 % sensitivity. Each threshold generates a measure of sensitivity and specificity which defines a single point on the ROC curve. The ROC curve is visualised by plotting 1-specificiy versus sensitivity[1] for all values of the tuning parameter and is a common evaluation tool in clinical medicine [29].

While the ROC curve itself is a two-dimensional representation of the classifier performance, it is sometimes convenient to use a scalar measure. The area under the ROC curve, *AUC*, is common [25] for this. A sample ROC curve and corresponding AUC is visualised in Figure 2.5. The statistical measure AUC is shown to be equivalent to the Wilcoxon rank-sum test [25], [30], [31], and it is therefore not necessary to deduce the whole ROC curve in order to obtain this measure. Instead, to calculate the AUC according to the Wilcoxon rank-sum test, the samples S are divided into positive (S_+) and negative (S_-) groups based on their actual class beloning. The AUC is then aquired by Equation (2.13).

$$\hat{A}(S, f_Z) = \frac{1}{|S_+| \cdot |S_-|} \sum_{x_i \in S_+} \sum_{x_j \in S_-} \mathrm{H}(f_Z(x_i) - f_Z(x_j)) \tag{2.13}$$

Equation (2.13) is a definition of the Wilcoxon Rank-Sum test where f_Z represents the output score from the classifier for a certain measurement x, and H is the Heaviside function defined by Equation (2.14).

$$\mathrm{H}\left(x\right) = \begin{cases} 1 & , \quad x > 0 \\ {}^{1}\!/_{2} & , \quad x = 0 \\ 0 & , \quad x < 0 \end{cases} \tag{2.14}$$

[1] It is sometimes plotted as specificity versus sensitivity.

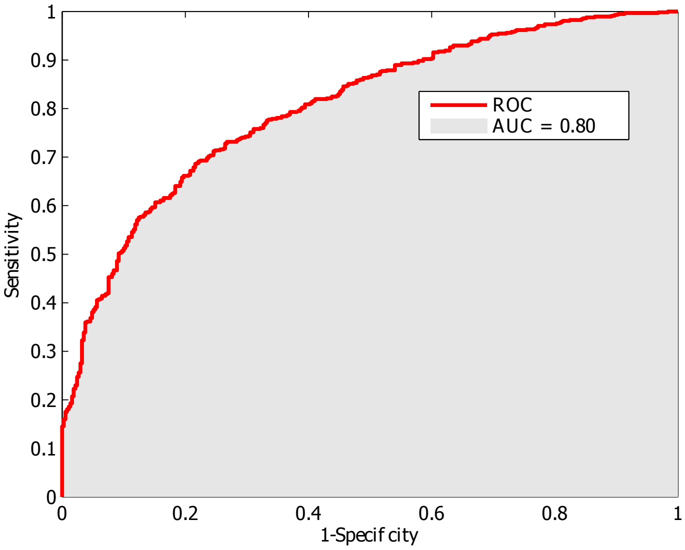


Figure 2.5: A sample ROC-curve (red) with the area under it shaded in gray.

Example

Lets take a simple example to get some hands on experience. Consider a case in which atlantic salmon is to be discriminated from atlantic cod. Hypothetically we assume that in the waters where we catch our fish there only exist cod and salmon, this will be a binary problem as we only have two different classes.

A skilled fisherman can distinguish between the two types of fish probably by the smell, while the less experienced amateur do this based on the look of the fish, see Figure 2.6 and Figure 2.7.

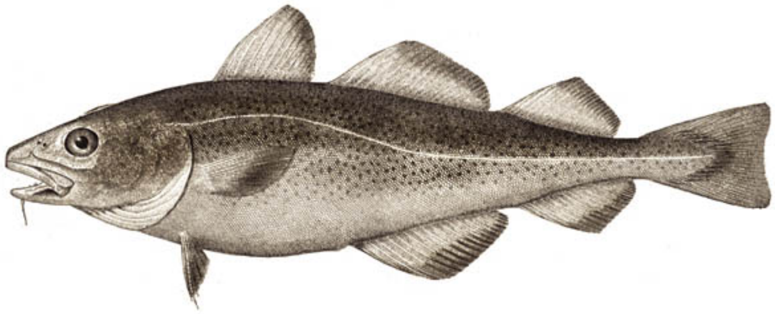


Figure 2.6: Cod. `http://upload.wikimedia.org/wikipedia/commons/a/a3/Atlantic_cod.jpg` (Visited on 2014-05-05)

However, as humans are somewhat lazy we want to automatise this process by the use of a camera and a computer. As we now we can discriminate the fish based on their looks, using a simple camera and a computer is not a bad choice. Now the task is to make the learning algorithm understand how to interpret the information in the image.

A machine is not aware of different types of fish and must therefore be told what is what. As humans we therefore have to identify certain characteristics, the *features*, for each fish species and tell the machine about them.

Consider the fish on a conveyor belt, the image of the camera must be preprocessed

Figure 2.7: Salmon. `http://upload.wikimedia.org/wikipedia/commons/archive/0/06/20091123150247!Salmo_salar_%28crop%29.jpg` (Visited on 2014-05-05)

in order to account for irregularities that might disturb the classification such as the orientation of the fish and unequal lighting conditions. After this preprocessing step, we have to extract features to be used by the following classification step in order to determine which type of fish it is.

We make an assumption that salmon in general are shorter than cod. This information suggests that the length can potentially be used as a feature. Using one feature as in this case will yield what is called a one-dimensional feature space. Suppose that the length of 30 fish of each species are determined, this results in something like what is shown in the bar plot in Figure 2.8.

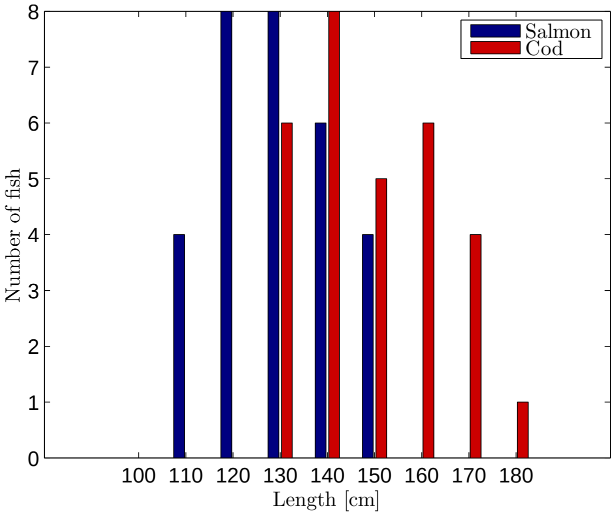


Figure 2.8: Distribution of length of cod and salmon from a sample of 30 salmon and 30 cod. The lengths are discretised with an error of ± 5 centimetres.

Now consider that we only want to catch cod and all salmon should be released back into the water. The threshold for the length could be set so that every fish longer than 150 centimetres are kept. Assuming that the the distribution in Figure 2.8 is correct, we will not catch any salmon. In order to acquire the confusion matrix, we first need to define what we will consider a positive and negative result. As we want to identify and catch cod, it is a good choice to select cod as being a positive result and salmon as a negative result. For the chosen threshold, eleven cod will be correctly identified, the other 19 cod will be considered as salmon, as will all the 30 salmon. No salmon will be wrongly classified as cod, and this yields the confusion matrix seen in Equation (2.15).

$$\begin{bmatrix} \text{TP} & \text{FP} \\ \text{FN} & \text{TN} \end{bmatrix} = \begin{bmatrix} 11 & 0 \\ 19 & 30 \end{bmatrix} \tag{2.15}$$

The accuracy of the classification algorithm using this threshold can be calculated according to Equation (2.11) and is found to be

$$\text{Accuracy} = \frac{11 + 30}{11 + 0 + 19 + 30} \approx 0.68 = 68\ \%$$

and is the empirical chance of a *correct* classification prediction. The sensitivity, by Equation (2.12a), is found to be

$$\text{Sensitivity} = \frac{11}{11 + 19} \approx 0.36\ldots = 37\ \%$$

which is equal to our classifiers' ability to identify cod. Equation (2.12b) gives the specificity, yielding

$$\text{Specificity} = \frac{30}{30 + 0} = 1 = 100\ \%$$

and is equivalent to the classifiers' ability to identify salmon.

What this tells us is that the classifier using the selected threshold can achieve that we catch no salmon. However some of the cod are also removed. As noted before, sensitivity can be traded for specificity and vice versa. In this case this is equal to say that we can tolerate that we catch some salmon after all, if we will identify and get even more cod. Setting the threshold to consider every fish being 120 centimetres or shorter being a salmon will identify all the cod, but will also classify some salmon as cod. For this threshold, the sensitivity will be increased to 100 % while the specificity will decrease to approximately 40 %. The accuracy on the other hand, is approximately the same, 70 %.

Recall that the ROC curve involves both sensitivity and specificity at different thresholds. By calculating the sensitivity and specificity while moving the threshold in the discrete steps seen in the x-axis in Figure 2.8, we acquire the ROC curve seen in Figure 2.9 and an AUC of 0.83. In this figure is also noted the thresholds of 150 centimetres and 120 centimetres.

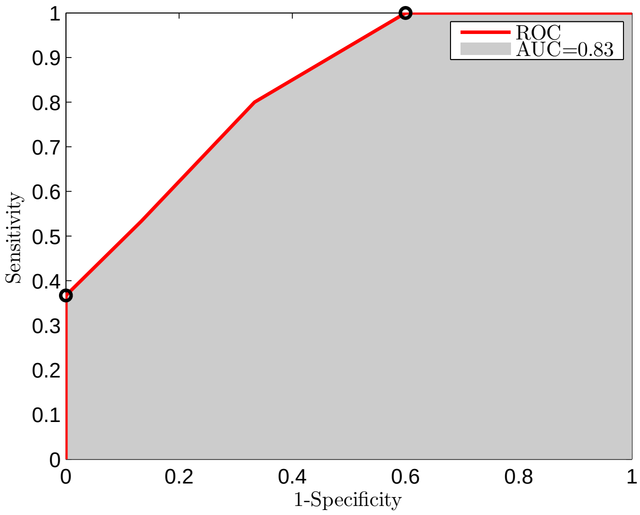


Figure 2.9: ROC curve for the separation of salmon and cod. Cod is considered the positive outcome. Uppermost and lowermost circles indicate the two different thresholds 120 centimetres and 150 centimetres respectively.

By inspection of the images in Figure 2.6 and Figure 2.7, one might suggest that the brightness of the scales differ and that this brightness might be suitable for classification.

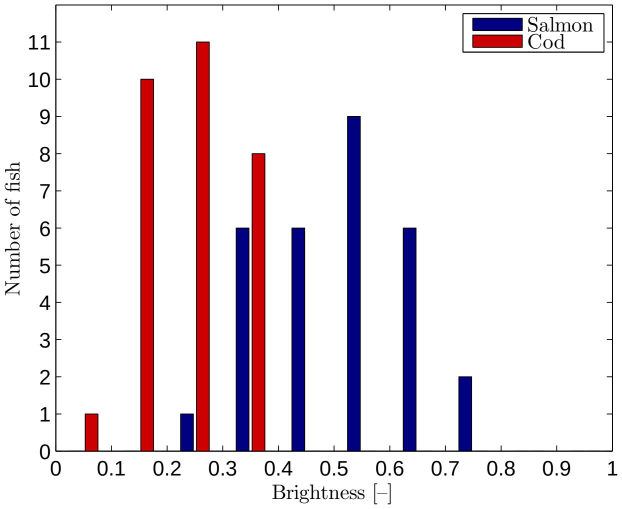


Figure 2.10: Fish brightness.

Only considering the brightness will yield the bar plot seen in Figure 2.10. As can be seen, using brightness is not suitable for discriminating the two fish species either. However, using both of the features might give a better outcome. Adding another feature will add one dimension to the feature space, in this example giving a two-dimensional problem. In fact, for this specific example, the boxplots in Figure 2.8 and Figure 2.10 are acquired by discretising the datapoints in the two-dimensional feature space seen in Figure 2.11 by the dashed lines. The two dimensional feature space is illustrated in Figure 2.11.

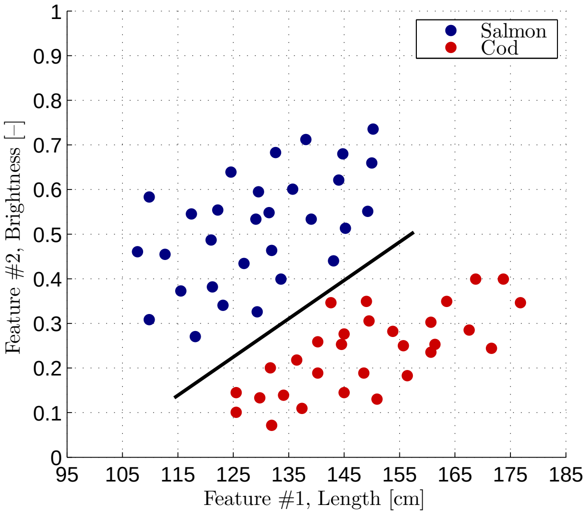


Figure 2.11: Fish brightness and length making up the two-dimensional feature space. The dashed lines are used for discretising the length to acquire the two box plots seen in Figure 2.8 and Figure 2.10.

Notably, the fish species are fully separable using two features as seen by the black line.

2.2.2 Linear subspace model

The microwave data acquired in the clinical dataset has high dimensions that is significantly larger than the sample size and is therefore regarded as a HDLSS problem [17], [18]. There are various ideas and concept on how to treat such problems and using linear subspace models (LSMs) is one [32].

In this thesis, all classifiers are based upon a linear subspace model. This model creates subspaces for each class from training data and then compare the size of the measurement within each of the subspaces. The feature extraction algorithm is based on a method described in the original paper by Yu and McKelvey (2013) [33] and Persson, Fhager, Trefná, *et al.* (2014) [34].

