THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Classification of High Dimensional Signals with Small Training Sample Size

with Applications towards Microwave Based Detection Systems

Yinan Yu

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2013

Classification of High Dimensional Signals with Small Training Sample Size

with Applications towards Microwave Based Detection Systems

Yinan Yu

Technical report number: R016/2013
ISSN 1403-266X

Department of Signals and Systems
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone + 46 (0)31 – 772 1000

Typeset by the author using LaTeX.

Printed by Chalmers Reproservice
Göteborg, Sweden 2013

to my parents

Abstract

Classification techniques attempt to resolve the problem of categorizing data into two or more classes. The data distribution is therefore the most critical fact to be aware of. Unfortunately, specifications of data generators are not available in real life and a probabilistic density parameterization is not always applicable, especially for the situation of High Dimensional data with Low (training) Sample Size (HDLSS). This raises the importance of developing data driven techniques, where the data model is assumed according to partially accessible prior knowledge or cross-validation. There are various popular data assumptions, such as centroid-based models, linear subspace models, manifold data structures, etc, and one should take into consideration the model accuracy, computational complexity, generalization ability, and be aware of possibilities of overfitting. When the dimensionality of the data is much higher than the training sample size, all issues appear as its nature and there is no easy way to find a good trade-off.

In this work, we mainly focus on the first two types of data models and develop corresponding classification techniques. The first objective is to automatically learn the data generating model with limited amount of training samples available. With the assumed data model, the second step is to maximize the class separability with respect to the model assumption. The applications studied encompass both simulated and measured microwave signals for stroke type diagnostics and wood quality assessment. The results are analyzed and compared with more classical approaches.

Keywords: Linear Subspace Learning, Discriminant Analysis, Class Separability, Classification, High Dimensional Data, Small Sample Size

Publications

This thesis is based on the following appended papers:

Paper 1 Yinan Yu, Tomas McKelvey, A Subspace Learning Algorithm For Microwave Scattering Classification With Application To Wood Quality Assessment, *Proceedings of IEEE International Workshop On Machine Learning For Signal Processing*, September, 2012, Santander, Spain

Paper 2 Yinan Yu, Tomas McKelvey, Sun-Yuan Kung, A Classification Scheme For 'High-Dimensional-Small-Sample-Size' Data Using SODA And Ridge-SVM With Microwave Measurement Applications, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May, 2013, Vancouver, Canada

Paper 3 Yinan Yu, Tomas McKelvey, A unified subspace classification framework developed for diagnostic system using microwave signal, *submitted to 21st EUSIPCO, 2013*

Other publications:

Paper 4 Yinan Yu, Tomas McKelvey, A Classification Scheme Based on Generalized Subspace Model for Microwave Scattering Signal Classification, *submitted to Journal of Integrated Computer-Aided Engineering (ICAE)*

Paper 5 Andreas Fhager, Yinan Yu, Tomas McKelvey, Mikael Persson, Stroke Diagnostics with a Microwave Helmet, *7th EuCAP*, April, 2013, Gothenburg, Sweden

Paper 6 Stefan Candefjord, Johan Winges, Yinan Yu and Tomas McKelvey, Microwave technology for localization of traumatic intracranial bleedings a numerical simulation study, *accepted to 35th Annual International IEEE EMBS Conference*, July, 2013, Osaka, Japan

Contents

II Included Papers

Part I

Introduction

Chapter 1

Introduction

This thesis is organized as follows. The first chapter gives an introduction of the ongoing projects. Followed by motivations, a first grasp of machine learning techniques is attempted. Chapter 2 gives a brief review on the challenges and existing solutions. In Chapter 3, we summarize the contribution and included papers.

1.1 Statistical classification

Statistical classification is the subject of finding an automatic way of identifying an unknown data vector under the assumption that it has been generated from a set of the recognized categories. This is based on the model assumptions obtained from domain background, experience, and the statistical analysis of the training data set. As shown in Figure 1.1, the training data set contains all the observations with a known category label. The free parameters in the classification model are estimated from the training data.

Before we proceed, we have to assume that there is always an intrinsic structure in the data, i.e. the data generating model is not completely random. It does not matter if it is hidden or obvious, a hyperplane or a manifold, two dimensional or with infinite dimensions, etc. We might not be able to see it with our Euclidean eyes, but the structure is out there. This assumption motivates us to explore and extract the information contained in the data in some smart way.

Spam detector – an example

One classic real-life example is the spam detector. An email is identified as a spam if the output of the classifier y is +1 and 'not a spam' if -1. The output y is called 'label'. The input vector is called the feature vector which contains the information visible to the classifier. In this case, the feature vector could contain some classical keywords, such as 'win', 'dollars' or 'money'. Before a spam detector can be applied with satisfaction,

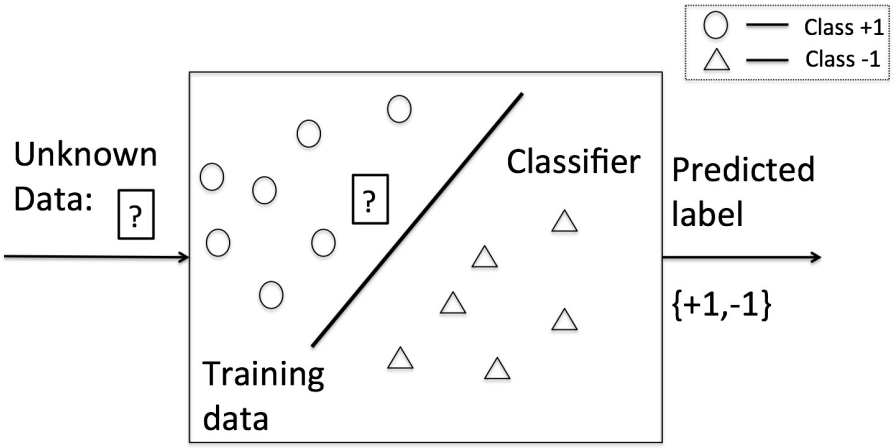


Figure 1.1: The solid line represents a classification model splitting the training data into two subsets. The classifier is then applied to identify an unknown object.

two main steps are required: training and testing. *Training* is analogous with our learning procedure. With a model in mind, the machine takes the features from all the emails as inputs and being told if those are spams or not. After learning from enough examples, we decide to test if the computer is 'smart' enough to identify unknown emails. This is called the 'testing' step. Once we obtain statistically 'good' testing results with confidence, the construction of the classifier is complete.

In this chapter, we will briefly introduce the concept of classification techniques, then proceed with a more detailed review in later chapters.

1.2 Machine learning for classification: An introduction

1.2.1 Why machine learning?

We as human are rigorous but lazy creatures. From observations and evidences, we want the most reasonable inferences without being involved in the fussy and boring calculations. We need aid from other intelligent entities: the computers. We urge them to learn, to decide, to behave fast and accurately. This is where machine learning comes in and quickly becomes one of the most popular and exciting subjects.

There are so many problems that a computer can learn to solve. According to the type of required outcomes, these problems mainly fall into two categories: regression and classification. As shown in Figure 1.2, given an input data set, the main difference of the two is the continuity of the desired output y. The problem is called regression when the output is continuous (real or complex numbers) and classification otherwise ($y \in \{-1, +1\}$). Intuitively, regression solvers are evaluated by the 'goodness of fit'

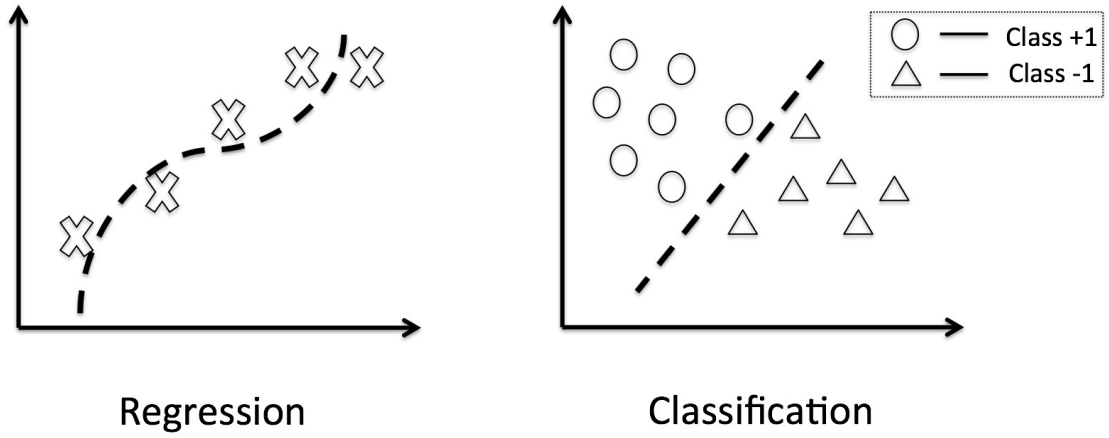


Figure 1.2: Two classic problems machine learning can be applied to: [left] regression and [right] classification

[1, 2] and classifiers are generally assessed by the probability of detection (P_D) along with the probability of false alarm (P_{FA}) [3]. The focus of the thesis work is on the development of classification techniques.

1.2.2 Machine learning for classification

Figure 1.3 shows the flow involved in solving a classification problem in general. Given a set of p dimensional input vectors

$$\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\} \tag{1.1}$$

where

$$\boldsymbol{x}_i = \left[x_1, \cdots, x_p\right]^T \in \mathbb{C}^p \tag{1.2}$$

and the corresponding label y (+1 or -1) for each $\boldsymbol{x}_i$, we would like to train a computer to associate an unknown testing data with the correct label. Before we can work on the classification models, we need to take a look at the measurements we get from the lab, for example, images, radar signals, stock data, or medical testing results. They are given by the specialists and we assume that informative input $\boldsymbol{x}$ is preliminarily extracted according to the specific domain. Unfortunately, however, the acquired data is never clean, compact nor illustrative enough. This raises the importance of feature selections and transformations which can be applied to extract or enhance the desired information.