It is considered that a measurement $\boldsymbol{x}$ of S-parameters can be written as a linear combination of basis vectors

$$\boldsymbol{x} = \sum_{i=1}^{n_c} \alpha_i^c \boldsymbol{u}_i^c + e = \boldsymbol{U}_c \boldsymbol{\alpha}^c + e \tag{2.16}$$

where the matrix $\boldsymbol{U}_c$ defined as in Equation (2.17a) and the corresponding weights in $\boldsymbol{\alpha}_i^c$.

$$\boldsymbol{U}_c = [\boldsymbol{u}_{c,1} \cdots \boldsymbol{u}_{c,n_c}] \tag{2.17a}$$

$$\boldsymbol{U}_c^H \boldsymbol{U}_c = \boldsymbol{I} \tag{2.17b}$$

where $\boldsymbol{I}$ denotes the identity matrix and ^{H} the Hermitian conjugate[2]. The weight vector $\boldsymbol{\alpha}$ for a class c can then be written as

$$\boldsymbol{\alpha}^c = \mathbf{U}_c^H \boldsymbol{x} - \mathbf{U}_c^H e \tag{2.18}$$

where the error is considered small and thus can be neglected.

2.2.3 Inner product subspace classification

Classification using the linear subspace model is made by reducing the dimensions of the weight vector α^c for each class by using the inner product, hence its common name *Inner-product subspace classifier*, or *ISC*. A large size *within* the class for a measurement correspond to a small distance from the measurement *to* the class [33]. The size d within the class is calculated according to Equation (2.19).

$$d = \boldsymbol{\alpha}^H \boldsymbol{\alpha} \tag{2.19}$$

The class prediction is then acquired by the difference in the inner product, calculated by Equation (2.20).

$$\delta(\boldsymbol{x}) = \underbrace{\boldsymbol{x}^H \boldsymbol{U}_1}_{\boldsymbol{\alpha}_1^H} \underbrace{\boldsymbol{U}_1^H \boldsymbol{x}}_{\boldsymbol{\alpha}_1} - \underbrace{\boldsymbol{x}^H \boldsymbol{U}_2}_{\boldsymbol{\alpha}_2^H} \underbrace{\boldsymbol{U}_2^H \boldsymbol{x}}_{\boldsymbol{\alpha}_2} \tag{2.20}$$

The prediction function $\hat{f}$ for this case can then be written as in Equation (2.21).

$$\hat{f}(\boldsymbol{x}) = \begin{cases} +1 & \delta(\boldsymbol{x}) \;>\; 0 \\ -1 & \delta(\boldsymbol{x}) \;\leq\; 0 \end{cases} \tag{2.21}$$

[2] The Hermetian conjugate is defined as the complex conjugate of all elements in the transposed matrix.

2.2.4 Support vector machines

Support vector machines (SVM), also known as *Support Vector Networks*, are learning algorithms proposed in the mid 1990's and are derived from statistical learning theory which has its origin in the 1960's. The goal of using SVM for classification is to map input features to a high dimensional feature space and in that space construct a separating hyperplane between two classes [25], [35]. The idea of SVM is to use *support vectors*, a small amount of training data, to determine an optimal hyperplane with the maximal margin [36]. As we shall see, SVM predicts a measurement x into either positive or negative and is therefore a binary decision algorithm[3]. More detailed discussions on the topic of SVM can be found in Cortes and Vapnik (1995), Vapnik (1999), Steinwart and Christmann (2008), Chamasemani and Singh (2013). [35]–[38].

Optimal separating hyperplanes

Consider ℓ training data x with the class label c defined as

$$(x_1, c_1), \cdots, (x_\ell, c_\ell) \quad , \quad x \in \mathbb{R}^n \quad , \quad c \in \{+1, -1\}$$

to be linearly separable, that is if the inequalities in Equations (2.22) are satisfied [36].

$$\boldsymbol{w} \cdot \boldsymbol{x}_i + b \geq \quad 1 \quad , \quad c_i = 1 \tag{2.22a}$$

$$\boldsymbol{w} \cdot \boldsymbol{x}_i + b \leq \; -1 \quad , \quad c_i = -1 \tag{2.22b}$$

These equations can be condensed [38] into a more convenient form

$$c_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 \quad , \quad i = 1, \cdots, \ell. \tag{2.23}$$

The optimal separating hyperplane is unique and defined as

$$\boldsymbol{w}_0 \cdot \boldsymbol{x} + b_0 = 0. \tag{2.24}$$

If the vectors are separated perfectly and there is maximal distance between the closest vector and the hyperplane as illustrated in Figure 2.12, it is said that the vectors are separated by the *optimal hyperplane* [35].

The vectors that lies on the margin satisfies the condition $c_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) = 1$ and are the *support vectors* that is used to deduce the maximum margin classifier. The vector $\boldsymbol{w}_0$ that defines the optimal hyperplane can be written as a linear combination of the training vectors [35], [36]:

$$\boldsymbol{w}_0 = \sum_{i=1}^{\ell} c_i \alpha_i^0 \boldsymbol{x}_i = \sum_{\text{support vectors}} c_i \alpha_i^0 \boldsymbol{x}_i \quad , \quad \alpha_i^0 \geq 0 \tag{2.25}$$

where α_i^0 is a Lagrange multiplier from solving an optimisation problem [36] and only non-zero for the support vectors. Inserting the expression for $\boldsymbol{w}_0$ into the definition of the optimal separating hyperplane in Equation (2.24) will yield Equation (2.26) [35].

$$\sum_{i=1}^{\ell} \alpha_i^0 (\boldsymbol{x}, \boldsymbol{x}_i) + b_0 = 0 \tag{2.26}$$

[3] There exist multi-class SVM based on combining several binary SVMs [37]. However, in this thesis only binary classification is considered.

where the $(\boldsymbol{x}, \boldsymbol{x}_i)$ denotes the inner product. The decision function $\hat{f}(\boldsymbol{x})$ for the input $\boldsymbol{x}$ is then defined as in Equation (2.27).

$$\hat{f}(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{\ell} \alpha_i^0 (\boldsymbol{x}, \boldsymbol{x}_i) + b_0\right) \tag{2.27}$$

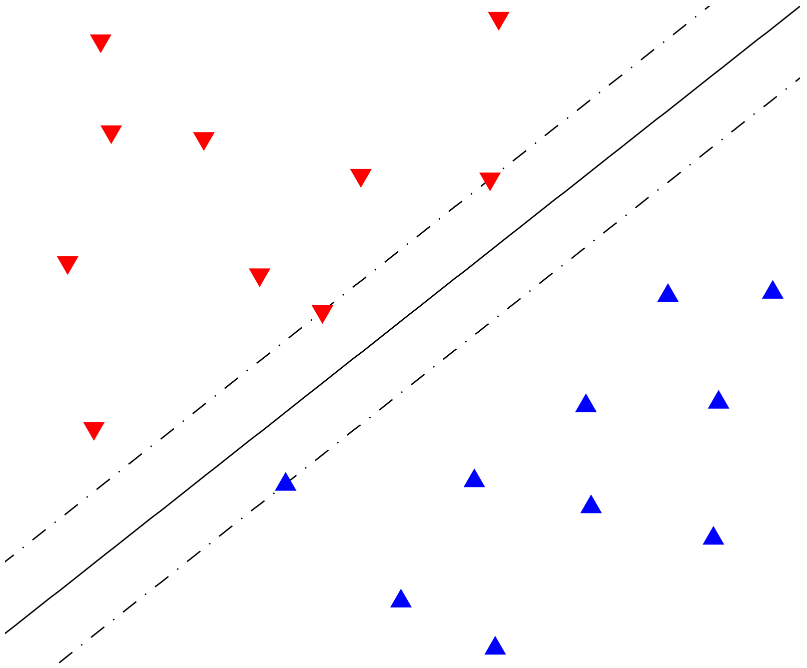


Figure 2.12: Linear SVM applied to two-dimensional perfectly separable training data. The decision boundary is the solid black line while the dotted lines are the margins. The vectors that lie on these margins are the *support vectors* as they do satisfy the condition $c_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) = 1$.

The non-separable case

Data might not be linearly separable due to, for example, outliers or noise in which case there is no separating hyperplane [38]. The standard approach to account for this case is to allow the algorithm to make mistakes for a cost [37]. Formally, the non-negative slack variables ξ_i are introduced which can be thought of as the closest distance from the measurement x_i to its corresponding margin, see Figure 2.13.

Together with the slack variable ξ_i the cost parameter C and the functional in Equation (2.28) is introduced.

$$\Phi(\boldsymbol{\xi}) = (\boldsymbol{w}, \boldsymbol{w}) + C \sum_{i=1}^{\ell} \xi_i \tag{2.28}$$

where the parameter $C > 0$ is an arbitrarily chosen constant. There might be an issue due to the loosened constraints, namely if the slack variables becomes too large [38]. To deal with this issue, they are incorporated in Equation (2.23) and Equation (2.28) shall thus be minimised with subject to constraints Equation (2.29) with respect to $\boldsymbol{w}$ and b [25], [35], [36], [38].

$$c_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 - \xi_i \quad , \quad \xi_i \geq 0 \quad , i = 1, \cdots, \ell. \tag{2.29}$$

The training data that violate the margins are excluded and support vectors are selected without them to create an optimal separating hyperplane using the remaining training data [36].

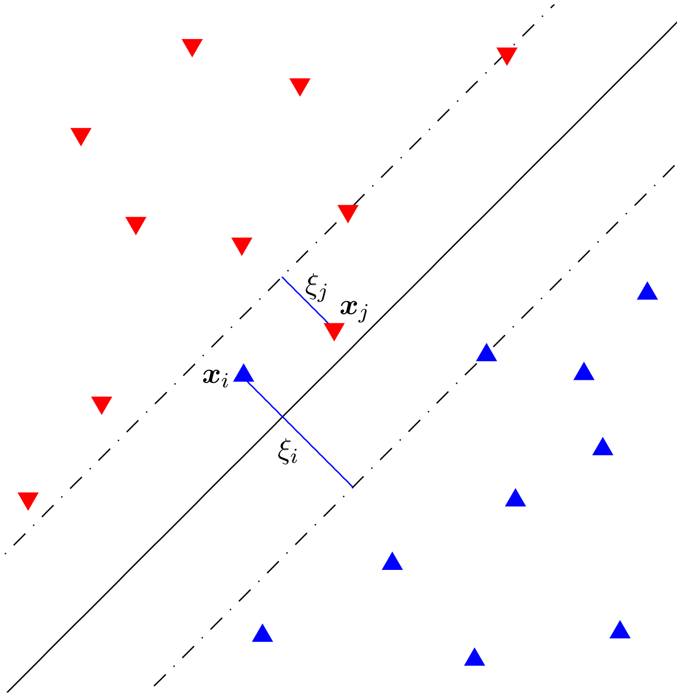


Figure 2.13: Linear SVM applied to two-dimensional linearly non-separable data. Note the slack variables for the two data points that is on the wrong side of their margin. The support vectors used to construct this optimal separating hyperplanes are located on the dotted margins.

The value of the cost parameter C is crucial as it relates to the smoothness of the decision boundary. A small value of C implies a large margin whereas a smaller margin is acquired by a large value of C [25]. A too high value for the cost parameter will therefore overfit the test data, which means that the generalisation of the classifier will be bad.

Non-linear data in input space and kernels

However, if the training data is poorly separable with a linear boundary, the generalisation of the model might be low even for the optimal hyperplane [37]. In this case, it is possible to map the input vectors to a feature space of very high dimensions with a nonlinear mapping [35] and then find the optimal separating hyperplane as before, now in the high-dimensional space. The idea is to construct a linear decision function in a feature space, that in fact will be a nonlinear decision function in the input space. This mapping K simply replaces the former inner product, making the decision function as in Equation (2.30) [35].

$$\hat{f}(\boldsymbol{x}) = \text{sign}\left(\sum_{\text{support vectors}} \alpha_i K(\boldsymbol{x}_i \cdot \boldsymbol{x}) + b_0\right) \tag{2.30}$$

The function or *kernel* K has the constraint that it must satisfy the condition given in [35], [36]

$$\int K(x,y)z(x)z(y)dxdy \geq 0. \tag{2.31}$$

for any functions $z(x)$ and $z(y)$ satisfying Equation (2.32).

$$\int z^2(x)dx \leq \infty \tag{2.32}$$

Learning machines that uses decision functions of the type seen in Equation (2.30) given any specific function K are called support vector machines. By selecting different functions for the inner products K, different learning machines are created. The linear kernel is the dot product, see Equation (2.33a) and the *radial basis function* kernel is defined as in Equation (2.33b) [25].

$$K(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u} \bullet \boldsymbol{v} = \boldsymbol{u}^{\mathrm{T}} \boldsymbol{v} \tag{2.33a}$$

$$K(\boldsymbol{u}, \boldsymbol{v}) = exp\left(-\gamma \|\boldsymbol{u} - \boldsymbol{v}\|^2\right) \tag{2.33b}$$

The parameter γ is called the width. The RBF-kernel allows non-linear decision boundaries as shown in Figure 2.14.

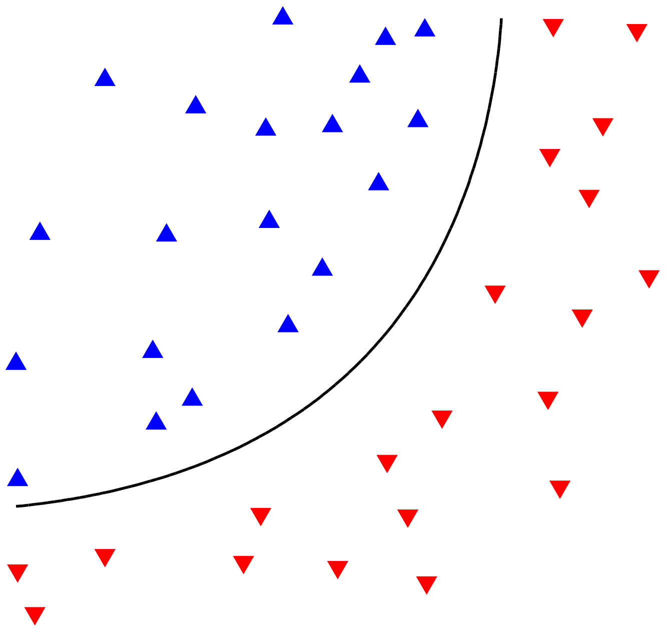


Figure 2.14: SVM with the RBF-kernel applied.