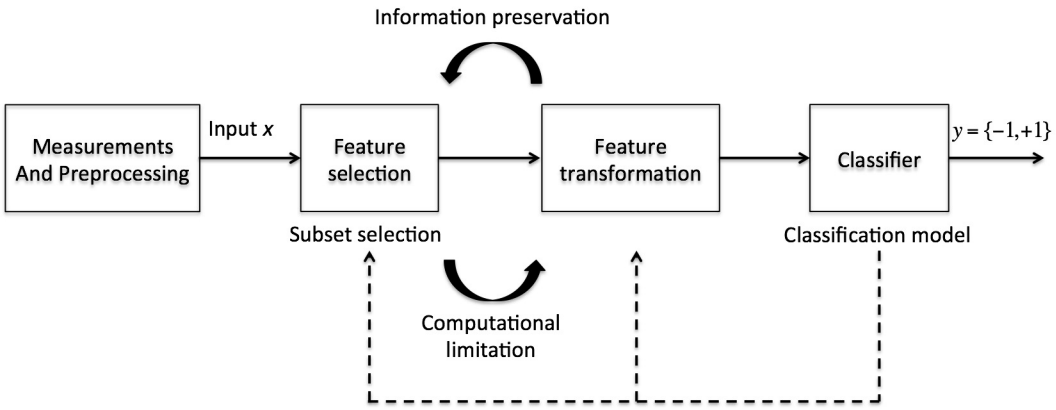


Figure 1.3: The flow chart of the development of classification techniques. There are three major steps: the feature selection, feature transformation and classifier.

Feature selection

Features are the multi-dimensional variables used as the input vector of the classifier. Feature selection attempts to select a subset of the coordinates of the vector with respect to some predefined criteria. A review of some well known techniques can be found in [4]. Generally speaking, the approaches mainly fall into 3 classes: the wrappers, filters and embedded methods.

- Wrappers: strongly correlated with the classification model, i.e. they use the output of the classifier as feedback to optimize the subset selection procedure.
- Filters: selecting the features by evaluating feature scores (e.g. Fisher score) based on some ranking systems; in general independent of the classification model.
- Embedded methods: where the selection is a part of the classification model (e.g. LASSO based feature selection [5]).

Notice that by definition, feature selection only involves selecting a subset of the original features, which means it selects certain coordinates of the original data vector. No data transformations are performed. This is the difference between feature selection and transformation techniques.

Feature transformation

Feature transformations are information extraction techniques. They transform the data from the original vector space to a new feature space, in which the features are more organized, informative and the class separability is hopefully enhanced. There are linear,

multi-linear and nonlinear techniques. Classical examples include Principal Component Analysis (PCA) which assumes the data lie on a linear subspace; tensor based data representations [6], and manifold learning techniques [7].

The order of the feature selection and transformation steps can be switched according to the objectives and challenges. Since feature selection discards dimensions, better preservation of information can be achieved by placing feature transformation ahead of it. However, for computational reasons, sometimes it is impossible to process the raw data if the dimension is very large. In this case, discarding some data is necessary before any transformations.

Classification model and learning techniques

The main purpose of a learning technique is to build a classifier with high performance. That is, given a constructed feature vector $\boldsymbol{x}_i \in \mathbb{C}^p$, we would like to find a function $f : \mathbb{C}^p \to \mathbb{R}$, which maps the features to a binary label or a discrete output in the multiclass case.

- Classification model: generative vs discriminative models:
  In statistical learning, the classification models fall into two categories: generative models and disciminative models.

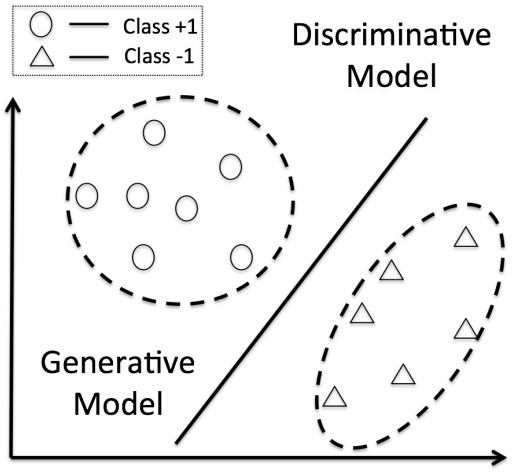


Figure 1.4: One illustrative example of the comparison between generative model and discriminative model.

Generative models learn a model of the joint probability distribution $P(\boldsymbol{x}, y)$ and the generalization ability reflects on the probability detection by computing the conditional probability $P(y|\boldsymbol{x})$ on the testing data and picking the label y corresponding to the largest $P(y|\boldsymbol{x})$. On the other hand, discriminative model learns

the posterior $P(y|\boldsymbol{x})$ directly. As shown in Figure 1.4, discriminative model draws a line to split the data from two classes, while generative classifier models the two distributions explicitly. It is believed that a discriminative model is superior compared to a generative model if the data model is not well chosen and estimated. An introduction and comparison can be found in [8].

- Learning techniques: supervised learning vs unsupervised learning:
  Supervised learning models are contrary to unsupervised learning approaches with respect to the visibility of the labels for training data. In supervised learning, the labels are known to the computer for training. Figure 1.5 shows the general idea. Usually, classifiers are trained using supervised learning techniques, while when the unsupervised learning are used, the technique is usually called clustering.

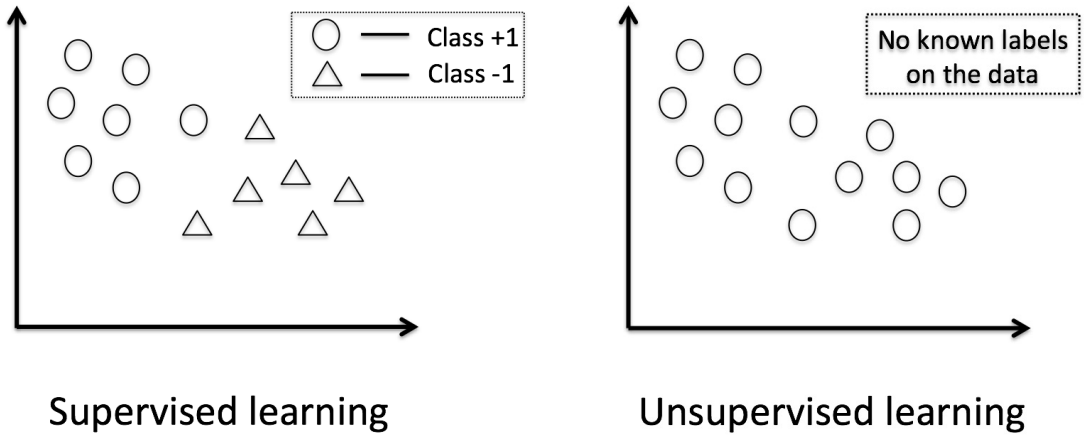


Figure 1.5: One illustrative example of the comparison between supervised learning model and unsupervised learning model.

Evaluation of experimental results

The performance of a classifier is evaluated on testing samples which are independent of the training data used for tuning the classifier. There are many ways of evaluating classification results. Throughout this thesis, the results are evaluated by the probability of detection P_D and probability of false alarm P_{FA}:

$$P_D = \frac{\#\text{ of correctly classified samples from class +1}}{\#\text{ of samples from class +1}} \tag{1.3}$$

$$P_{FA} = \frac{\#\text{ of incorrectly classified samples from class -1}}{\#\text{ of samples from class -1}} \tag{1.4}$$

If we plot P_D versus P_{FA}, we have a curve called Receiver Operating Characteristic (ROC) shown in Figure 1.6. One fixed classifier will give one point in the P_D and P_{FA} space. However, if we parametrize the classifier with an extra parameter (which influence the performance), a sweep with this parameter will yield the ROC curve. The larger the area under the curve is, the better the classifier behaves.

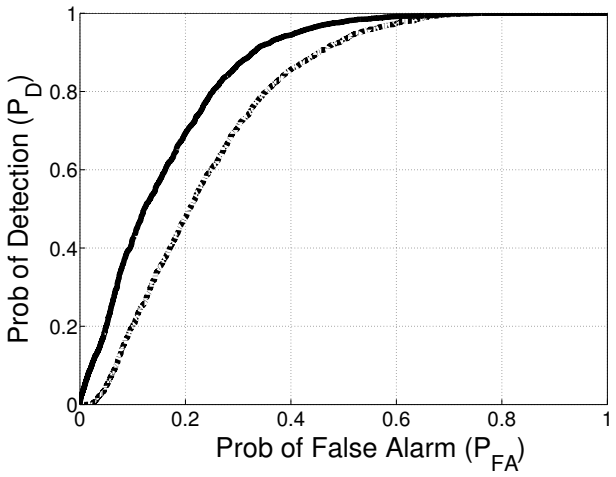


Figure 1.6: One example of a Receiver Operating Characteristic (ROC) curve. The two curves indicate the performance of two classifiers. The solid curve covers a larger area and thus has a better performance compared to the dashed curve. Also in this case, for any fix false alarm rate, the solid curve has a higher P_D which makes it consistently better.

With a high performance on the testing data, the classifier is considered capable of identifying samples from each class and is able to be applied to real applications. Clearly, the performance on the predictive capability is of our main interest, since it is not especially impressive if the classifier tells you what is already known. This is usually called generalization ability, which is an important measure for classification algorithms.

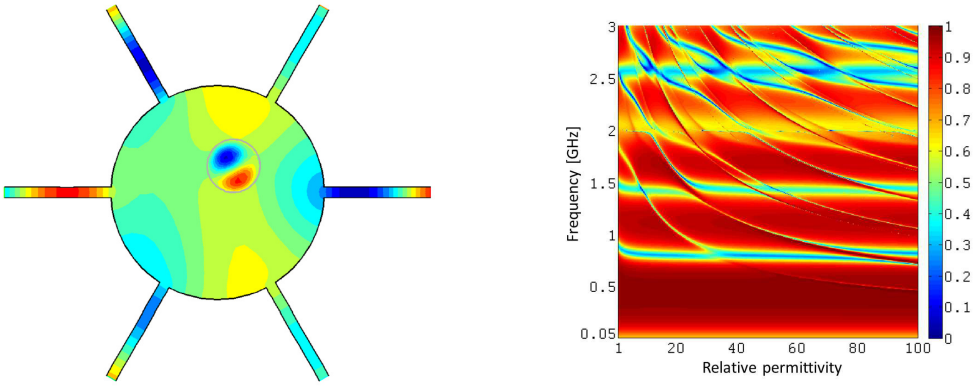
1.3 Microwave based detection system

The applications involved are based on Multi-Input-Multi-Output (MIMO) microwaves systems equipped with wide-band transceivers [9, 10]. Real life applications are normally in three dimensions, i.e. a cylinder or a ball shaped cavity. Nevertheless, one illustrative example in two dimensional space is shown in Figure 1.7(a). There are 6 antennas behaving as both transmitters and receivers in an alternating fashion, i.e. only one antenna is transmitting signals at one time. Inside the boundary, there is an object

with higher permittivity. Shaded level shows example field solution (H_z-component at 1 GHz) when the right-most port is excited. Note that when antenna p behaves as a transmitter, and antenna q is listening to the transmitted signal, the antenna pair (p, q) is called a 'channel'. The signal measured at each channel (p, q) is the scattering parameter denoted by $S_{p,q}(\omega)$ in the frequency domain, which is defined as:

$$S_{p,q}(\omega) = -\frac{H^-_{0,p}(\omega)}{H^+_{0,q}(\omega)} \tag{1.5}$$

where $H^-_{0,p}(\omega)$ is the complex amplitude of the fundamental mode at frequency ω of the outgoing wave at port p and $H^+_{0,q}(\omega)$ is the system excitation amplitude at port q.



(a) (b)

Figure 1.7: (a): An example of a port-to-port system with six sensors mounted on a circular 'cavity'. (b): The corresponding propagation of microwaves inside the cavity, i.e. the magnitude of one of the S parameters as a function of frequency and relative permittivity.

The measured S parameters at frequency point ω_i over all the N_c channels construct the S - matrix as:

$$\boldsymbol{S}(\omega_i) = \begin{bmatrix} S_{1,1}(\omega_i), & \cdots, & S_{1,N_c}(\omega_i) \\ S_{2,1}(\omega_i), & \cdots, & S_{2,N_c}(\omega_i) \\ & \vdots & \\ S_{N_c,1}(\omega_i), & \cdots, & S_{N_c,N_c}(\omega_i) \end{bmatrix} \tag{1.6}$$

If the microwave network is passive and reciprocal then $\boldsymbol{S}(\omega_i) = \boldsymbol{S}(\omega_i)^T$.

In Figure 1.7(b), the colors indicate the (normalized) amplitude of the $S_{1,1}$ parameter defined in Eq.(1.5) as a function of permittivity over all the frequencies from 0.05 GHz to 3 GHz. As we can see, there are apparent patterns in the frequency responses as a

function of changing permittivities, from which we would like to explore and associate to certain properties of the measured object. Of course, there are challenges, such as the robustness, the high dimensionality of the raw signal, the complex properties of the object, etc.

1.3.1 Measurements and prototypes

Algorithms are aiming to serve real world applications after all. While simulations assist us with analyzing the limitations and properties of the algorithms and techniques, empirical testing is one of the most powerful tools which evaluates the theoretical development. In this section, we are going to introduce the background of some ongoing projects.

Stroke diagnostics

Strokes can be mainly classified into two types: ischemic stroke and hemorrhagic stroke. They are caused respectively by interruption of the blood supply or the burst of blood vessels. Special treatments are required depending on the type of the stroke. Standard techniques for diagnosing and monitoring stroke are EEG, CT, MRI, and fMRI. Common for these modalities is the relatively involved procedures and the associated time involved not only in the procedure itself but also to get access.

Figure 1.8: The measurement device used in the stroke type detection project.

The closest technique with respect to level of technology and accessibility is ultrasound. However, in contrast to microwaves, ultrasound face problems passing through the scull bones. Similar problems are observed for near Infrared imaging and spectroscopy and impedance measurements and impedance tomography. This raises the

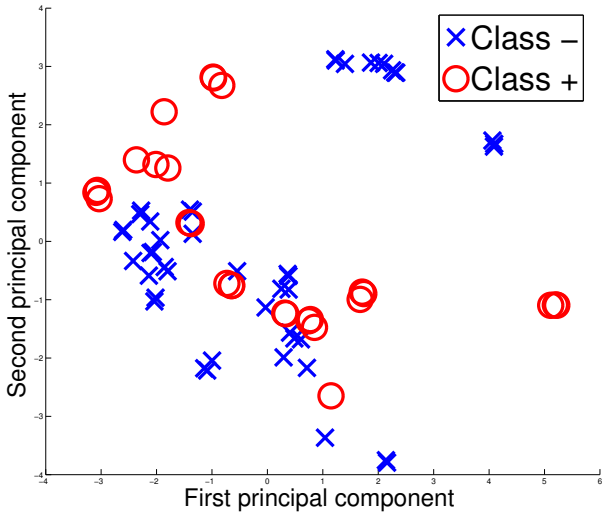


Figure 1.9: The first two principal components (obtained using PCA) of the raw measurement. As we can see that linear separability is not always guaranteed.

importance of our application. The purpose is to apply point-of-care concepts to fast and efficient stroke diagnosis using equipment with low cost.

As shown in Figure 1.8, the space contained in the measurement device is the cavity that we would like to measure. Twelve antennas are mounted inside the cavity with a water bag between each antenna and the head. The sensors are transmitting and receiving signals in turns and the measured signals are the complex S parameters defined in Eq.(1.5). Due to the penetration ability of the microwave signals, the internal properties of the brain can be perceived by the sensors. On the other hand, from a classification point of view, the measured signals cannot be used as feature vectors without any processing. For visualization purposes, in Figure 1.9, we apply a classical dimensionality reduction technique called Principal Component Analysis (PCA) [11] to obtain the projection of the high dimensional signal on the first 2 principal components. If the data set is linearly separable in this 2 dimensional space, the classification problem will be relatively simple to solve.

However, as we can see from Figure 1.9, the first two principal components are clearly not separable in this case. This raises the importance of developing robust classification algorithms to be able to extract useful information and overcome the classification difficulties. Another challenge is that the dimensionality of the raw measurement is gigantic ($\sim 3 \times 10^4$) and possibly full of redundant information. One example of the amplitude of $S_{1,1}$ as a function of frequency is shown in Figure 1.10.

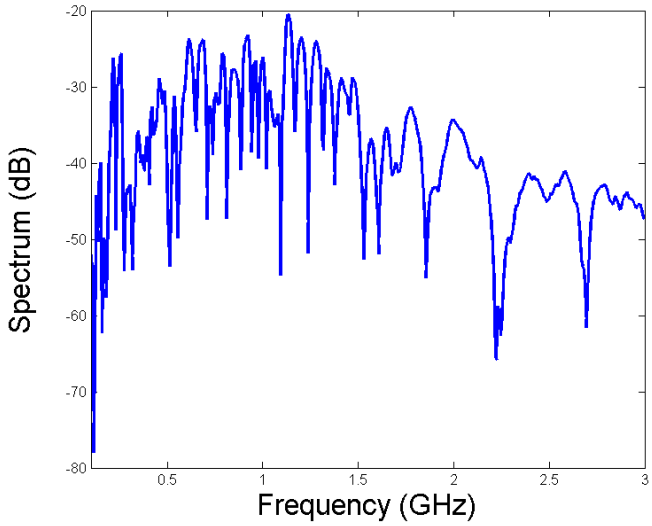


Figure 1.10: The magnitude of measured $S_{1,1}$ over a frequency bandwidth of $0.1 - 3\ GHz$. In this case, the signal is transmitted by the first antenna and received by itself

.

Wood quality assessment

Another application is wood quality assessment using microwave measurements. With or without our awareness, timber industry plays a key role, both in our daily life and industrial uses as a great sustainable resource. For example, a large amount of market requirement comes from the construction industry, chemistry, transportation, packaging industry, etc. To reduce unnecessary costs, there is an increasing demand of automatically assessing the quality of timber before any processing.

Figure 1.11: This picture shows an example of (a) perfectly healthy log, (b) half rotten log and (c) rotten log

.

Some examples from a sawmill can be found in Figure 1.11. As we can see that the condition of the log is very hard to identify from outside. The aim of this application

is to provide a classifier for the timber industry to identify healthy logs with low false alarm rate. The principal setup of the microwave system is illustrated and described in Figure 1.12.

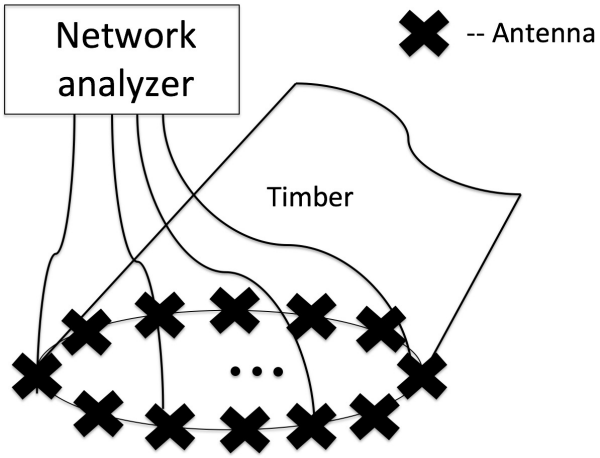


Figure 1.12: This figure shows the principal setup of the wood quality assessment project. The cylinder material indicates the log we would like to evaluate. We place the 'belt' like antennas around the log and all of them are connected to a network analyzer. The mechanism of the system has been briefly presented in the previous section.