2.2.5 Large margin nearest neighbour

The large margin nearest neighbour, LMNN, classification algorithm builds upon the k-nearest neighbours (kNN) algorithm [19] and is one of the oldest pattern classification procedures [20]. The simple idea in kNN classification is to label a measurement depending on it's local neighbourhood which does not require any model to be fit [25].

Consider the ℓ measurements in the training data as

$$(x_1, c_1), \cdots, (x_\ell, c_\ell) \quad , \quad c \in \mathcal{C}$$

where each measurement x_i take values in some metric space and c_i is the class label from the set $\mathcal{C}$. Consider the set of points with known label (training data) as in Equation (2.34).

$$x'_\ell = \{x_1, x_2, \cdots, x_\ell\} \tag{2.34}$$

The closest neighbour to the unknown measurement x is the one that satisfies the condition in Equation (2.35) for any defined metric d [20].

$$\min d(x_i, x) = d(x'_\ell, x) \quad , \quad i = 1, 2, \cdots, \ell \tag{2.35}$$

By removing the nearest neighbour to the measurement x from the set of measurements with the known label the condition in Equation (2.35) will yield the second nearest neighbour and so forth. The predicted class label $\hat{c}$ of the unknown measurement x is determined by a majority vote of its k nearest neighbours. If the vote comes to a tie (in case an even number of neighbours are considered), this will be broken at random [25]. Figure 2.15 illustrates the kNN concept for $k = 3$ in two dimensions.

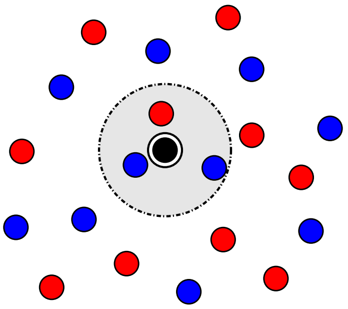


Figure 2.15: The concept of kNN classification in 2D for $k = 3$. The unclassified data (black in the center) is assigned a label due to a majority vote of it's three closest neighbours within the shaded area. In this example, there are two blue neighbours and one red neighbour, thus the test data will be assigned as belonging to the blue class.

As the nearest neighbours are determined by the condition in Equation (2.35), the defined metric d is of vital importance. According to Weinberger and Saul (2009) [19], the accuracy of any kNN algorithm is significantly dependent on the metric being used.

In this project a kNN-algorithm is used with an alternative metric to the Euclidean distance. The idea is to estimate statistical regularities in the data that can be estimated from the training dataset. This implementation is called *Large margin nearest neighbour*, abbreviated *LMNN* and uses the *Mahalanobis* metric. This procedure can be regarded as applying a linear transform $\mathbf{L}$ to the input space before applying the kNN-algorithm that uses Euclidean metric [19]. The conceptual idea is shown in Figure 2.16. A metric is any

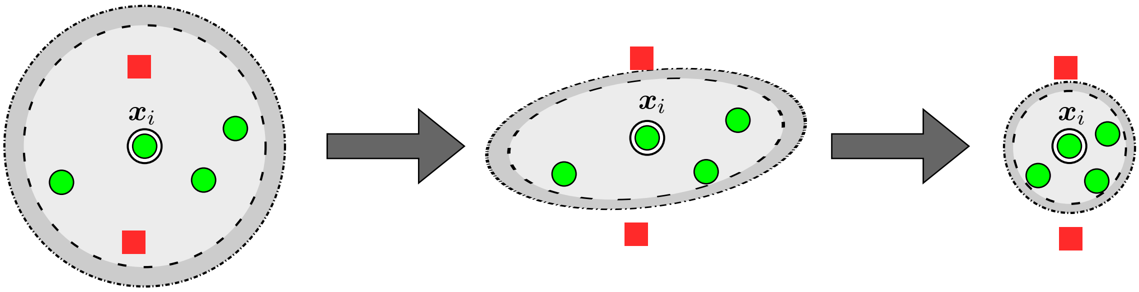


Figure 2.16: The green discs share the same label, while the red squares have different labels, and is therefore called *impostors*. By applying a linear transformation to the input space the impostors can be pushed outside of the margin. After the applied linear transformation the green discs appear closer.

mapping $d : X \times X \to \mathbb{R}$ that satisfies the following four conditions [39]:

1. $d(x, y) \geq 0 \quad , \quad x, y \in X$
   The distance is always positive.

2. $d(x, y) = 0 \iff x = y$
   A distance of zero is only true for the distance from the point to itself.

3. $d(x, y) = d(y, x) \quad , \quad x, y \in X$
   The distance is symmetric.

4. $d(x, y) \leq d(x, z) + d(z, y) \quad , \quad x, y, z \in X$
   The triangular inequality.

However, if the metric does satisfy all but the second condition, it is commonly referred to as a *pseudometric*.

The general form of the squared distance using Euclidean distances after performing a linear transformation by $\mathbf{L}$, is shown in Equation (2.36) [19].

$$d_{\mathbf{M}}(\boldsymbol{x}, \boldsymbol{y}) = \|\mathbf{L}(x_i - x_j)\|_2^2 = (\mathbf{L}x_i - \mathbf{L}x_j)^{\mathrm{T}}(\mathbf{L}x_i - \mathbf{L}x_j) = (x_i - x_j)^{\mathrm{T}}\mathbf{M}(x_i - x_j) \quad (2.36)$$

where $\mathbf{M} = \mathbf{L}^{\mathrm{T}}\mathbf{L} \succeq 0$ is a square matrix. This equation is a generalised form, from which the Euclidean metric can be derived by setting $\mathbf{M}$ to the identity matrix $\mathbf{I}$.

The constraint on $\mathbf{M}$ is that it is positive semidefinite which implies that it has no negative eigenvalues. This constraint will be satisfied if the matrix $\mathbf{M}$ is formed from any real-valued matrix $\mathbf{L}$. It shall also be noted that if $\mathbf{M}$ is a square matrix of full rank, the metric satisfies all the four conditions of being a metric, otherwise it will not satisfy the second condition and thus will be a pseudometric.

The goal will then be to estimate either the positive semidefinite matrix $\mathbf{M}$ or the linear transform $\mathbf{L}$ which uniquely defines the former matrix [19]. In the specific application for LMNN, the transformation being sought is the one achieves what is seen in Figure 2.16, namely to draw similarly labelled datapoints closer while pushing away impostors.

2.3 Cross-validation

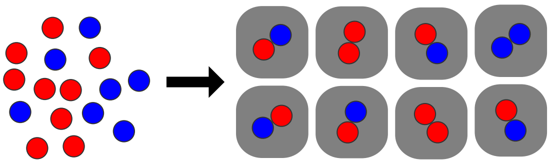
In order to estimate the performance of the classifier, the cross-validation procedure is used, which is a frequently used method to estimate prediction errors [25].

There are different cross-validation methods such as k-fold cross-validation (KFCV) and leave pair out cross-validation (LPOCV). Common for these algorithms are that they use one dataset and divides it into two datasets, called *training* data and *test* data. The training data is used to train the classification algorithm and the testing data is used to evaluate the performance of the trained classifier. In general, what differs the cross-validation methods is how the training and testing data is selected.

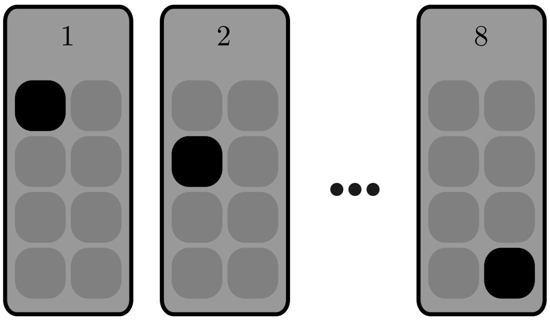
2.3.1 *k*-fold cross-validation

In k-fold cross-validation, the dataset is randomly divided into k equally sized groups or *folds* where one is used as validation data and $k - 1$ folds as training data [25]. After classification is performed, another of the k folds is used for testing while the other $k - 1$ data is used for training. This procedure is repeated until all folds has been selected as the testing set once. This procedure illustrated in Figure 2.17.

In this project, a special case of the k-fold cross-validation is used, called *leave one out cross-validation*, LOOCV. This means that the fold number k was set to be equal to the number of datapoints in the analysis, thus every testing set consisting of one datapoint that is ”left out”. For small-sample studies, it has been pointed out that the AUC received using LOOCV suffer from substantial negative bias [40].



(a) Division into $k = 8$ folds.



(b) Iterative procedure run $k = 8$ times.

Figure 2.17: k-fold cross-validation with $k = 8$. The red and blue samples in to the left in (a) are randomly ordered into eight groups which can be seen to the right. In (b) is seen the seven groups used to train the classifier (gray) and one group to be tested (black). The procedure is iterated until all groups have been excluded once, as seen in (b).

2.3.2 Leave pair out cross-validation

The *leave pair out cross-validation*, LPOCV, method is a validation method for binary classification problems in which two datapoints, one from each class, is left out. Furthermore, the training dataset is restricted to consist of equal amount of datapoints from each participating class. This method has proven successful to remove negative bias in AUC estimation for datasets similar in size to those used in this project [40], [41]. This concept is illustrated in Figure 2.18.

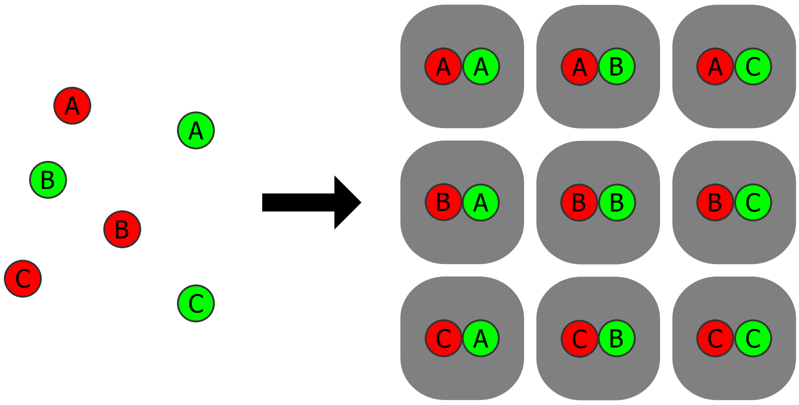


Figure 2.18: Leave pair out cross-validation. As there exist three samples from each class, this will generate nine pairs as every possible pair is tested. Like for the k-fold cross-validation, one of these groups is left out while the other eight are used for training of the classifier. The procedure continues until all groups has been left out once.

2.3.3 Acquiring AUC

How the AUC is retrieved can also vary among these cross-validation methods [41]. The AUC can be retrieved either using *pooling* or *averaging*. In pooling, the classifier scores from each test set is stacked and the AUC is calculated from the combined outputs. In averaging, a AUC is computed for each test set and is then averaged over all folds. For the case of using LOOCV, pooling is the only alternative. Additionally, the pooling strategy requires that the outputted scores are comparable and can be globally ordered.

2.3.4 Bootstrap and pseudo Monte Carlo procedure

There is an option to perform classification on the clinical data using a *pseudo Monte Carlo*, procedure and will henceforth be denoted *MC*. This means that a larger dataset than there actually is can be simulated as there are several observations for each measurement of every patient. By doing the classification iteratively while at every iteration randomly select which of the observations to be included, the number of combinations are increased. The total number of combinations is the number of observations of each measurement raised to the number of patients. This concept is illustrated in Figure 2.19 for a set of four measurements, each repeated three times. The measurements that are not selected for processing are discarded during that evaluation.

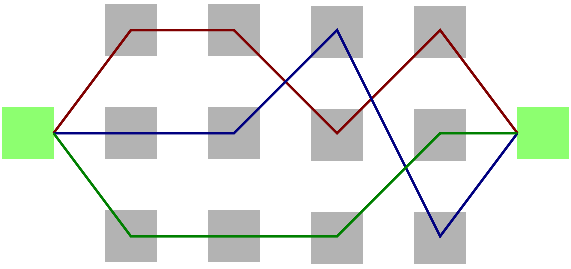


Figure 2.19: Pseudo Monte Carlo procedure. The number of unique combinations using the pseudo Monte Carlo procedure can be thought of as how many unique ways there exist to connect the two green boxes when the limitation is to take one vertical step. In the figure, three ways are drawn and in total there exist $3^4 = 81$ unique ways.

In order to achieve a statistical measure on the standard error of the estimate, a *bootstrap* method is issued. This means that a sample of MC simulations are selected at random, generating what is called a *bootstrap sample* to calculate average AUC over as many MCs selected. This procedure is repeated several times, while at every time new samples are drawn from the total numbers of MC simulations. Consider the set of all MC simulations are denoted by $\boldsymbol{Z} = \{z_1, z_2, \ldots, z_N\}$ where z_i denotes the i:th MC simulation. A bootstrap sample $\mathbf{Z}_i^*$ is acquired by randomly draw $n < N$ datasets from $\boldsymbol{Z}$ and can be used to predict an estimate of any parameter $\hat{\theta}$ [25], see Figure 2.20.

Using these B bootstrap samples, the standard error of the estimate $\hat{\theta}$ can then be achieved using the sample standard deviation of the bootstrap samples $\hat{\theta}_i^*$, as seen in Equation (2.37) [42].

$$S_{\hat{\theta}} = \sqrt{\frac{1}{B-1} \sum \left(\hat{\theta}_i^* - \bar{\theta}^* \right)^2} \tag{2.37}$$

where the notation $\bar{\theta}^* = \sum \hat{\theta}_i^* / B$ is the sample mean.