1.3.2 Simulations

Some properties of the system can be analyzed using computer simulations by modeling the microwave propagation with respect to the given boundary conditions. With the simulated signals, various physical properties of the wave behaviors and experimental analysis of the algorithms can be tested with full control over the adjustable parameters, which allows us to explore different possibilities.

For instance, a geometric setup can be found in Figure 1.13, where a two dimensional model of a patient's skull is simulated. In this simplified setting, one spot of bleeding area is presented on the right half of the brain-model. To provide different scenarios, we have different bleeding sizes, positions and head shapes. The corresponding microwave responses are computed and simulated.

To summarize, this type of systems can be used to inspect unknown properties from the target objects. The applications considered in this thesis are automatic stroke diagnostics system and wood quality assessment. Simulations are used to further explore and analyze the behavior of the classification algorithms.

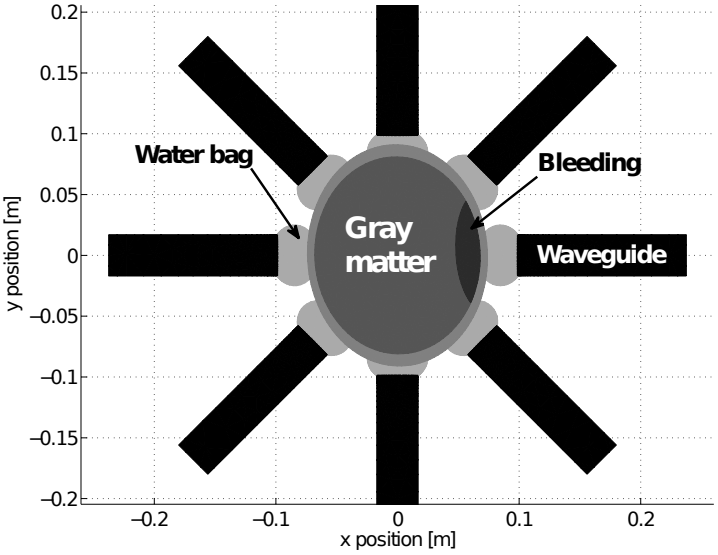


Figure 1.13: The geometry of a two dimensional model of a patient's skull used for simulation.

Chapter 2

Classifiers for HDLSS: A Brief Review

2.1 Challenges

For any machine learning problem, there always exist inevitable challenges. Such as:

- Learning with large-scale data: we acquire larger and larger data set every day. For example, in financial applications, we have huge amount of data every second and we have to process them fast. We need to extract useful information, deal with outliers, recognize patterns and make real time decisions.
- Curse of dimensionality: problems occur when the dimensionality of the data is out of our storage or computational capacity. In such cases, the running time of the algorithms might explode.
- Highly redundant data set: the redundancy in the data might blind our algorithms and thus causes inefficient learning and computations.
- Low sample size: for some applications, we have an opposite yet even more disturbing situation called 'low sample size' problem, which means we can only acquire a certain amount of samples for training. In such scenarios, the distribution of the data cannot be fully explored and issues such as difficulty in testing, over fitting, and insufficient statistical description occur.
- Missing data: missing data is a common problem in real-life applications. Some dimensions might not be available or reliable during measurements which can be identified using domain knowledge.

The main focus of this work are possible solutions to the 'curse of dimensionality' and 'low sample size' problems. In some literature it is called the 'High Dimensionality with Low Sample Size' (aka: HDLSS) scenario. Some classical techniques are reviewed in Section 2.2.

2.1.1 What will not be covered

The focus of this thesis work is the development of classification techniques for tackling the HLDSS issue. Therefore, there are certain topics that will not be covered:

a. Feature selection and feature transformation techniques for dimension reduction (see. Section 1.2.2).

b. Unsupervised approaches for clustering.

2.2 Brief review: $p \gg N$

In some literature, HDLSS is described as the $p \gg N$ problem, where p is the number of dimensions in a data vector and N is the sample size. The problems of large dimensionality and low sample size are closely connected. Good survey regarding this issue and some existing solutions can be found in the following references: [12, 13, 14, 15, 16].

2.2.1 HLDSS issue

It is a twofold problem. First, in statistical learning, the significance of the statistics plays a key role. When the sample size is extremely small, any estimated parameters of the assumed model cannot be fully trusted. The 'rule of thumb' saying that 'the minimum sample size N is 30' turns out to be a myth [17]. In such cases, we should be very careful about what the algorithms have actually learned from the limited data. On the other hand, when the data vector has extremely high dimensions, the algorithms suffer from computational issues (e.g. complexity and singularity). As a general rule, the larger the dimensionality, the more samples are required for the algorithms. Roughly speaking, problems occur in the HLDSS case, such as:

i) The global distribution is impossible to grasp. Namely, it is extremely difficult to establish a statistical model with high confidence levels based on a few samples of training data.

ii) Outliers are hard to detect.

iii) Covariance structure of high dimensional data vectors is captured by the covariance matrix. When the sample size is extremely small compared to the dimensionality, the covariance matrix cannot be estimated properly, due to the facts that 1) it is computationally impossible; 2) there are not enough samples to catch the covariance structure; 3) the estimation error is related to the level of $\frac{p}{N}$ [18].

iv) Information redundancy. For large number of p, redundant dimensions are mixed with the discriminant information, which makes the classifier less efficient. For

example, it has been shown [19] that when the dimensionality is extremely high, the generalization ability of SVMs is degraded.

v) Overfitting is not evitable.

With the increasing data capture and storage capacity, it is quite often that we acquire data with high dimensions, such as high resolution images, genetics analysis, medical tomography and etc. There are many people working on possible solutions of such issue over the decade:

- For both theorists and practitioners, a linear subspace model can be a reasonable choice. For data driven approaches, without any prior knowledge of the data generating function, a linear assumption is simple and turns out to be very practical in real world applications.
- The data structure can be explored in a localized fashion.
- Some techniques are designed for dealing with outliers by assigning weights to each data points with respect to predefined criteria.
- Feature selection and reduction techniques.
- Regularization techniques as constraints on the dimensionality.

Some technical details and corresponding references are introduced in Section 2.2.2.

2.2.2 Review of classical techniques

Given training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \mid \boldsymbol{x}_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\}, i = 1, \cdots, N\}$, where y_i is the class label of vector $\boldsymbol{x}_i$, a model with parameters $f(\mathcal{D}; \boldsymbol{\theta})$, we would like to train a classifier to predict an unknown label $\hat{y}$.

Support Vector Machine (SVM)

Support Vector Machine (SVM) [20, 21] is probably one of the most well known classification techniques. Intuitively, the idea is to find a hyperplane with maximum margin, which separates $\boldsymbol{x}_i$'s of class $y_i = +1$ from $y_i = -1$. A good description of numerical details can be found in [22] and also summarized as follows.

Let us first consider the simplest scenario. Assume the data vectors are linearly separable, which means that the training vectors from the two classes can be separated in the high dimensional space with a hyperplane. This hyperplane $f_{hyp}(\boldsymbol{x})$ satisfies the following relation between: 1) the normal vector to the hyperplane $\boldsymbol{w}$, 2) the training data $\boldsymbol{x}_i$ and 3) a bias b:

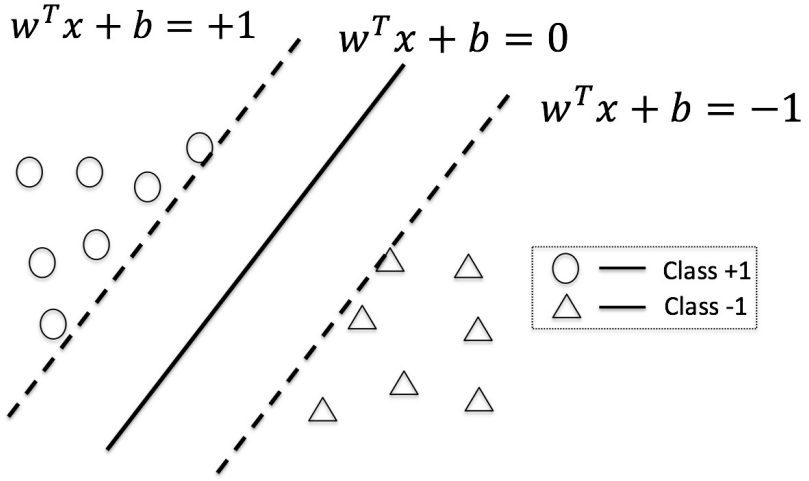


Figure 2.1: An illustration of SVM in two dimensions.

$$f_{hyp}(\boldsymbol{x}) \equiv \boldsymbol{w}^T \boldsymbol{x} - b = 0 \tag{2.1}$$

Define two such hyperplanes as shown in Figure 2.1:

$$\begin{aligned} \boldsymbol{w}^T \boldsymbol{x}_i + b \geq +1, \ \text{ for } \ y_i = +1 \\ \boldsymbol{w}^T \boldsymbol{x}_i + b \leq -1, \ \text{ for } \ y_i = -1 \end{aligned}$$

which means, to describe all the data points in the feature space, we have:

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1, \ \ \forall i \tag{2.2}$$

And the margin between the two hyperplanes is $2\frac{1}{\|w\|}$. Therefore, to summarize the idea of SVM, we want to:

- Maximizing the margin between the two data sets;
- Keeping the training data from each class on the right side of the hyperplane.

The maximum margin SVM can be defined as a constrained optimization problem:

$$\begin{array}{ll} \underset{\boldsymbol{w},b}{\text{minimize:}} & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \\ \text{subject to:} & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1, \forall i \end{array}$$

However, when the dimensionality of vector $\boldsymbol{x}_i$ is high, the computations can be very time consuming. Therefore, to solve the optimization problem, the dual can be formulated as follows.