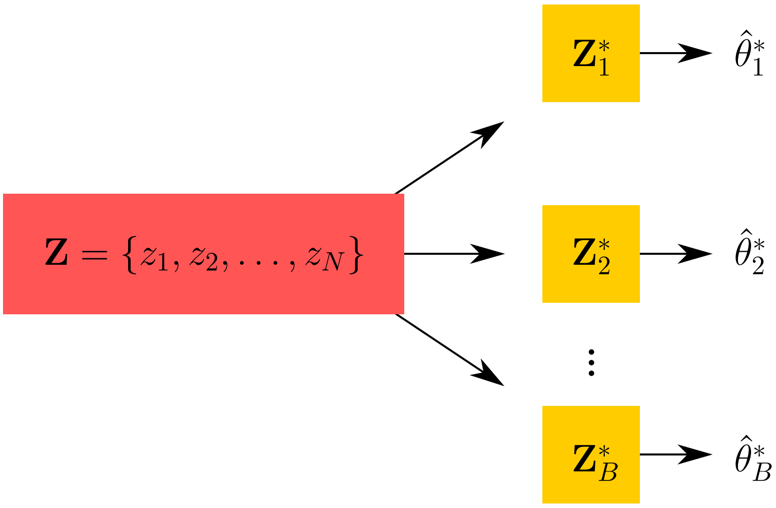


Figure 2.20: The bootstrap method. From the full dataset $\mathbf{Z}$, B bootstrap samples are drawn. From each of the bootstrap samples a statistical quantity $\hat{\theta}_i^*$ can be deduced.

Chapter 3

Materials and methods

In this chapter is presented what data has been used to perform the classification and the implementation of the methods briefly described in the theory section.

3.1 Data acquisition

The data used in this project was provided by Medfield Diagnostics. The patients in the study had suffered from either ICH or IS and microwave measurements were performed after undergoing CT-investigation and the type of stroke was confirmed by a physician from CT-images.

Three clinical datasets, denoted A, B-1 and B-2 were used for testing the algorithms. These datasets were acquired using different measurement setups. The frequency span considered in dataset A is 100-3000 MHz with a step size of 7.25 MHz. In datasets B-1 and B-2, the frequency step size is the same, but the frequency span considered is 100-1898 MHz. The dimensionality of both the datasets are high and depends on the number of frequency points considered and the number antenna combinations used. In total for dataset A there are 26466 dimensions, whereas for the datasets B-1 and B-2 there are 8964 dimensions.

Due to the fact that the data in the datasets is acquired with different measurement equipments, they are treated separately. The distribution of clots and bleedings is shown in Table 3.1 for dataset A and in Table 3.2 for dataset B-1 and B-2. It shall be pointed out that in the B-1 dataset, there were three patients having stroke mimics and one suffering from a subdural haematoma. Stroke mimics are diseases that mimic the symptoms for a stroke but is not [43]. A subdural haematoma is a bleeding but not in the same region as when having a stroke and can also be regarded as a stroke mimic.

Table 3.1: Distribution of patients in dataset A.

ICH	10 (40%)
IS	15 (60%)
Σ	25 (100%)

3.1.1 Measurement equipment

Schematic figure on how measurements are performed is seen in Figure 3.1. Antennas are placed around the head and one antenna starts sending a microwave which is received at

Table 3.2: Distribution of patients in dataset B-1 and B-2. In the B-1 dataset, there are three patients considered ICH but are stroke mimics and one patient labelled ICH suffered a subdural haematoma.

	B-1	B-2	Σ
ICH	25 (51%)	12 (27%)	37 (40%)
IS	24 (49%)	32 (73%)	56 (60%)
Σ	49 (53%)	44 (47%)	93 (100%)

the other antennas. When this is done, another antenna is sending while all others are receiving and this is repeated until all antennas has transmitted once.

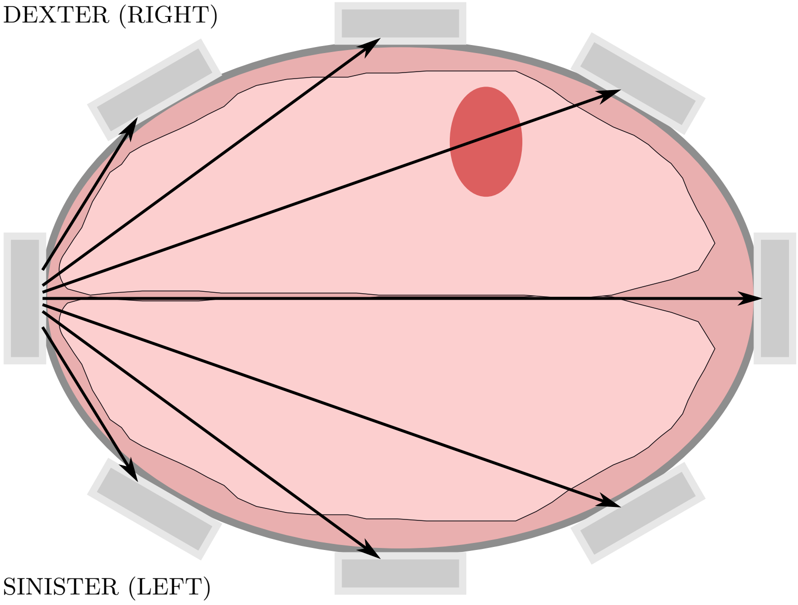


Figure 3.1: Transversal illustration of the head. Each antenna sends a signal that is received at the other antennas. This procedure is repeated in turn until all antennas has acted transmitter once. In this figure, a bleeding is located in the right hemisphere (red blob).

All of the antennas are connected to a vector network analyser that controls the measurement and measures the S-parameters. The antennas used emits most energy in the frequency span from around 1 GHz as this is where the magnitude drops below −3 dB, as seen in the reflection curve in Figure 3.2. As the reflection curve reaches −3 dB, this can be interpreted that 50 % of the power is emitted from the antenna. This frequency interval is therefore of interest as the hypothesis is that diagnostic information is located here.

The LOOCV method was used to verify the output of the classifier. Due to the sizes of the datasets and the distribution of ICH and IS, see Table 3.1 and Table 3.2, general k-fold was not used.

Leave-pair-out cross-validation was not used in this project due to the computational resources needed to test every possible pair combination.

3.2 Preprocessing

Preprocessing is an important step and suits to make data input to the classifier to be similar and consistent. As such, irregularities depending on for example the individual

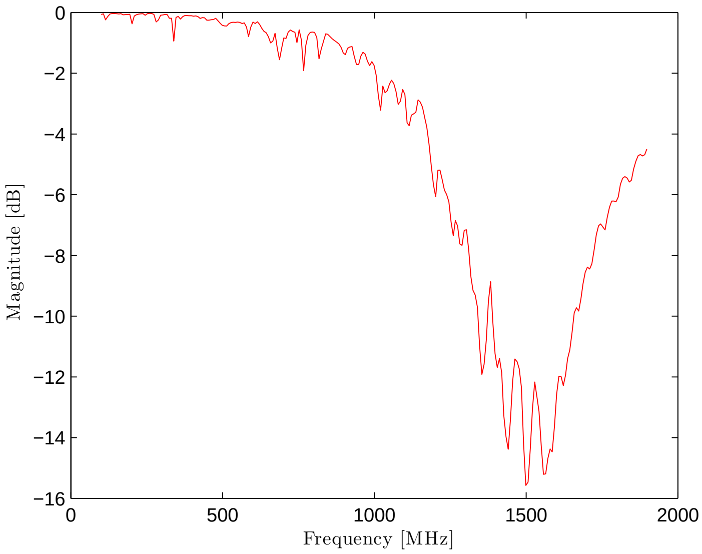


Figure 3.2: Reflection curve for an antenna in the frequency span 100–1898 MHz. At −3 dB 50 % of the energy is emitted from the antenna.

measurement setup should be handled in this step. During this project, preprocessing of the measurements was done in the way that in previous cases has been empirically tested by Medfield Diagnostics. This preprocessing was taking the logarithm for each antenna combination and normalise it. Although preprocessing is an important step for the classification outcome and some general assumptions can indicate what kind of preprocessing might be useful, there is no ideal way of selecting preprocessing procedure but to test empirically.

3.3 Classification

The classification implementation in this project is visualized in Figure 3.3 and henceforth the term *classifier* will be used for the process of discrimination, i.e. predicting the class given the features. Some classifiers output a classifier score that can be used to acquire the ROC-curve.

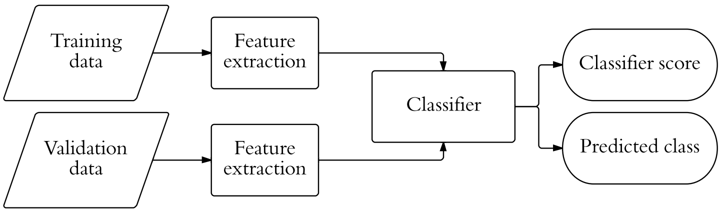


Figure 3.3: Classifier concept. Features are extracted from the data that is used by the classifier to predict the class belonging of the validation data. Some classifiers also output a score that can be used to calculate the ROC-curve.

Throughout this project, measurements from all antenna combinations except the reflection S-parameters were used. However, the frequency points used was varied.

3.3.1 Inner-product subspace classification

The ISC algorithm has two parameters called r and n that can be changed to bias the outcome. The r parameter is used to remove variations in the data that is not dependent on what is to be classified, i.e. variations related to other phenomena than IS and ICH. The n parameter is described as a noise reduction parameter. More detail on how these parameters work is found in [33], [34].

3.3.2 Support vector machines

Classification using support vector machines was made using the free LIBSVM package[1] [44]. SVM was applied with two different kernels, the linear kernel (henceforth denoted LSVM) and the RBF kernel (henceforth denoted RBFSVM).

For the LSVM, the internal parameter optimisation to acquire the cost parameter C was done by letting the SVM algorithm perform internal k-fold cross-validation for different values of C. The value yielding the best cross-validation accuracy is then selected. This internal cross-validation procedure utilised $k = 5$, that is, the data was divided into five groups and was performed in the training step of the algorithm. The parameters values checked are the ones listed in Equation (3.1).

$$C = \{0.5, \quad 1, \quad 2, \quad 4, \quad 8\} \tag{3.1}$$

When the RBF kernel was used, there parameter γ is added. Parameter optimisation are done in a similar way as for the linear SVM by using a grid search and compare cross-validation accuracy for each combination of C and γ. This procedure is one proposed way of finding the optimal parameters C and γ according to Hsu, Chang, and Lin (2010) [45]. The parameter values checked are the ones listed in Equation (3.2).

$$\gamma = \{0.0625, \quad 0.125, \quad 0.25, \quad 0.5, \quad 1, \quad 2\} \tag{3.2}$$

It shall be noted that good parameter values are highly dependent on the data and no theoretical consensus exist on how to select these parameters for a given dataset. It is noted by Steinwart and Christmann (2008) [38] that the cross-validation procedure might be associated with some disadvantages, such as overfitting. The reason for this is that there exist a relation between the training and testing data as they are drawn from the same set.

3.3.3 Large margin nearest neighbour

The algorithm to perform large-margin nearest neighbour classification is described in the paper by Weinberger and Saul. The implementation is also made by Weinberger [19] and can be found online[2]. For this project the LMNN version 2.4 was used. The parameter k was selected to be equal to one, meaning that the closest neighbour determines the class label of the measurement.

3.4 Testing setup

Classification using LSVM, RBFSVM and LMNN was performed using different input to the algorithms shown in Table 3.3. The ISC uses this weight vector and the parameters r and n to acquire the sizes within the subspaces, d_1 and d_2.

[1] The LIBSVM package can be acquired from `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

[2] The LMNN implementation can be found on `http://www.cse.wustl.edu/~kilian/code/code.html`.

Table 3.3: Input to the SVM and LMNN. The variable d stands for the sizes within the subspaces, as of equation (2.19) and α is the weight vector as in Equation (2.18). Indices 1 and 2 denotes class 1 and 2 respectively.

Variables	d_1,d_2 d_1,d_2,d_1d_2,d_1^2,d_2^2 $\boldsymbol{\alpha}_1$,$\boldsymbol{\alpha}_2$

While the sizes d_1 and d_2 are scalars, it shall be noted that the dimension of the vectors $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ depend on the sizes of the subspaces $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ respectively, see Equation (2.18).

For each measurement $\boldsymbol{x}$, the variables were stacked to give one long vector characteristic for that very measurement. For the first case where the two sizes within the subspaces were used, the feature vector had a length of two and in the second case a length of five, see Table 3.3. For the third case where the weight vector was used, the setup was a bit different as the dimensions of $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ are dependent on the training data sizes and are in general complex. As the used classification algorithms assumes real numbers, each weight vector was divided into its real and imaginary part and stacked according to equation (3.3).

$$\text{Variables} = \left[\text{Re}(\boldsymbol{\alpha}_1)^{\text{T}}, \text{Im}(\boldsymbol{\alpha}_1)^{\text{T}}, \text{Re}(\boldsymbol{\alpha}_2)^{\text{T}}, \text{Im}(\boldsymbol{\alpha}_2)^{\text{T}}\right] \tag{3.3}$$

It shall be noted that when considering the sizes within the subspaces as output d from the ISC, the internal tuning parameters from the ISC are adding to the overall complexity, greatly increasing computation time. This methodology can be seen as performing classification with the ISC for feature extraction. Table 3.4 together with the amount of parameters needed for each setup.

Table 3.4: Table of setups to test. For the classification setups starting with ISC followed by another classifier, the sizes within the subspaces were acquired from the ISC algorithm was given as input to the classifier.