Derive the optimality from the dual: The Lagrangian of this constrained optimization problem is formulated as:

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_i a_i(1 - (y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)) \tag{2.3}$$

with variables $\boldsymbol{w}$, b and a_i are the Lagrange multipliers. In order to get rid of the dependency on the original variables, the partial derivatives are computed:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} = \boldsymbol{w} - \sum_i a_i y_i \boldsymbol{x}_i = 0, \quad \frac{\partial\mathcal{L}}{\partial b} = \sum_i a_i y_i = 0 \tag{2.4}$$

$$\text{The Lagrangian becomes: } \mathcal{L}(\boldsymbol{a}) = \sum_i a_i - \frac{1}{2}\sum_{l,k} a_l y_l \underbrace{\boldsymbol{x}_l^T\boldsymbol{x}_k}_{\boldsymbol{K}_{l,k}} y_k a_k \tag{2.5}$$

The matrix $\boldsymbol{K}$ is the Gram matrix of all possible inner products of $\boldsymbol{x}_i$, $\forall i$.
The dual problem can thus be formulated as:

$$\begin{array}{ll} \underset{\boldsymbol{a}}{\text{minimize:}} & \frac{1}{2}\sum_{l,k} a_l y_l \boldsymbol{K}_{l,k} y_k a_k - \sum_i a_i \\ \text{subject to:} & \sum_i a_i y_i = 0 \\ & a_i > 0, \quad \forall i \end{array}$$

By solving the Lagrangian dual problem with respect to vector $\boldsymbol{a} = \begin{bmatrix} a_1, a_2, \cdots, a_N \end{bmatrix}^T$, whose dimensionality is associated with the sample size $\dim(\boldsymbol{a}) = N < p < \infty$ and ergo consumes less computational power, together with Equation (2.4), the optimal solution of the variables $\boldsymbol{w}$ and b in the primal problem can be obtained. However, note that the computational complexity grows quadratically [23]. When the training size is large ($N \gg 10000$), the computation is extremely heavy.

On the other hand, SVMs have the advantage of computational simplicity in $p > N$ scenarios. However, when the dimensionality of the feature vector is much higher than the sample size ($p \gg N$), although with a perfect separation on the training sets, the generalization ability might be degraded [19]. One reason is that the global distribution of the data set is not visible to the classifier. Therefore, without any modification, SVM is not a good candidate for solving the extremely low sample size problem.

Kernel trick

Kernel tricks can be applied to achieve a better performance. The kernel trick [24, 25, 26] is a way of mapping original feature vectors into a new inner product space. This

map ϕ: $S \to V$ is defined as:

$$\boldsymbol{K}_{ij} = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle \tag{2.6}$$

where $\phi(\boldsymbol{x})$ maps vector $\boldsymbol{x}$ to a Reproducing Kernel Hilbert Space (RKHS) [25, 27] and K is called the kernel function, which represents the inner product in the $\phi(\boldsymbol{x})$ space. It can be considered as a similarity measure.

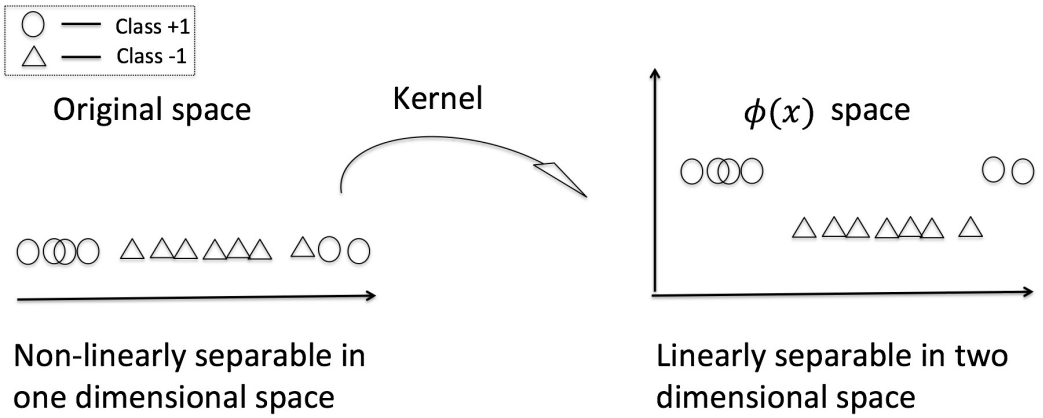


Figure 2.2: Kernel tricks map the data set to a higher dimensional vector space equipped with inner product K without adding any computational complexity.

The main advantages of kernel tricks include:

- neither the mapping nor the high dimensional inner product needs to be computed explicitly;
- (hopefully) the data sets are linearly separable in the new inner product space V equipped with a chosen kernel, where only the existence of $\phi()$ need to be guaranteed. If the kernel is well chosen, the complexity of the learning model can be reduced dramatically with a higher performance.

Artificial Neural Networks (ANN)

Another classic example is the Artificial Neural Networks (ANN) [28, 29, 30, 31, 32, 33]. A simple illustration of ANN can be found in Figure 2.3. Retrospect to 1943, the famous McCulloch and Pitt published a paper titled "A logical calculus of the ideas immanent in nervous activity" introducing the concept of perceptron, which was then followed by a book called "The Organization of Behavior" by Donald Hebb in 1949. These publications tried to explain and even mimic how the human brain works.

So it begins. As the continuous development of the computational power and due to the mysterious and fantastic nature of human brains, there was a neural network frenzy

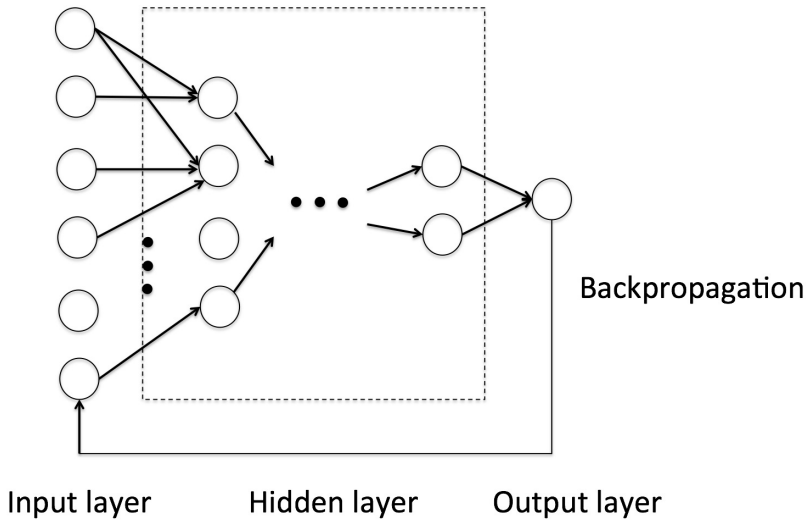


Figure 2.3: The nodes are called neurons. There are 3 types of layers: the input layer, the hidden layer and the output layer. Input vectors with N dimensions

.

during the 80's and 90's. It cooled down afterwards due to the fact that people realized they could not grasp and control the learning neurons in a systematic way. Nevertheless, the neural networks are still one of the most intuitive and recognized learning machines.

The main reason that ANN is introduced in this section is that it represents a synthetic approach which includes linear and non-linear classifiers as special cases. Not only is it mimicking the way of how our brain works, which makes it intuitive, but also gives graphical structures which is always preferable.

The simplest neural network is called the Perceptron Algorithm [34]. As shown in in Figure 2.3, without any hidden layers, the output is represented by a linear expression as follows:

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} \tag{2.7}$$

where $\boldsymbol{x}$ is the input vector and $\boldsymbol{w}$ denotes the vector containing the weights, which are the parameters to be estimated and optimized. This is usually done by the backpropagation algorithm [32].

A simplest decision rule can then be:

$$y = \begin{cases} +1 & \text{if } f(\boldsymbol{x}) > 0 \\ -1 & \text{otherwise} \end{cases} \tag{2.8}$$

Study shows that the generalization error of multi-layer neural networks does not increase for linearly increasing $\frac{p}{N}$. However, the computational complexity for the train-

ing step is an issue, not to mention the difficulty of the structure design. Moreover, overfitting problem is hard to tackle when $\frac{p}{N}$ is extremely large.

Finally, as a personal note, I am not saying I am a huge fan of ANN. But I did not enjoy listing all the references for nothing neither. It is an attempt of human being to understand and even mimic the functionality of our brain. It is an epic shot after all.

Linear Discriminant Analysis (LDA)

The idea behind Linear Discriminant Analysis (LDA) [35, 36] is to find a linear transformation $\boldsymbol{w}$ which projects the data sets onto a one dimensional subspace where the class separability is maximized. One intuitive example is shown in Figure 2.4.

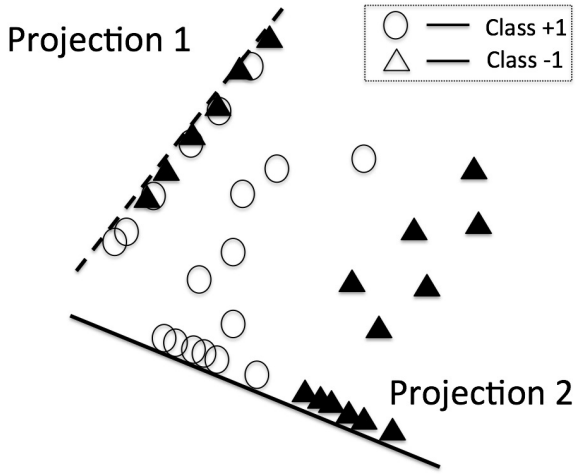


Figure 2.4: Linear Discriminant Analysis: learning an optimal linear transformation corresponding to the best separation ability. We can see clearly that projecting all the data points to the solid one dimensional space, the class separability could be improved.

The separability is measured over the 'between-class scatter matrix' $\boldsymbol{S}_B$ and the 'within-class scatter matrix' $\boldsymbol{S}_W$, which are respectively defined as:

$$\boldsymbol{S}_W = \frac{1}{N_+}\sum_{i=1}^{N_+}(\boldsymbol{x}_i^+ - \boldsymbol{\mu}^+)(\boldsymbol{x}_i^+ - \boldsymbol{\mu}^+)^T + \frac{1}{N_-}\sum_{j=1}^{N_-}(\boldsymbol{x}_j^- - \boldsymbol{\mu}^-)(\boldsymbol{x}_j^- - \boldsymbol{\mu}^-)^T$$

$$\boldsymbol{S}_B = (\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-)(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-)^T \tag{2.9}$$

where $\boldsymbol{\mu}^+$ and $\boldsymbol{\mu}^-$ are the mean vector estimated from the corresponding class $+$ and $-$. In LDA, we want to find a vector $\boldsymbol{w}$ which maximizes

$$\frac{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}} \tag{2.10}$$

When $\boldsymbol{S}_W$ is full rank, the vector $\boldsymbol{w}$ which maximizes Equation (2.10) is the generalized eigenvector corresponding to the largest generalized eigenvalue of the problem $\boldsymbol{S}_B\boldsymbol{w} = \boldsymbol{S}_W\boldsymbol{w}\lambda$. In the case of a full rank $\boldsymbol{S}_W$ the solution can be determined as the eigenvector of the Fisher matrix $\boldsymbol{F} = \boldsymbol{S_W}^{-1}\boldsymbol{S_B}$, corresponding to the largest eigenvalue.

In the $p \gg N$ scenario, the within class scatter matrix $\boldsymbol{S}_W$ is singular, and the $p \times p$ estimated covariance matrix causes computational problems. Modifications such as Diagonal LDA and Regularized Discriminant Analysis are presented in [37, 36] to overcome such issues.

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) [38] is an ancient story. While with the development of computing capacity, it becomes one of the most recognized classification techniques by the practitioners. Due to computational reasons, there are many ways of implementing the K-Nearest Neighbors algorithm. More detailed information can be found in [39, 40]. The general idea is that the decision of the unknown label depends on the majority vote of the K-nearest neighbor points. As shown in Figure 2.5, the unknown sample is predicted as a sample from Class 2 (triangles) by the voting rule from the label of $K = 5$ nearest neighbors.

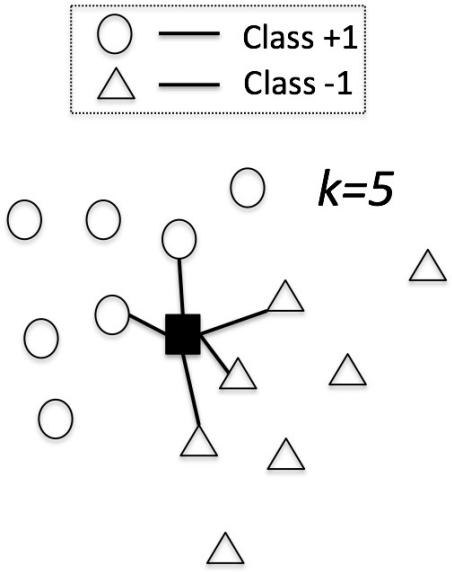


Figure 2.5: The label of the square data is unknown. By looking at the labels of $k = 5$ nearest neighbors, the unknown sample is predicted as a Class 2 (triangle).

Decisions of KNN rely on a localized scheme. However, for $p \gg N$, the nonparametric (implicit) estimation of the density function often results in high generalization

errors. Regularized version of KNN has been proposed in some literature [41] and better results are achieved. However, the computational complexity is still an issue in such cases.

Likelihood Ratio Test (LRT)

This technique [42, 43] has an even longer history but being continuously exploited over the decades. It is widely used in signal detection applications.

Suppose we have two hypothesis H_0 and H_1. Assume that the parameters of the two distributions are known. Figure 2.6, for instance, shows two different Gaussian distributed data sets sitting around known mean $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ with covariance matrix $\sigma^2\boldsymbol{I}$. The two hypothesis are therefore described as:

$$\begin{aligned} H_0: &\quad \boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\mu}_0, \sigma^2\boldsymbol{I}) \\ H_1: &\quad \boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{\mu}_1, \sigma^2\boldsymbol{I}) \end{aligned}$$

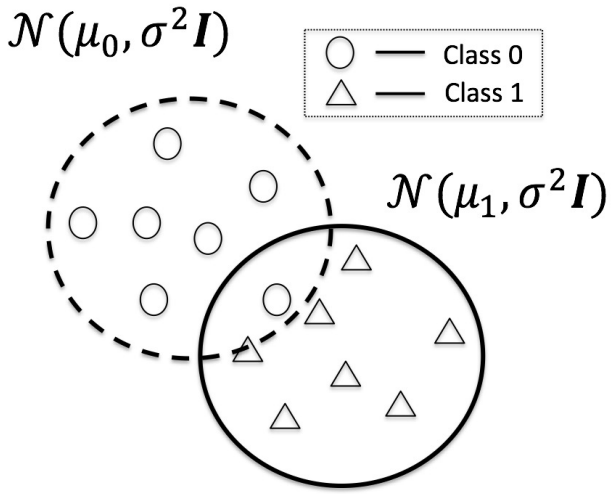


Figure 2.6: An illustration of LRT in Gaussian settings. The two sets of the data both follow Gaussian distribution with known means and covariance.

The log Likelihood Ratio Test of a testing sample $\boldsymbol{x}$ is then written as:

$$\log L(\boldsymbol{x}) = \log \frac{p(\boldsymbol{x}|\boldsymbol{\mu}_0, \sigma^2\boldsymbol{I})}{p(\boldsymbol{x}|\boldsymbol{\mu}_1, \sigma^2\boldsymbol{I})} \tag{2.11}$$

We can set a threshold γ. If $\log L(\boldsymbol{x}) > \gamma$, $\boldsymbol{x} \in H_0$, otherwise $\boldsymbol{x} \in H_1$. The level of γ will determine the P_D and P_{FA}. This test is optimal in the sense that it maximizes the P_D for a given level of P_{FA}.

However, the assumption of known data distribution does not hold in general. Given a set of training data, the parameters need to be estimated, which is extremely problematic for $p \gg N$ case. Furthermore, since we are directly dealing with probability density functions, the problem of tractability arises. Even under the Gaussian assumption, we have the simplest scenario [44], yet a large covariance matrix to deal with, for $p \gg N$.

Matched Subspace Detectors (MSD) and Linear Regression Classifiers (LRC)

Subspace models are frequently used for high dimensional signal vectors, where each data point $\boldsymbol{x}_c^i$ drawn from class c ($c \in \{1, 2\}$) is assumed to be generated from the following generating function:

$$\begin{aligned} \boldsymbol{x}_1^i &= \boldsymbol{U}_1 \boldsymbol{\alpha}_1^i + \boldsymbol{e} \\ \boldsymbol{x}_2^i &= \boldsymbol{U}_2 \boldsymbol{\alpha}_2^i + \boldsymbol{e} \end{aligned} \tag{2.12}$$

where the columns of $\boldsymbol{U}_c$, denoted as $\{\boldsymbol{u}_{c,l}\}$ represent the orthonormal basis of the corresponding linear subspace with $l \in \{1, \cdots, D_c\}$; $\boldsymbol{\alpha}_c^i$ is the weighting vector; and $\boldsymbol{e}$ is random noise.

With a given $\boldsymbol{U}_c$, we can compute the distance $d_c(\boldsymbol{x}^i)$ from $\boldsymbol{x}^i$ to the linear subspace spanned by its orthonormal columns $\{\boldsymbol{u}_{c,l}\}$:

$$d_c(\boldsymbol{x}^i) = \left\| \boldsymbol{x^i} - \boldsymbol{P}_c \boldsymbol{x}^i \right\|_2 = \left\| \boldsymbol{x^i} - \boldsymbol{U}_c \boldsymbol{U}_c^H \boldsymbol{x}^i \right\|_2 \tag{2.13}$$

where $\boldsymbol{P}_c$ denotes the projection matrix and $\boldsymbol{U}_c^H$ is the Hermitian transpose of the matrix $\boldsymbol{U}_c$.

Given one unlabeled signal $\boldsymbol{x}^i$, the class label $\hat{c}^i$ is estimated according to the following criterion:

$$\hat{c}^i = \arg \min_c d_c(\boldsymbol{x}^i) \tag{2.14}$$

In practice, the classifier associated with the model assumption given in Equation (2.12) and criterion in Equation (2.14) can be constructed by defining the projection matrix $\boldsymbol{P}$ as:

$$\boldsymbol{P}_c = \boldsymbol{X}_c (\boldsymbol{X}_c^T \boldsymbol{X}_c)^{-1} \boldsymbol{X}_c^T \tag{2.15}$$

where, data matrix $\boldsymbol{X}_c$ is constructed by placing all training data from class c as its columns:

$$\boldsymbol{X}_c = \left[\boldsymbol{x}_c^1,\ \boldsymbol{x}_c^2,\ \cdots,\ \boldsymbol{x}_c^{N_c} \right] \tag{2.16}$$

In the literature, under the model assumption introduced in Equation (2.12), a signal detection technique called Matched Subspace Detector (MSD) [45, 46, 47, 48, 49, 50, 51, 52] has been proposed and analyzed based on the derivation of Generalized Likelihood Ratio Test (GLRT) [44]. Different scenarios are taken into consideration, such

as known/unknown basis $\boldsymbol{U}_c$, covariance matrix, etc. In the machine learning community, a family of data driven classification techniques called Linear Regression Classifier (LRC) [53] have been presented for face recognition in 2010. Modifications and extensions of LRC are developed accordingly, such as Principal Component Regression Classifier (PCRC), Improved Principal Component Regression Classifier (IPCRC) [54], Robust LRC [55], Ridge Regression Classifier (RRC) [56], Unitary Regression Classifiers (URC) [57], etc. These existing techniques generalize the idea of LRC. For example, PCRC and IPCRC tend to improve the performance by manipulating the principal components in the PCA space; RRC is developed to handle degenerated cases using ridge regression and URC tries to minimize the total within class projection error. These extensions mainly fall into two categories: 1) those minimizing within-class-projection-error using different regularization, and 2) those improving discrimination ability by manipulating principal components. However, these approaches do not take between-class-scattering and within-class-errors into account simultaneously. Furthermore, computational complexity for training is yet another issue.