Procedure	Input variables	Parameters
ISC	–	r, n, f_{min}, f_{max}
LSVM	$\boldsymbol{\alpha}_1$,$\boldsymbol{\alpha}_2$	C, f_{min}, f_{max}
RBFSVM	$\boldsymbol{\alpha}_1$,$\boldsymbol{\alpha}_2$	$C, \gamma, f_{min}, f_{max}$
LMNN	$\boldsymbol{\alpha}_1$,$\boldsymbol{\alpha}_2$	k, f_{min}, f_{max}
ISC - LSVM	d_1,d_2	$r, n, C, f_{min}, f_{max}$
ISC - RBFSVM	d_1,d_2	$r, n, C, \gamma, f_{min}, f_{max}$
ISC - LMNN	d_1,d_2	$r, n, k, f_{min}, f_{max}$
ISC - LSVM	d_1,d_2,d_1d_2,d_1^2,d_2^2	$r, n, C, f_{min}, f_{max}$
ISC - RBFSVM	d_1,d_2,d_1d_2,d_1^2,d_2^2	$r, n, C, \gamma, f_{min}, f_{max}$
ISC - LMNN	d_1,d_2,d_1d_2,d_1^2,d_2^2	$r, n, k, f_{min}, f_{max}$

As can be seen in Table 3.4, using the ISC to acquire the sizes within the subspaces adds two parameters compared to use the weight vector α. Therefore, in order to get comparable results, the initial ISC was used to select parameters r, n, f_{min} and f_{max}. This will address one of the aims, to compare whether a different classification algorithm can outperform the ISC.

A frequency resolution of 72.5 MHz was used for both clinical studies.

3.4.1 Control evaluations

In order to verify the results, control evaluations were performed in which the class labels of the data is randomly assigned. For the ISC and LMNN algorithms the control evaluations used an equal class distribution, whereas for the SVM algorithms the class distribution was not explicitly defined. This was due to that the SVM algorithms behaves as a majority class classifier if the data cannot be separated.

The hypothesis is that a classifier trained with data that does not contain any useful information will therefore not be able to classify the data correctly.

3.5 Code implementation

New functions and scripts were developed with Medfield Diagnostics' current code as reference. The reason for this was to add features and tidy up the former program to make it more flexible as well as decrease the computation time. The programs can be divided into the *classification procedure* itself, and *analysis software*.

3.5.1 Classification procedure

The classification procedure was built keeping the essentials such as the actual classification and adding several features that might be of interest for the analysis. Continuously the new version was checked with the old implementation for reference. The key concept while redesigning the classification software was to build modules that can easily be swapped to add new functionality and to be able to do reproductions of earlier classifications.

Computation time was decreased by using MATLAB's parallel computing toolbox which unlocks the full potential of modern multi-core processors as well as suitable for data clusters.

The implementation in this project first imported the data after which certain preprocessing was issued. After this step, data there was an option to select patients to include based on meta data such as measurement equipment or hair length. Classification was performed using the selected patients and the output from the classification was saved and could later be visualised by the analysis software. The code implementation in MATLAB follows the diagram shown in Figure 3.4.

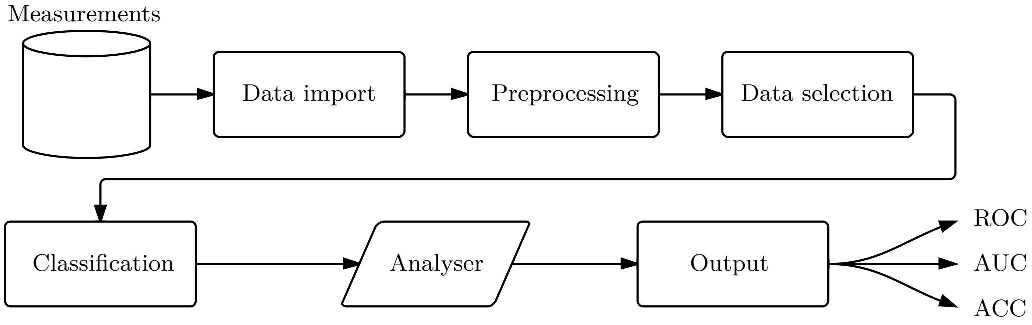
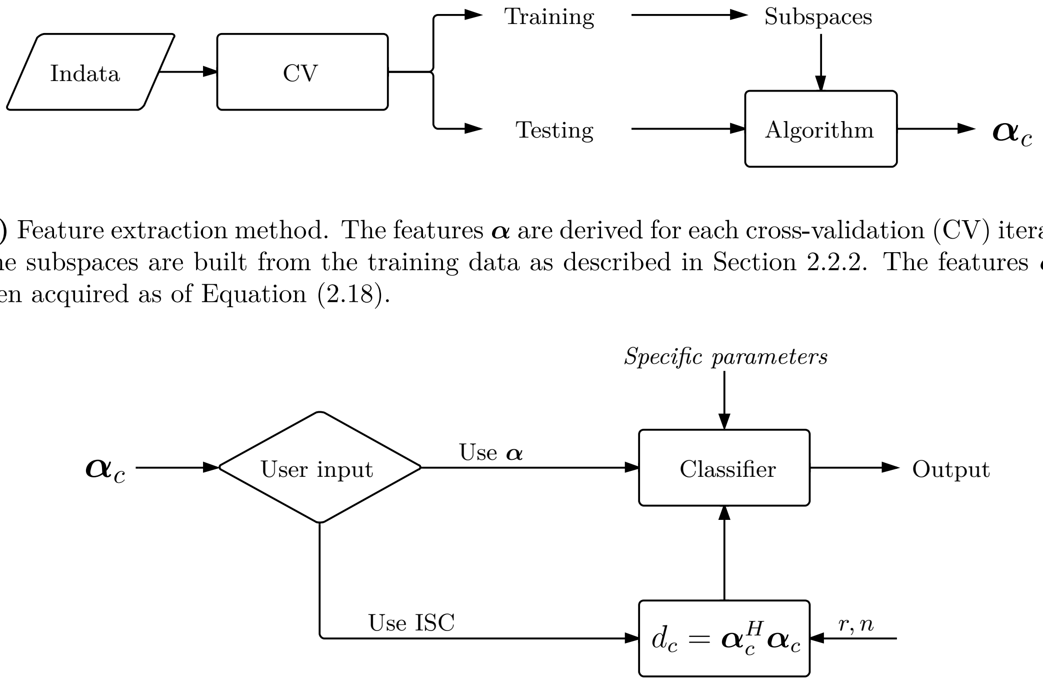


Figure 3.4: Overview of the program dataflow. Data is imported from a measurement database, is preprocessed and after that a selection based on metadata such as measurement equipment and hair length is performed. The selected data is sent to the classification algorithm that performs all numerical operations and this data output is analysed in a separate program that gives an interpretable result such as ROC, AUC or accuracy (ACC).

The generation of subspaces are done for every CV iteration as the training data is changed. From these subspaces the weight vectors $\boldsymbol{\alpha}$ are acquired.

The weight vector is then used as input to the classifier in which it is beforehand selected whether the weight vector is used as input directly to the classification algorithm or if the ISC should be used. These data flow is illustrated in Figure 3.5.



(a) Feature extraction method. The features $\boldsymbol{\alpha}$ are derived for each cross-validation (CV) iteration. The subspaces are built from the training data as described in Section 2.2.2. The features $\boldsymbol{\alpha}$ are then acquired as of Equation (2.18).

(b) Classification procedure. The features $\boldsymbol{\alpha}$ can be treated differently depending on the settings of the specific evaluation. Either the $\boldsymbol{\alpha}$ are used as input to the classifier directly, or the sizes within the subspaces are acquired as of Equation (2.19) and then fed to the classifier.

Figure 3.5: Detail of the classification algorithm. A detailed view of the classification step in Figure 3.4.

3.5.2 Analysis

In order to analyse the output generated by the classification procedure, an analysis software was developed. Apart from the earlier software, this introduced the possibility to acquire statistical measures such as ROC and accuracy in hindsight as well as during the computation.

Chapter 4

Results and discussion

4.1 Classification procedures

For the classifying setups that used ISC to achieve sizes within the subspaces as in Equation (2.19) prior to classification (see Figure 3.5b), the internal parameters of the ISC was fixed. Values are rounded to two decimals and a value is reported as 0.00 if it is found to be lower than 0.005. Standard deviations was acquired using a bootstrap procedure using 99 Monte Carlo simulations randomly drawn from the total of 100.

The control evaluations performed is done by randomly dividing the full dataset into the two classes representing ICH and IS respectively, i.e. the labels of the data is randomised.

Each of the following subsections are organised as follows. First presented will be the evaluations performed with fixed internal parameters of ISC that yielded good classification outcome for the ISC. These results are acquired by using the sizes within the subspaces as input. After this, the evaluations presented are when the weight vector α was used as input to the classifiers.

4.1.1 Using sizes within the subspaces as features

By issuing a parameter sweep over the internal parameters r and n of the ISC algorithm, a good value for these parameters could be picked out. Fixing these parameters and then select a frequency interval that corresponded to high AUC for the ISC algorithm, the initial hypothesis was that these settings would also be good for the other algorithms. This procedure used the sizes within the subspaces, d_1 and d_2, of the measurement x in the respective subspace as input to the SVM and LMNN algorithms. A test where interactions were used (d_1,d_2,$d_1 d_2$,d_1^2 and d_2^2) was also performed.

As the ISC algorithm was used to pick parameter values, it gives reason to believe that there might be a bias in favour of this algorithm.

4.1.1.1 Dataset A

For these evaluations, the frequency interval was fixed to be 898–1460 MHz. An amount of 100 MC iterations were used to acquire the results. The results from these evaluations are shown in Table 4.1. The standard deviation σ was approximated using bootstrap.

As can be seen in Table 4.1, the AUC using the LSVM or the RBFSVM algorithms generated an AUC of below 0.5. This normally suggests that the features given to the learning algorithm cannot be used in a way to discriminate between the two classes. It shall also be noted that the AUC of the control evaluations, in which class labels are

Table 4.1: AUC±σ using the size within the subspaces as input. The control column is the result when performing classification using data in which the labels are randomised. These results was acquired in the frequency span 898–1460 MHz.

Method	Features	AUC	Control evaluation
ISC	–	0.86 ± 0.01	0.47 ± 0.02
ISC-LSVM	d_1, d_2	0.41 ± 0.01	0.40 ± 0.01
ISC-RBFSVM	d_1, d_2	0.42 ± 0.01	0.42 ± 0.01
ISC-LSVM	d_1, d_2, d_1d_2, d_1^2, d_2^2	0.40 ± 0.01	0.39 ± 0.01
ISC-RBFSVM	d_1, d_2, d_1d_2, d_1^2, d_2^2	0.42 ± 0.01	0.42 ± 0.01

randomly assigned to the data, yields a desired result.

The fact that the AUC falls below 0.5 when using correct class labels (i.e. not a control evaluation) can be a consequence of that the classes are unequally distributed as seen in Table 3.1, with 40 % ICH and 60 % IS. Balancing the dataset to contain an equal distribution of both classes is one option to remove such bias. However, due to the relatively low sample size this was not considered as data is basically thrown away which might also affect the outcome.

The results from running the LMNN algorithm is shown in Table 4.2.

Table 4.2: Results from LMNN classification.

Features	Sensitivity	Specificity	Accuracy
d_1, d_2	0.35 ± 0.02	0.57 ± 0.03	0.48 ± 0.01
Control evaluation	0.47 ± 0.03	0.45 ± 0.03	0.46 ± 0.01
d_1, d_2, d_1d_2, d_1^2, d_2^2	0.37 ± 0.03	0.56 ± 0.03	0.48 ± 0.01
Control evaluation	0.46 ± 0.02	0.46 ± 0.03	0.46 ± 0.01

As can be seen from Table 4.2, the results do not suggest that there is any valuable information for the LMNN algorithm. In fact, the values of sensitivity and specificity is almost the same as the distribution of ICH and IS in the dataset (40 % and 60 % respectively). It can be seen that during the control evaluations, sensitivity and specificity approaches a value closer to 0.5. The reason for this is that in the control evaluations for the LMNN algorithm the distribution of the classes were close to equal.

4.1.1.2 Dataset B-1

For these evaluations, the fixed frequency interval was set to 825–1533 MHz. The outcome when performing classification using the sizes within the subspaces as input feature for the SVM algorithms is shown in Table 4.3.

Like for dataset A, the results in Table 4.3 show that using the sizes within the subspaces as feature input to the LSVM and RBFSVM did not yield a good result. Again the ISC algorithm outperforms the other algorithms although the result on dataset B-1 is worse than for dataset A. The control evaluations show the expected results as they approaches an AUC of 0.5, which is the expected outcome of a classification algorithm which select the outcome randomly.

For the LMNN algorithm, using the subspace sizes as input features acquired the results seen in Table 4.4.

As was observed for dataset A, the results using LMNN did not show any promising results when using the sizes within the subspaces as feature input. However, the sensitivity

Table 4.3: AUC using the sizes within the subspaces as input. The control column is the result when confusing the classifier by randomly assigning labels to the data. These results were acquired in the frequency span 825–1533 MHz.

Method	Features	AUC	Control evaluation
ISC	–	0.77 ± 0.01	0.49 ± 0.01
ISC-LSVM	d_1, d_2	0.51 ± 0.00	0.44 ± 0.01
ISC-RBFSVM	d_1, d_2	0.54 ± 0.00	0.45 ± 0.01
ISC-LSVM	d_1, d_2, d_1d_2, d_1^2, d_2^2	0.51 ± 0.00	0.43 ± 0.01
ISC-RBFSVM	d_1, d_2, d_1d_2, d_1^2, d_2^2	0.53 ± 0.00	0.44 ± 0.01

Table 4.4: Results for performing classification with the LMNN algorithm.

Features	Sensitivity	Specificity	Accuracy
d_1, d_2	0.47 ± 0.03	0.66 ± 0.03	0.57 ± 0.01
Control evaluation	0.53 ± 0.03	0.40 ± 0.03	0.47 ± 0.01
d_1, d_2, d_1d_2, d_1^2, d_2^2	0.48 ± 0.03	0.66 ± 0.03	0.57 ± 0.01
Control evaluation	0.53 ± 0.03	0.42 ± 0.03	0.47 ± 0.01

and specificity did not follow the trend of being close to the class distribution, which for dataset B-1 was 51 % ICH and 49 % IS which is an improvement. Again, the control evaluations yield expected results.

4.1.1.3 Dataset B-2

For the B-2 data, the frequency interval selected when using the sizes within the subspaces as feature input is 100–1823 MHz. The results from these classifications are shown in Table 4.5.

Table 4.5: AUC using the sizes within the subspaces as input. The control column is the result when confusing the classifier by randomly assigning labels to the data. These results was acquired in the frequency span 100–1823 MHz.

Method	Features	AUC	Control evaluation
ISC	–	0.78 ± 0.01	0.49 ± 0.02
ISC-LSVM	d_1, d_2	0.49 ± 0.01	0.40 ± 0.01
ISC-RBFSVM	d_1, d_2	0.51 ± 0.01	0.43 ± 0.01
ISC-LSVM	d_1, d_2, d_1d_2, d_1^2, d_2^2	0.48 ± 0.01	0.42 ± 0.01
ISC-RBFSVM	d_1, d_2, d_1d_2, d_1^2, d_2^2	0.51 ± 0.01	0.43 ± 0.01

The result follows the same trend that was observed in the previous datasets, A and B-1, that the sizes within the subspaces as feature input to LSVM and RBFSVM does not yield a good result. Again the ISC can be seen outperforming the other algorithms. The same trend can be seen for the LMNN algorithm which generated the results seen in Table 4.6.

As before for datasets A and B-1, the LMNN algorithm does not seem to be able to find discriminative information using the sizes within the subspaces. It shall be noted that the specificity seems to be very high while the sensitivity is very poor. This might be a fact of the skewed data in the B-2 dataset that consists of twelve ICH and 32 IS. This means that the subspaces for ICH and IS differ vastly in size where IS is almost three times as large. Such difference might introduce a significant bias to the largest class, which would

Table 4.6: Results for performing classification with the LMNN algorithm.

Features	Sensitivity	Specificity	Accuracy
d_1, d_2	0.21 ± 0.01	0.94 ± 0.01	0.65 ± 0.00
Control evaluation	0.46 ± 0.03	0.53 ± 0.03	0.50 ± 0.01
d_1, d_2, $d_1 d_2$, d_1^2, d_2^2	0.04 ± 0.01	0.96 ± 0.01	0.71 ± 0.00
Control evaluation	0.48 ± 0.03	0.45 ± 0.03	0.46 ± 0.01

partly explain the high specificity which means the certainty in identifying IS.

4.1.2 Using weight vectors as features

Using the weight vectors as features made the use of the ISC algorithm obsolete. The weight vectors acquired from the LSM are passed to the different classification algorithms instead of first reducing these vectors to the scalars d_1 and d_2. As ISC was not used, the parameters r and n were not used. The possible bias introduced when selecting these parameters in favour of the ISC algorithm is therefore not an issue.

Using the weight vectors as input features was evaluated for different frequency intervals. One selected by the means of where the ISC performed best and one based on a frequency sweep in which the frequency span that achieved good result for the specific classification algorithm was selected.

4.1.2.1 Dataset A

The frequency span and the corresponding AUC are shown in Table 4.7.

Table 4.7: Frequency spans investigated when using the weight vector α as input feature.

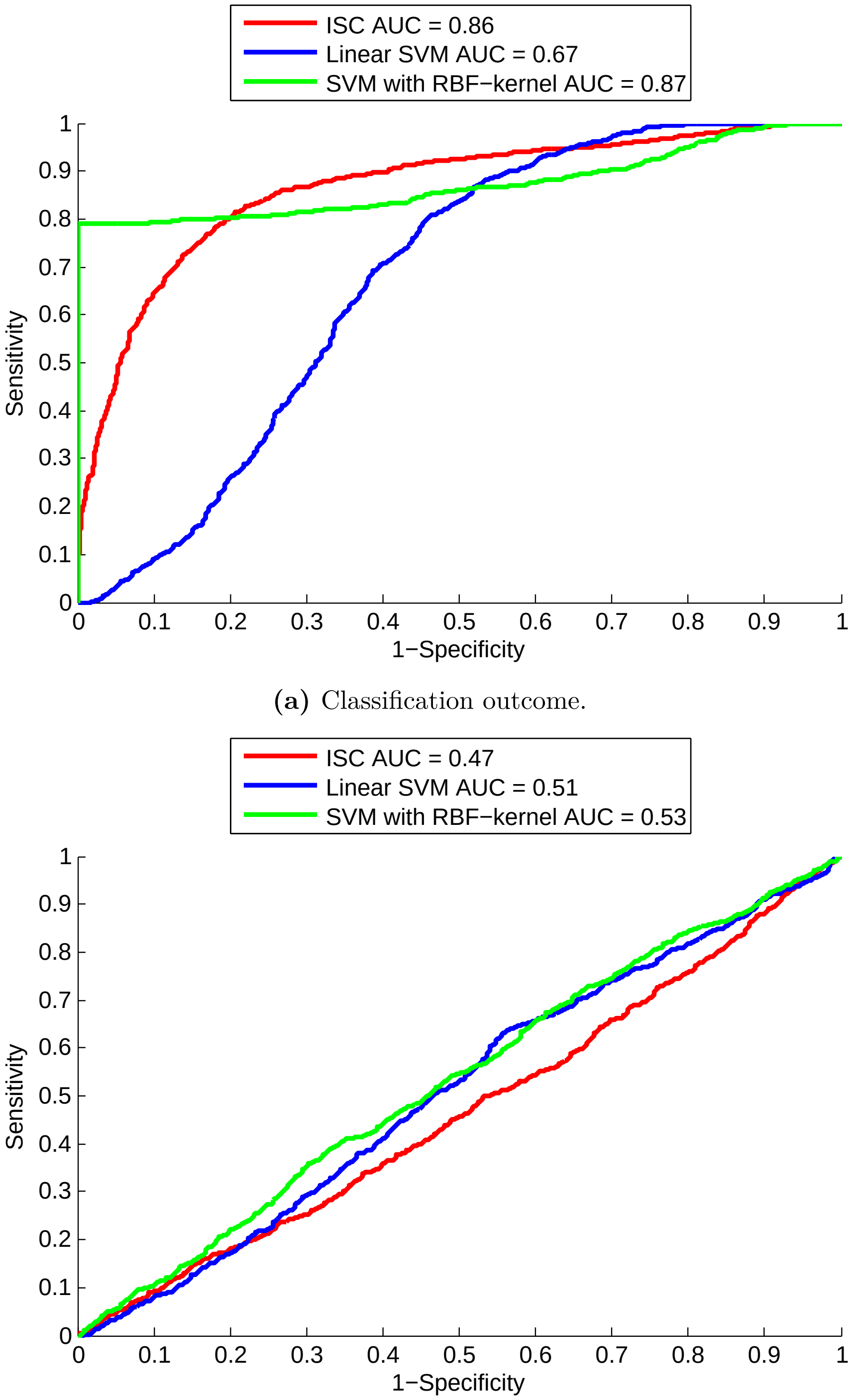
Method	Frequency span	AUC	Control
LSVM	897.5–1460	0.41 ± 0.01	0.51 ± 0.02
LSVM	1478–1750	0.67 ± 0.01	0.51 ± 0.02
RBFSVM	897.5–1460	0.07 ± 0.00	0.50 ± 0.01
RBFSVM	1333–1750	0.87 ± 0.01	0.53 ± 0.02

From these results can be concluded that the LSVM and RBFSVM performs significantly better than when using the sizes within the subspaces as inputs, seen in Table 4.1. In fact, the RBFSVM actually acquires an AUC that is slightly higher than that of ISC (0.87 compared to 0.86). The ROC-curves for the cases when ISC, LSVM and ISC operated in their best frequency spans (the frequency span in Table 4.7 yielding best AUC) are shown in Figure 4.1a and the corresponding control evaluation in Figure 4.1b.

From Figure 4.1a it can be seen that all algorithms perform better than a coin flip classifier that would have a ROC-curve along the diagonal. The RBFSVM can also be seen to have a point in which sensitivity is almost 0.8 while maintaining a specificity of 1. This means that in this point all IS patients are correctly classified as well as around 80 per cent of all ICH patients.

In Figure 4.1b, each label is randomly assigned, which yields expected results, as all algorithms have an AUC around 0.5, suggesting that they cannot discriminate the data.

The result of using the LMNN algorithm with the weight vector as input feature is shown in Table 4.8. Unfortunately, this method does not seem to yield satisfiable results. The highest sensitivity was found to be 0.36, meaning that in the best case 36 per cent of



(a) Classification outcome.

(b) Control evaluation outcome.

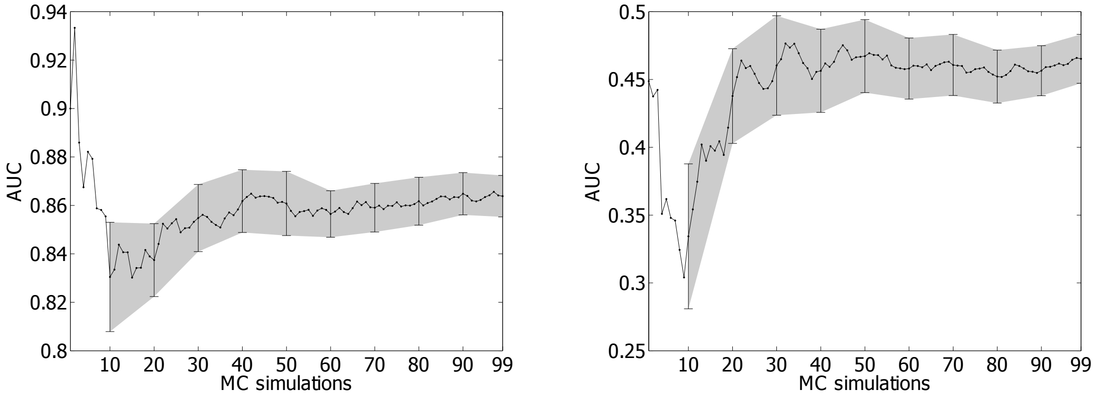
Figure 4.1: ROC curves for dataset A. The frequency span for ISC was 898–1460 MHz, for the LSVM algorithm 1478–1750 MHz and for the RBFSVM algorithm 1333-1750 MHz.

all ICH is identified. The control evaluations on the other hand yielded expected results as the sensitivity and specificity can be observed to approach 0.5.

Statistical analysis of the standard deviation with respect to increasing number of MC-simulations was performed issuing a bootstrap simulation at a given point using 50 iterations. For the ISC procedure, the obtained results are illustrated in Figure 4.2.

Table 4.8: Frequency spans investigated when using the weight vector α as input feature to the LMNN algorithm.

Frequency span	Sensitivity	Specificity	Accuracy
897.5–1460	0.17 ± 0.01	0.97 ± 0.01	0.65 ± 0.00
897.5–1460 (*Control evaluation*)	0.49 ± 0.03	0.47 ± 0.03	0.48 ± 0.01
752–1895	0.36 ± 0.02	0.89 ± 0.01	0.68 ± 0.01
752–1895 (*Control evaluation*)	0.51 ± 0.04	0.45 ± 0.04	0.48 ± 0.01



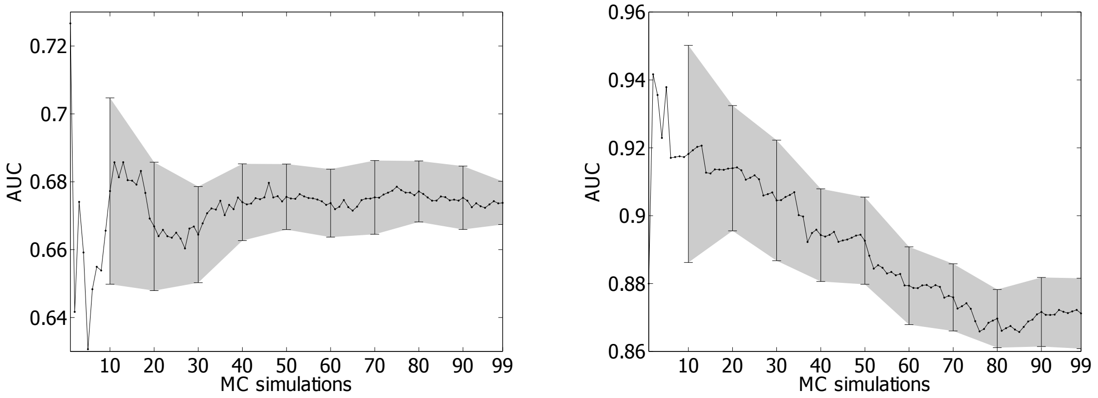
(a) Standard evaluation.

(b) Control evaluation with random labels.

Figure 4.2: Dataset A. AUC with estimated standard deviation from bootstrap samples acquired from 50 iterations for 10 different points. Standard deviation was calculated for 10, 20, 30, 40, 50, 60, 70, 80, 90 and 99 MC iterations. Note the different y-scales.

What can be seen from Figures 4.2 are that the AUC seems to converge and does not change much after 30-40 MC simulations. It can also be seen that the standard deviation is slightly decreasing when more MC simulations are added.

For the SVM algorithms, the results from estimating the standard deviation using the bootstrap method are shown in Figure 4.3. As for Figures 4.2a and 4.2b, it can be seen that the standard deviation decreases with the addition of more MC simulations. For the LSVM in Figure 4.3a it can also be seen that the AUC seems to converge after around



(a) LSVM.

(b) RBFSVM.

Figure 4.3: Dataset A using LSVM and RBFSVM. AUC with estimated standard deviation from bootstrap samples acquired from 50 iterations for 10 different points. Standard deviation was calculated for 10, 20, 30, 40, 50, 60, 70, 80, 90 and 99 MC iterations. Note the different y-scales.

40 MC simulations. However, for the RBFSVM seen in Figure 4.3b the AUC cannot be observed to quickly converge. It can though be seen that between 70-90 MC simluations the drop in AUC seems to diminish.

4.1.2.2 Dataset B-1

Using the weight vector as input was performed for different frequency intervals as seen in Table 4.9.

Table 4.9: Frequency spans investigated when using the weight vector α as input feature.

Method	Frequency span	AUC	Control evaluation
LSVM	825–1533	0.12 ± 0.00	0.51 ± 0.01
LSVM	100–1895	0.37 ± 0.01	0.49 ± 0.01
RBFSVM	825–1533	0.31 ± 0.01	0.49 ± 0.01
RBFSVM	680–1315	0.45 ± 0.01	0.49 ± 0.01

Unlike for dataset A, using the weight vector as input did not yield good results for data from dataset B-1. Recalling that the datasets was acquired using different measurement equipments, this might affect the results. The control evaluations, yielding an AUC of around 0.5, was expected as there is not thought to be any information suitable for discrimination.