Chapter 3

Contribution of Included Papers

There are three papers included in this thesis. In Paper 1, a generalized subspace model is introduced to overcome the issue arising from the mixture of subspaces. Given a testing sample, the set of active training data for each class is learned automatically. In Paper 2, Linear Discriminant Analysis (LDA) is extended to a multi-dimensional result for dimensionality reduction. Combined with a classification technique called Ridge-SVM, the scheme outperforms many classical techniques. Paper 3 presents a subspace learning technique generalized from Matched Subspace Detector (MSD) attempting to maximize the class separability. The main contribution is summarized in the diagram shown in Figure 3.1 and is also discussed in the following sessions.

3.1 The Generalized Subspace Model and Training Data Selection

In Section 2.2.2, we introduced subspace classification techniques based on the model assumed in Equation (2.12). However, in some scenarios, there are a number of subspaces involved in the classification problem but we group them together into a smaller number of "super classes" due to the fact that labels are only given for the "super classes". Therefore, each "super class" contains the union of several subspaces models which need to be identified.

Based on this observation, we develop a generalized subspace model in Paper 1. Instead of a linear subspace spanned by $\boldsymbol{U}_c$, each $\boldsymbol{x}_c^i$ is considered to be generated from one out of a set of linear subspaces spanned by the 'smaller' basis $\boldsymbol{U}_c^k$, where $k \in \{1, \cdots, K_c\}$, and K_c is the total number of such subspaces. By 'smaller' basis, one can imagine that the subspace spanned by the basis appeared in Equation (2.12) is now a set of K_c linear subspaces spanned by some low dimensional bases.

Let $\mathscr{U}_c = \{\boldsymbol{x}^i : \boldsymbol{x}^i \in class\ c\}$. We assume:

$$\mathscr{U}_c = \bigcup_{k \in \{1 \ldots K_c\}} \mathscr{U}_c^k \tag{3.1}$$

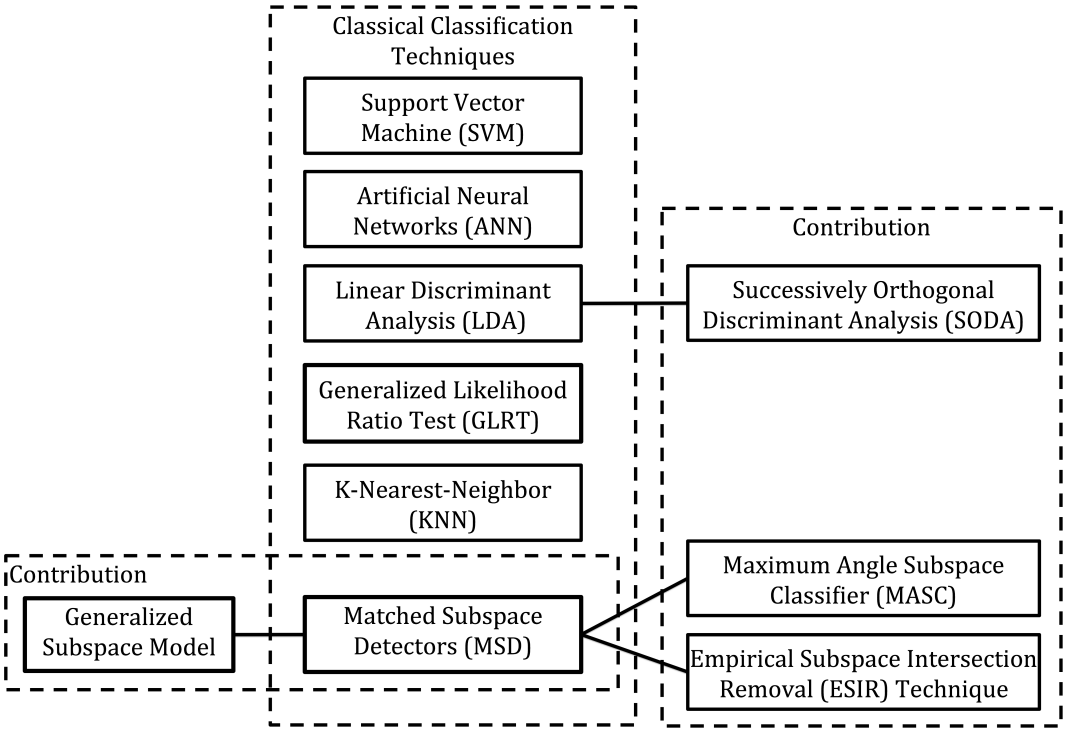


Figure 3.1: The structure of the relationship between presented techniques.

where $\mathscr{U}_c^k$ is a subset of $\mathscr{U}_c$ which is a linear subspace defined as:

$$\mathscr{U}_c^k = \left\{ \boldsymbol{x}^i : \quad \boldsymbol{x}^i = \sum_{l=1}^{D_c^k} \beta_l \boldsymbol{u}_{c,l}^k \right\} \tag{3.2}$$

where D_c^k is the dimension of the subspace spanned by $\boldsymbol{U}_c^k = \left\{ \boldsymbol{u}_{c,l}^k \right\}$ with $l \in \{1, \cdots, D_c^k\}$, and β_l is the corresponding coefficient. Hence, we can say that each class is defined as a union of K_c sub-spaces where each subspace is represented by basis matrix $\boldsymbol{U}_c^k$ with orthonormal columns.

Accordingly, the signal model becomes:

$$\begin{aligned} \boldsymbol{x}_1^i &= \boldsymbol{U}_1^{k_1(i)} \boldsymbol{\beta}_1^i + \boldsymbol{e} \\ \boldsymbol{x}_2^i &= \boldsymbol{U}_2^{k_2(i)} \boldsymbol{\beta}_2^i + \boldsymbol{e} \end{aligned} \tag{3.3}$$

where, the weighting vector $\boldsymbol{\beta}_c^i$ depends on the sample number i, the class label c, as well as the basis number $k_c(i)$. Note that the purpose of writing $k_c(i)$ is to show that k_c is a function of the sample index i.

Designing a classifier along the lines of the one in Equation (2.14) we need to determine the distance of a test sample x^i to both classes. Without ambiguity, we write k instead of $k_c(i)$ for convenience.

Therefore, $d_c^k(\boldsymbol{x}^i)$ is indicating the distance from $\boldsymbol{x}^i$ to the subspace $\boldsymbol{U}_c^k$:

$$d_c^k(\boldsymbol{x}^i) = \left\|\boldsymbol{x}^{\boldsymbol{i}} - \boldsymbol{U}_c^k(\boldsymbol{U}_c^k)^H \boldsymbol{x}^i\right\|_2 \tag{3.4}$$

And $\hat{c}$ can be estimated in the same way as in Equation (2.14) with a slight modification.

$$\hat{c}^i = \arg \min_c \min_k \; d_c^k(\boldsymbol{x}^i) \tag{3.5}$$

Therefore, in order to make the regression on the unknown sub-basis, we need to select the subset of the training data for $\boldsymbol{x}^i$ from each class. In this paper, we applied both L1 & LS regression for the training data selection as a nearest neighbor test in a subspace fashion.

3.2 Successively Orthogonal Discriminant Analysis (SODA)

In Linear Discriminant Analysis (LDA) introduced in Section 2.2.2, we are seeking for a linear transformation mapping the data vector onto $\mathbb{R}$ where a decision can be made. The idea behind this is that the transformed subspace provides the maximum class separability on the training data according to Fisher's criteria.

So can we do better? That is, instead of a one dimensional decision boundary, we would like to end up in a higher dimensional vector space, where the same criteria can be met. In Paper 2, a technique called Successively Orthogonal Discriminant Analysis (SODA) is developed, which is formulated as follows:

SODA formulation. *The matrix* $\boldsymbol{W} = \left[\boldsymbol{w}_1 \cdots \boldsymbol{w}_k\right]$ *defines a map* $\boldsymbol{x} \rightarrow \boldsymbol{x}'$*, whose columns satisfy:*

$$\begin{aligned} \underset{\boldsymbol{w}_i}{\textit{maximize}} \quad & \frac{\boldsymbol{w}_i^T \boldsymbol{S}_B \boldsymbol{w}_i}{\boldsymbol{w}_i^T \boldsymbol{S}_W \boldsymbol{w}_i} \\ \textit{subject to} \quad & \boldsymbol{w}_i \perp \boldsymbol{w}_{1,\ldots,i-1} \\ & \boldsymbol{w}_i^T \boldsymbol{w}_i = 1 \\ & \boldsymbol{w}_i \in \textit{Span}(\boldsymbol{S}_W) \end{aligned} \tag{3.6}$$

where $\textit{Span}(\boldsymbol{S}_W)$ *denotes the range space of matrix* $\boldsymbol{S}_W$. □

It has been proven in the paper that the problem can be solved by the following procedure presented in Algorithm SODA.

Algorithm SODA

- Let $\boldsymbol{S}_W^{(0)} = \boldsymbol{S}_W$
  $\boldsymbol{F}^{(1)} = (\boldsymbol{S}_W^{(0)})^+ \boldsymbol{S}_B$
- For $i = 1 : k$
  - Solve for $\boldsymbol{F}^{(i)} \boldsymbol{w}_i = \lambda_i \boldsymbol{w}_i$
    where λ_i is the only non-zero eigenvalue of $\boldsymbol{F}^{(i)}$.
  - Let $\boldsymbol{D}^{(i)} = \boldsymbol{I}_{m \times m} - \boldsymbol{w}_i \boldsymbol{w}_i^T$ be the deflation matrix
    $\boldsymbol{S}_W^{(i)} = \boldsymbol{D}^{(i)} \boldsymbol{S}_W^{(i-1)} \boldsymbol{D}^{(i)}$
    $\boldsymbol{F}^{(i+1)} = (\boldsymbol{S}_W^{(i)})^+ \boldsymbol{S}_B$
- Form matrix: $\boldsymbol{W} = \left[\boldsymbol{w}_1 \cdots \boldsymbol{w}_k\right]$
- Transformation of the features: $\boldsymbol{x}' = \boldsymbol{W}^T \boldsymbol{x}$

3.3 Maximum Angle Subspace Classifier (MASC) and Empirical Subspace Intersection Removal (ESIR)

Let us revisit Equation (2.12) introduced in Section 2.2.2:

$$\begin{aligned} \boldsymbol{x}_1^i &= \boldsymbol{U}_1 \boldsymbol{\alpha}_1^i + \boldsymbol{e} \\ \boldsymbol{x}_2^i &= \boldsymbol{U}_2 \boldsymbol{\alpha}_2^i + \boldsymbol{e} \end{aligned} \tag{3.7}$$

where $\boldsymbol{U}_c$, $c \in \{1, 2\}$, are the subspace spanned by the data from class c. In practice, $\boldsymbol{U}_c$ is unknown to the user. Most existing techniques emphasize on the quality of the estimated $\boldsymbol{U}_c$. However, for classification purposes, the between class contrast is equivalently important. In Paper 3, a between class distance metric is defined and maximized under the constraint of the within class regression error. However, the computational complexity is relatively high, which motivates us to develop a fast algorithm as an approximation called Empirical Subspace Intersection Removal (ESIR) technique.