The ROC-curves acquired for the B-1 dataset using SVM is shown in Figure 4.5 compared to ISC. As can be seen, the ISC clearly outperforms both LSVM and RBFSVM for the B-1 dataset.

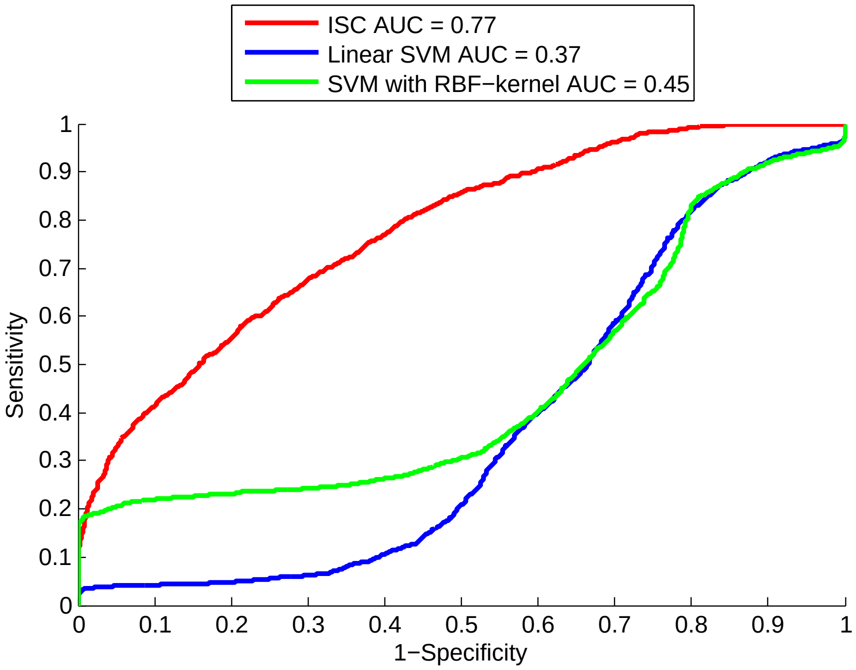


Figure 4.4: Standard evaluation.

Figure 4.5: ROC curves for the B-1 dataset. The frequency span for ISC was 825–1533 MHz, for the linear SVM algorithm 100–1895 MHZ and for the SVM algorithm with RBF-kernel 680–1315 MHz.

The results by the LMNN algorithm, operated in the frequency span 825–1533 MHz, is presented in Table 4.10.

Table 4.10: Frequency spans investigated when using the weight vector α as input feature to the LMNN algorithm.

Frequency span	Sensitivity	Specificity	Accuracy
825–1533	0.82 ± 0.02	0.17 ± 0.01	0.50 ± 0.01
825–1533 (*Control evaluation*)	0.51 ± 0.03	0.42 ± 0.03	0.47 ± 0.01

From Table 4.10 it can be seen that LMNN is significantly better in detecting ICH for dataset B-1 compared to dataset A in which the scenario was almost reversed with a low specificity and high sensitivity. The reason for this might be the different distribution of classes which in the B-1 dataset is almost equal (51 % ICH and 49 % IS) as well as the different measurement setup. However, the specificity is extremely low which results in a 50 % accuracy.

4.1.2.3 Dataset B-2

The resulting AUC acquired with the different algorithms using the weight vector α are shown in Table 4.11.

Table 4.11: Frequency spans investigated when using the weight vector α as input feature.

Method	Frequency span	AUC	Control evaluation
LSVM	100–1823	0.55 ± 0.00	0.48 ± 0.01
LSVM	608–1098	0.71 ± 0.00	0.48 ± 0.01
RBFSVM	100–1823	0.00 ± 0.00	0.46 ± 0.01
RBFSVM	825–1095	0.73 ± 0.02	0.48 ± 0.01

The result when applying LSVM and RBFSVM on the B-2 dataset yielded results clearly better than for the B-1 dataset and both algorithms were observed to yield results significantly better than the coin flip classifier. However, in one frequency span the RBFSVM was found to yield an AUC of zero, which is a strange result and might be because of training issues in the SVM algorithm which will be further discussed in Section 4.2.2.

The ROC curves using the SVM algorithms is shown in Figure 4.7 compared to the ISC algorithm. It can be observed that despite ISC acquiring higher AUC, there exists some points on the ROC-curve for RBFSVM that yield a better sensitivity-specificity tradeoff.

For the LMNN algorithm, the results generated is shown in Table 4.12.

Table 4.12: Frequency spans investigated when using the weight vector α as input feature to the LMNN algorithm.

Frequency span	Sensitivity	Specificity	Accuracy
100–1823	0.74 ± 0.02	0.49 ± 0.02	0.56 ± 0.01
100–1823 (*Control evaluation*)	0.44 ± 0.03	0.54 ± 0.03	0.49 ± 0.01

The sensitivity and specificity acquired by LMNN in this case could be identified as having the same performance as the LSVM algorithm in Figure 4.7, for some given tradeoff. Hence, the LMNN algorithm does not perform better than either the ISC or the RBFSVM on the B-2 dataset.

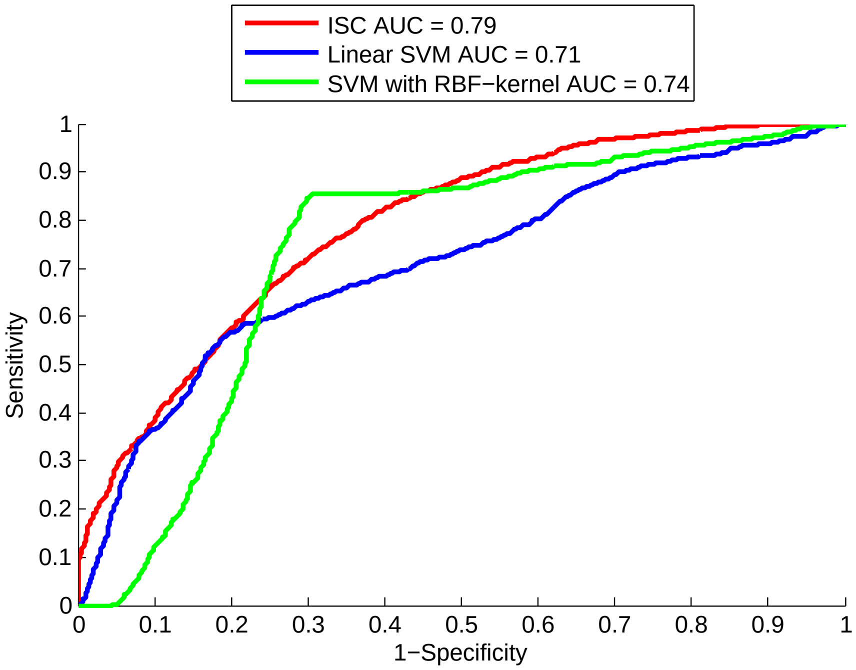


Figure 4.6: Standard evaluation.

Figure 4.7: ROC curves for dataset B-2. The frequency span for ISC was 100–1823 MHz, for the LSVM algorithm 608–1098 and for the RBFSVM algorithm 825–1098 MHz.

4.2 General discussion

This thesis project aimed to reimplement a version of the current classifier and investigate different classification algorithms, specifically SVM and LMNN, to see if these could improve the classification performance.

A reimplementation of the classifier in MATLAB has been made that has the ability to use custom classifiers. Three different classifiers has been tested, LSVM, RBFSVM and LMNN using three different setups of input features; sizes within the subspaces, sizes within the subspaces with interactions and a weight vector acquired from the LSM.

4.2.1 Selection of input features

4.2.1.1 Using sizes within the subspaces

Using sizes within the subspaces as features, as acquired with the ISC, did not show any promising results for any of the algorithms apart from the ISC algorithm itself. It shall though be considered that the internal parameters and the frequency span was selected based on parameter settings in which ISC performs well.

The AUC of the SVM algorithms is somewhat consistent throughout all three datasets studied in the sense that using the sizes within the subspaces from the ISC algorithm or using interactions of these does not perform any better than a ”coin flip” classifier. The same can be said about the LMNN algorithm which never achieved a higher sensitivity than 0.48 which means that slightly more than half of the ICH patients were missed in the best case. It shall also be stressed that there might be other parameter settings for the ISC that can improve the SVM and LMNN algorithms but that are not beneficial for the ISC algorithm. To find these, a sweep over the ISC parameters can be made in which the custom classifiers are tested for every parameter setting. Such procedure is computationally expensive and using a computer cluster would be beneficial.

The conclusions from these experiments are that the sizes within the subspaces, d_1 and d_2, outputted from the ISC using parameters for which it performs good, does not necessarily improve the performance of other classifiers.

4.2.1.2 Using weight vector

Using the weight vector as input to the algorithms showed improvements compared to using the sizes within the subspaces as features for all datasets. For dataset A, the RBFSVM yielded a slightly higher AUC than the ISC, see Figure 4.1a, but for a majority, neither SVM or LMNN did outperform the ISC algorithm in terms of AUC.

There are promising results in using the SVM with the RBF-kernel for classification purposes for some input features for some datasets. The LMNN algorithm did not produce results that outperform the ISC algorithm and the lack of tuning parameters to trade sensitivity for specificity is a drawback in this scenario.

4.2.2 Training procedures

The subspaces are acquired using the training data. When training the classifiers, the data used for training must of course be treated in the same way as the testing data. Therefore, Equation (2.18) is applied for every training data as well as testing data and in case of using the ISC this is follwed by acquiring the sizes within the subspaces as by Equation (2.19).

Given that the subspaces $\boldsymbol{U}_c$ are acquired using the training data, this operation might introduce a bias as the training data is in the subspaces and can be thought of as treated differently compared to testing data.

One might argue that the comparison now being made is unfair due to the bias introduced when comparing sizes within the subspaces of data actually in the subspace to sizes within the subspaces of data not used to create the subspaces. A possible way of addressing this problem might be to divide the training data into two parts, one from which subspaces are derived and one from which the features are extracted. An operation performed in this way will not introduce any bias in the extracted features. However, it must be taken into consideration that the loss of training data to estimate the subspaces might not be accounted for by the absence of potential bias in the training of the classifiers. It shall also be stressed that this potential bias might be resolved by having more training data.

The idea of dividing the training dataset in two, from which the subspaces are acquired from one part and the classifier trained with the other, would be interesting to apply to all of the tested classification algorithms.

It was found that in some cases all given training data was used as support vectors. With that in mind and recalling the definition of a support vector as a vector that lies on the margin, this means that all available training data is on the margin. However, it shall be noted that the training data was not always perfectly separable.

As noted by Xia, Lyu, Lok, *et al.* (2005) [46] and Cortes and Vapnik (1995) [36], the expected value of the probability of a wrong classification on a test example is related to the expected value of the number of support vectors and the number of training vectors. This holds only if the training vectors are completely separable by the SVM. This bound is described as in Equation (4.1).

$$E[\Pr(error)] \leq \frac{E[\#\ support\ vectors]}{\#\ training\ vectors} \tag{4.1}$$

Therefore, using all training vectors as support vectors and if the training data is perfectly classified, this will bound the expected probability of testing error to 100 % and might also imply bad generalisation capability [46].

The "training error" that might occur when using all available data as support vectors might therefore be a reason for acquiring AUC lower than 0.5 which suggest worse performance than a coin flip classifier.

4.2.3 Class distribution and data processing

It has been noted before that the dimensions of the raw measurements are very high and therefore the dimension reduction is achieved using the linear subspace model that reduces the measurements to the weight vector $\boldsymbol{\alpha}$. In the ISC this is the step prior to calculating the sizes within the subspaces. However, using either the sizes within the subspaces or the weight vector in SVM, these features are used to map the data again to higher dimensions to acquire an optimal separating hyperplane. It might seem odd that first reduce dimensions but then again map them to a higher-dimensional space. It might therefore be the case that information that might be of use to the SVM or the LMNN algorithms might be removed. This project however, did not intend to redesign the feature extraction algorithm using the linear subspace model, but it still is an interesting thought if dimension reduction could have been done in another way to benefit other decision algorithms. The results acquired when using the sizes within the subspaces as input to the algorithms compared to using the weight vector indicate that information of value to the subsequent classifiers are removed.

Further, the training datasets used contained patients that were of different age and gender. In dataset B-1, there were also patients included that had stroke mimics (diseases that look like stroke) and one with subdural haematoma. The time from stroke onset to measurement was also not considered when analysing the data. There are many metadata correlations that would be of interest when designing and improving the classification algorithms as some patients might introduce a bias.

One important task in classification is the preprocessing step. In this thesis, preprocessing has not been varied, but set statically to a preprocessing algorithm that has been used previously. This algorithm was selected based on the empirical results for the ISC algorithm. However, this does not necessarily mean that this is the ideal for other algorithms [47]. Investigating preprocessing methods must therefore be done once for every classifier used.

It is suggested that some feature extraction technique might be tested to select a subset of features from the weight vector that has the most impact on classification for both SVM and LMNN. By reducing the dimensions and identify features in the weight vector that have a positive contribution a better classification might be possible.

4.2.4 Cross-validation methods

Throughout this project, the LOOCV method has been used to evaluate the performance of the classifier. However, it has been noted that using LOOCV might suffer from substantial negative bias [40].

The reason for this method not being used in this project was due to the computational complexity. Consider a case in which m are the number of patients belonging to one class and n are the number of patients belonging to the other class. In LOO, the number of folds are equal to the number of patients included, i.e. $m + n$. Testing all possible pair combinations of all m patients of one class and n patients of the second class yield in total $n \cdot m$ folds.

Chapter 5

Conclusion

The reimplementation of the classification algorithm to create a faster and more versatile program being able to use different classification algorithms was successful.

Of the different classifiers studied, using the weight vector acquired from the LSM performed better in comparison to using the sizes within the subspaces acquired from the ISC. This conclusion holds for when the internal ISC parameters are selected based upon the outcome when using the ISC for classification and cannot be interpreted as a general case.

SVM used with an RBF kernel did perform well and acquired a higher AUC than the ISC algorithm for one of the datasets. For another dataset, SVM with the RBF kernel could be observed to yield a better ROC-curve than the ISC algorithm even though the AUC was lower. SVM with a linear kernel was never observed to outperform SVM with an RBF kernel or the ISC algorithm. The LMNN algorithm was not found to perform convincing results on any of the datasets.

The SVM with the RBF kernel as well as the ISC remains are found to be interesting techniques for future investigation and evaluation. Even though the LMNN algorithm did not perform good, the concept of using a different metric than the Euclidean remains interesting and should be further investigated.

Chapter 6

Future work

For SVM it is of interest to find a better way to acquire the ideal parameters C and γ. Cross-validation accuracy has a bias to the largest class and it would therefore be interesting to acquire the parameters to use the AUC measure. It would also be interesting to study the impact of different values for the parameters apart from those involved in this study. For unbalanced datasets, Eitrich and Lang (2006) [48] proposed optimisations on how to find optimal parameters for SVM for unbalanced datasets that might be of interest.

For the feature extraction, it would be interesting to test the case where the dataset is divided in two parts, one to estimate the subspaces and one to extract features that is used to train the algorithms, as previously discussed. Also using a feature selection algorithm to identify which of the features in the weight vector that might be of value for classification.

It would also be of interest to study the impact of including the reflection S-parameters in the data as opposed to only the transmission S-parameters.

Some datasets have reference measurements performed on a phantom in connection to the actual patient measurement. It would be interesting to investigate how these could be used to perform preprocessing of the data by e.g. normalising each measurement to the corresponding reference measurement. Exactly how this should be done is an open question.

Investigating the use of LPOCV instead of LOOCV to evaluate the classifier performance would be of interest to see if there is substantial negative bias.

As the MATLAB code produced as a part of this thesis is written very generally, this allows the investigation of other discrimination procedures than SVM and LMNN by simply creating a module. One method that in recent studies has shown efficient is the *support vector metric learning*, SVML, that combines the concepts of metric learning and support vector machines. This concept is described in Xu, Weinberger, and Chapelle (2013) [49].

The future work can be summarised as:

- Select internal parameters C and γ by using cross-validation AUC instead of cross-validation accuracy.
- Investigate other ranges for the parameter values C and γ.
- Split training dataset in two, one to estimate subspaces and one to train the custom classifiers.
- Study impact on different preprocessing methods and usage of reference measurements.

- Implement and test a plugin using SVML.
- Investigating impact of considering reflection coefficients.
- Test different cross-validation methods such as LPOCV

Bibliography

[1] A. D. Lopez, C. D. Mathers, M. Ezzati, *et al.*, "Global and regional burden of disease and risk factors, 2001: systematic analysis of population health data", *The Lancet*, vol. 367, no. 9524, pp. 1747–1757, May 2006. DOI: 10.1016/S0140-6736(06)68770-9.

[2] *The global burden of disease: 2004 update*, World Health Organization, 2004. [Online]. Available: http://www.who.int/healthinfo/global_burden_disease/GBD_report_2004update_full.pdf (visited on 11/25/2013).

[3] S. Semenov, "Microwave tomography: review of the progress towards clinical applications", *Philosophical Transactions of the Royal Society*, vol. 367, no. 1900, pp. 3021–3042, Aug. 2009. DOI: 10.1098/rsta.2009.0092.

[4] S. Y. Semenov and D. R. Corfield, "Microwave Tomography for Brain Imaging: Feasibility Assessment for Stroke Detection", *International Journal of Antennas and Propagation*, vol. 2008, pp. 1–8, Mar. 2008. DOI: 10.1155/2008/254830.

[5] M. W. Kurz, K. D. Kurz, and E. Farbu, "Acute ischemic stroke – from symptom recognition to thrombolysis", *Acta Neurologica Scandinavica*, vol. 127, pp. 57–64, Jan. 2013. DOI: 10.1111/ane.12051.

[6] G. J. Hankey, "The global and regional burden of stroke", *The Lancet*, vol. 1, no. 5, e239 –e240, Nov. 2013. DOI: 10.1016/S2214-109X(13)70095-0.

[7] K. Fassbender, C. Balucani, S. Walter, *et al.*, "Streamlining of prehospital stroke management: the golden hour", *The Lancet*, vol. 12, no. 6, pp. 585–596, Jun. 2013. DOI: 10.1016/S1474-4422(13)70100-5.

[8] K.-S. Hong, "Disability-Adjusted Life Years Analysis: Implications for Stroke Research", *Journal of Clinical Neurology*, vol. 7, no. 3, pp. 109–114, Sep. 2011. DOI: 10.3988/jcn.2011.7.3.109.

[9] J. Persson, J. Ferraz-Nunez, and I. Karlberg, "Economic burden of stroke in a large county in Sweden", *BMC Health Services Research*, vol. 12, no. 341, pp. 1–8, 2012. DOI: 10.1186/1472-6963-12-341.

[10] K. W. Muir, "Stroke", *Medicine*, vol. 37, no. 2, pp. 109–114, Feb. 2009. DOI: 10.1016/j.mpmed.2008.11.004.

[11] K. W. Muir, A. Buchan, R. von Kummer, *et al.*, "Imaging of acute stroke", *The Lancet*, vol. 5, no. 9, pp. 755–768, Sep. 2006. DOI: 10.1016/S1474-4422(06)70545-2.

[12] J. A. Chalela, C. S. Kidwell, L. M. Nentwich, *et al.*, "Magnetic resonance imaging and computed tomography in emergency assessment of patients with suspected acute stroke: a prospective comparison", *The Lancet*, vol. 369, no. 9558, pp. 293–298, Feb. 2007. DOI: 10.1016/S0140-6736(07)60151-2.

[13] I. Eshed, C. E. Althoff, B. Hamm, *et al.*, ”Claustrophobia and Premature Termination of Magnetic Resonance Imaging Examinations”, *Journal of Magnetic Resonance Imaging*, vol. 26, no. 2, pp. 401–404, Aug. 2007. DOI: `10.1002/jmri.21012`.

[14] S. Gabriel, R. W. Lau, and C. Gabriel, ”The dielectric properties of biological tissues: III. Parametric models for the dielectric spectrum of tissues”, *Physics in Medicine and Biology*, vol. 41, no. 11, pp. 2271–2293, Nov. 1996. DOI: `10.1088/0031-9155/41/11/003`.

[15] R. Pethig, ”Dielectric Properties of Biological Materials: Biophysical and Medical Applications”, *IEEE Transactions on Electrical Insulation*, vol. EI-19, no. 5, pp. 453–474, Oct. 1984. DOI: `10.1109/TEI.1984.298769`.

[16] D. M. Pozar, *Microwave Engineering*, 4th ed. Hoboken, NJ: John Wiley & Sons, Inc., 2012. [Online]. Available: `http://www.electron.frba.utn.edu.ar/~jcecconi/Bibliografia/Ocultos/Libros/Microwave_Engineering_David_M_Pozar_4ed_Wiley_2012.pdf`.

[17] S. J. Raudys and A. K. Jain, ”Small sample size effects in statistical pattern recognition: recommendations for practitioners”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 252–264, 1991. DOI: `10.1109/34.75512`.

[18] P. Hall, J. S. Marron, and A. Neeman, ”Geometric representation of high dimension, low sample size data”, *Journal of the Royal Statistical Society*, vol. 67, no. 3, pp. 427–444, Jun. 2005. DOI: `10.1111/j.1467-9868.2005.00510.x`.

[19] K. Weinberger and L. Saul, ”Distance metric learning for large margin nearest neighbor classification”, *The Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[20] T. Cover and P. Hart, ”Nearest neighbor pattern classification”, *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967. DOI: `10.1109/TIT.1967.1053964`.

[21] R. Pethig, ”Dielectric properties of body tissues”, *Clinical Physics and Physiological Measurement*, vol. 8, no. 4A, pp. 5–12, 1987. DOI: `10.1088/0143-0815/8/4A/002`.

[22] G. J. Tortora and B. Derrickson, *Essentials of anatomy and physiology*, 9th ed. John Wiley & Sons, Inc., 2013.

[23] S. Semenov, R. H. Svensson, and G. P. Tatsis, ”Microwave Spectroscopy of Myocardial Ischemia and Infarction. 1. Experimental Study”, *Annals of biomedical engineering*, vol. 28, no. 1, pp. 48–54, Jan. 2000. DOI: `10.1114/1.253`.

[24] S. Y. Semenov, R. H. Svenson, G. Simonova, *et al.*, ”Dielectric properties of canine acute and chronic myocardial infarction at a cell relaxation spectrum. I. Experiments”, in *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. ’Magnificent Milestones and Emerging Opportunities in Medical Engineering’*, vol. 1, 1997, pp. 202–205. DOI: `10.1109/IEMBS.1997.754504`.

[25] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed., ser. Springer Series in Statistics. Springer, 2009. [Online]. Available: `http://www-stat.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf` (visited on 12/03/2013).

[26] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley, 2001.

[27] T. Fawcett, ”An introduction to ROC analysis”, *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006. DOI: `10.1016/j.patrec.2005.10.010`.

[28] D. M. Powers, ”Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation”, *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, Feb. 2011.

[29] M. H. Zweig and G. Campbell, ”Receiver-Operating Characteristic (ROC) Plots: A Fundamental Evaluation Tool in Clinical Medicine”, *Clinical Chemistry*, vol. 39, no. 4, pp. 561–577, 1993.

[30] J. A. Hanley and B. J. McNeil, ”The meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve”, *Radiology*, vol. 143, no. 1, pp. 29–36, Apr. 1982. DOI: `10.1148/radiology.143.1.7063747`.

[31] D. J. Hand and R. J. Till, ”A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems”, *Machine Learning*, vol. 45, no. 2, pp. 171–186, Nov. 2001. DOI: `10.1023/A:1010920819831`.

[32] Y. Yu, ”Classification of High Dimensional Signals with Small Training Sample Size with Applications towards Microwave Based Detection Systems”, Lic. thesis, Chalmers University of Technology, Gothenburg, Sweden, Jun. 2013.

[33] Y. Yu and T. McKelvey, ”A unified subspace classification framework developed for diagnostic system using microwave signal”, in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, Sep. 2013.

[34] M. Persson, A. Fhager, H. D. Trefná, *et al.*, ”Microwave-based stroke diagnosis making global pre-hospital thrombolytic treatment possible”, *IEEE Transactions on Biomedical Engineering*, Jun. 2014. DOI: `10.1109/TBME.2014.2330554`.

[35] V. Vapnik, ”An Overview of Statistical Learning Theory”, *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, Sep. 1999. DOI: `10.1109/72.788640`.

[36] C. Cortes and V. Vapnik, ”Support-Vector Networks”, *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. DOI: `10.1023/A:1022627411411`.

[37] F. F. Chamasemani and Y. P. Singh, ”A New Boosting Multi-class SVM Algorithm”, *International Journal of Advanced Research in Computer Science*, vol. 4, no. 2, pp. 01–06, Feb. 2013.

[38] I. Steinwart and A. Christmann, *Support Vector Machines*, M. Jordan, J. Kleinberg, and B. Schölkopf, Eds. Springer, 2008.

[39] S. Shirali and H. L. Vasudeva, *Metric spaces*. Springer, 2006. DOI: `10.1007/1-84628-244-6`.

[40] A. Airola, T. Pahikkala, W. Waegeman, *et al.*, ”An experimental comparison of cross-validation techniques for estimating the area under the ROC curve”, *Computational Statistics and Data Analysis*, vol. 55, no. 4, pp. 1828–1844, 2011. DOI: `10.1016/j.csda.2010.11.018`.

[41] B. J. Parker, S. Günter, and J. Bedo, ”Stratification bias in low signal microarray studies”, *BMC bioinformatics*, vol. 8, no. 1, p. 326, 2007. DOI: `10.1186/1471-2105-8-326`.

[42] J. L. Devore, *Probability and Statistics for engineering and the sciences*, 7th ed. Brooks/Cole Cengage Learning, 2009.

[43] P. M. Fernandes, W. N. Whiteley, S. R. Hart, *et al.*, ”Strokes: mimics and chameleons”, *Practical neurology*, vol. 13, no. 1, pp. 21–28, Feb. 2013. DOI: 10.1136/practneurol-2012-000465.

[44] C.-C. Chang and C.-J. Lin, ”LIBSVM: A library for support vector machines”, *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, 27:1–27:27, 3 May 2011, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. DOI: 10.1145/1961189.1961199.

[45] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, ”A Practical Guide to Support Vector Classification”, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan, Tech. Rep., Apr. 2010. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf (visited on 05/05/2014).

[46] X.-L. Xia, M. R. Lyu, T.-M. Lok, *et al.*, ”Methods of Decreasing the Number of Support Vectors via k-Mean Clustering”, in *Advances in Intelligent Computing*, ser. Lecture Notes in Computer Science, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds., vol. 3644, Springer Berlin Heidelberg, 2005, pp. 717–726, ISBN: 978-3-540-28226-6. DOI: 10.1007/11538059_75.

[47] S. F. Crone, S. Lessmann, and R. Stahlbock, ”The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing”, *European Journal of Operational Research*, vol. 173, no. 3, pp. 701–800, 2006. DOI: 10.1016/j.ejor.2005.07.023.

[48] T. Eitrich and B. Lang, ”Efficient optimization of support vector machine learning parameters for unbalanced datasets”, *Journal of Computational and Applied Mathematics*, vol. 196, no. 2, pp. 425–436, Nov. 2006. DOI: 10.1016/j.cam.2005.09.009.

[49] Z. Xu, K. Q. Weinberger, and O. Chapelle, ”Distance Metric Learning for Kernel Machines”, arXiv, Tech. Rep., Jan. 2013. [Online]. Available: http://arxiv.org/abs/1208.3422v2.