3.4 Discussions and future work

We have so far introduced some classical techniques designed for the HDLSS issue. There are remaining challenges and the main purpose of this thesis work is to develop feasible solutions to resolve these difficulties and improve the classification performance. The applications play the role of real world validation and assigning meanings to rather abstract approaches. As a future work, we will continue our development on such techniques to tackle specific difficulties for classification problems. Moreover, studies on properties and limitations of proposed algorithms are under progress by both empirical testing and theoretical analysis.

References

[1] A. Gelman and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*, 1st ed. Cambridge University Press, 2006.

[2] W. Mendenhall and S. Terry, *A Second Course in Statistics: Regression Analysis*, 7th ed. Pearson, 2011.

[3] R. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[4] I. Guyon, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[5] U. Libal, "Feature Selection for Pattern Recognition by LASSO and Thresholding Methods - A Comparison," *Proceeding of Methods and Models in Automation and Robotics (MMAR), 16th International Conference on*, vol. 3, pp. 168–173, 2011.

[6] J. M. Landsberg, *Tensors: Geometry and Applications*, 1st ed. American Mathematical Society, 2011.

[7] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*, 1st ed. Springer, 2007.

[8] A. Y. Ng and M. Jordan, "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes," *Neural Information Processing Systems (NIPS)*, 2001.

[9] A. Fhager, T. McKelvey, and M. Persson, "Stroke Detection Using a Broad Band Microwave Antenna System," *4th European Conference on Antennas and Propagation (EuCAP)*, 2010.

[10] M. Persson, T. McKelvey, L. H. Fhager, A., and etc, "Advances in Neuro Diagnostic based on Microwave Technology, Transcranial Magnetic Stimulation and EEG Source Localization," *Asia Pacific Microwave Conference*, 2011.

[11] I. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, 2002.

[12] J. R. Sarunas and A. K. Jain, "Small Sample Size Effects in Statistical Pattern Recgonition: Recommendations for Practitioners," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 252–264, 1991.

[13] P. Hall, J. S. Marron, and A. Neeman, "Geometric Representation of High Dimension, Low Sample Size Data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 3, pp. 427–444, 2005.

[14] D. C. Hoyle, "Automatic PCA Dimension Selection for High Dimensional Data and Small Sample Sizes," *Journal of Machine Learning Research*, vol. 9, pp. 2733–2759, 2008.

[15] F. Jianqing, F. Yingying, and W. Yichao, "High-dimensional Classification," in *High-dimensional Data Analysis*. World Scientific, 2010, ch. 1, pp. 3–37.

[16] R. Vershynin, "How Close is the Sample Covariance Matrix to the Actual Covariance Matrix?" *Journal of Theoretical Probability*, vol. 25, no. 3, pp. 655–686, 2011.

[17] J. Cohen, "Things I have Learned (so far)," *American Psychologist*, vol. 45, no. 12, 1990.

[18] S. T. Smith, "Covariance, Subspace, and Instrinsic Cramer-Rao Bounds," *IEEE Transaction on Signal Processing*, vol. 53, no. 5, pp. 1610–1630, May 2005.

[19] A. Y. Ng, "Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance," *21st International Conference of Machine Learning (ICML)*, 2004.

[20] V. N. Vapnik, *Statistical Learning Theory*, 1st ed. Wiley-Interscience, 1998.

[21] B. E. Boser, I. M. Guyon, and V. Vladimir, "A Training Algorithm for Optimal Margin Classifiers," *Proceedings of the fifth annual workshop on Computational learning theory (COLT)*, 1992.

[22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes - the art of scientific computing*, 3rd ed. Cambridge University Press.

[23] K. I. Diamantaras and K. Margarita, "Binary Classification by Minimizing the Mean Squared Slack," *Proceeding of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2057–2060, 2012.

[24] T. Hofmann, B. Scholkopf, and A. J. Smola, "Kernel Methods in Machine Learning," *Annals of Statistics*, vol. 33, pp. 1171–1220, 2008.

[25] B. Schlkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* , 1st ed. The MIT Press, 2001.

[26] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed. Cambridge University Press, 2000.

[27] A. Nachman, "Theory of Reproducing Kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[28] D. A. Robinson, "Implications of Neural Networks for How We Think About Brain Function," *Behavioral and Brain Sciences*, vol. 15, no. 04, pp. 644–655, 1992.

[29] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1st ed. Oxford University Press, 1995.

[30] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall, 1998.

[31] M. Gregoire, O. Genevieve, and M. Klaus-Robert, *Neural Networks: Tricks of the Trade*. Springer Verlag, 1998, vol. 1524.

[32] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall, 2008.

[33] B. D. Ripley, *Pattern Recognition and Neural Networks*, 1st ed. Cambridge University Press, 2008.

[34] S. I. Gallant, "Perceptron-Based Learning Algorithm," *IEEE Transaction on Neural Networks*, vol. 01, no. 02, pp. 179–191, 1990.

[35] G. J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Wiley Interscience, 2004.

[36] D. M. Witten and R. Tibshirani, "Penalized Classification Using Fisher's Linear Discriminant," *Journal of the Royal Statistical Society: Series B*, vol. 73, no. 5, pp. 753 – 772, 2011.

[37] T. Hastie, R. Tibshirani, and F. Jerome, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.*, 2nd ed. Springer, 2009.

[38] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. 13, no. 01, pp. 21–27, 1967.

[39] G. Shakhnarovish, T. Darrell, and P. Indyk, "Nearest-Neighbor Methods in Learning and Vision," *IEEE Transaction on Neural Networks*, vol. 19, no. 2, 2008.

[40] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society, 1990.

[41] W. Wu and D. L. Massart, "Regularised Nearest Neighbour Classification Method for Pattern Recognition of Near Infrared Spectra," *Analytica Chimica Acta*, vol. 349, no. 1, pp. 253–261, 1997.

[42] S. S. Wilks, "The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses," *The Annals of Mathematical Statistics*, vol. 09, no. 01, pp. 60–62, 1938.

[43] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*, 1st ed. Prentice-Hall, 1998.

[44] B. Shearer, O. Zeitouni, J. Ziv, and N. Merhav, "When is the Generalized Likelihood Ratio Test Optimal?" vol. 38, no. 2, pp. 2–7, 1992.

[45] F. B. Scharf Louis L., "Matched Subspace Detectors," *IEEE Transactions on Signal Processing (TSP)*, vol. 42, no. 8, pp. 2146–2157, 1994.

[46] M. L. T. Kraut Shawn, Scharf Louis L., "Adaptive Subspace Detectors," *IEEE Transactions on Signal Processing (TSP)*, vol. 49, no. 1, pp. 1–16, 2001.

[47] Y. Jin and B. Friedlander, "A CFAR Adaptive Subspace Detector for Second-Order Gaussian Signals," *IEEE Transaction on Signal Processing*, vol. 53, no. 3, pp. 871–884, 2005.

[48] B. Olivier and L. L. Scharf, "CFAR Matched Direction Detector," *IEEE Transactions on Signal Processing (TSP)*, vol. 54, no. 7, pp. 2840–2844, 2006.

[49] H. Kwon, S. Member, and N. M. Nasrabadi, "Kernel Matched Subspace Detectors for Hyperspectral Target Detection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 178–194, 2006.

[50] A. Salberg, A. Hanssen, and L. L. Scharf, "Robust Multidimensional Matched Subspace Classifiers Based on Weighted Least-Squares," *IEEE Transaction on Signal Processing*, vol. 55, no. 3, pp. 873–880, 2007.

[51] N. Asendorf and R. R. Nadakuditi, "Improving and Characterizing the Performance of Stochastic Matched Subspace Detectors When Using Noisy Estimated Subspaces," *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, vol. 1, no. 1, pp. 1893–1897, Nov. 2011.

[52] ——, "The Performance of a Matched Subspace Detector that Uses Subspaces Estimated from Finite, Noisy, Training Data," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 1972–1985, 2013.

[53] I. Naseem, R. Togneri, and M. Bennamoun, "Linear Regression for Face Recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.

[54] S. Huang and J. Yang, "Improved Principal Component Regression for Face Recognition under Illumination Variations," *IEEE Signal Processing Letter*, vol. 19, no. 04, pp. 179–182, 2012.

[55] I. Naseem, R. Togneri, and M. Bennamoun, "Robust Regression for Face Recognition," *Pattern Recognition*, vol. 45, no. 01, pp. 104–118, 2012.

[56] D. Arpit, S. Wu, P. Natarajan, R. Prasad, and P. Natarajan, "Ridge Regression based classifiers for large scale class imbalanced datasets," *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pp. 267 – 274, 2013.

[57] S. M. Huang and J. F. Yang, "Unitary Regression Classification With Total Minimum Projection Error for Face Recognition," *IEEE Signal Processing Letter*, vol. 20, no. 05, pp. 443 – 446, May 2013.