THE UNIVERSITY OF
BUCKINGHAM

Measuring Confidence in Classification Decisions for Clinical Decision Support Systems:

A Gaussian Bayes Optimization Approach

Dongxu Han

School of Computing

University of Buckingham

Jan 2022

Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Computing

Declaration

I, the author of this thesis, hereby declare that the thesis entitled “Measuring Confidence in Classification Decisions for Clinical Decision Support System: A Gaussian Bayes Optimization Approach” submitted herein is the result of my own work and includes nothing which is the outcome of work done in collaboration with others except as declared and specified in the text. I further declare that no substantial parts of the thesis have already been submitted, or are concurrently submitted for any such degree, diploma or other qualifications at the University of Buckingham or any other universities or institutions.

Signature: **Date: 17th/Jan/2022**

Acknowledgement

First of all, I would like to thank you for reading this thesis. It took me a great effort in presenting this work in front of you, thank you for finding it. I am glad that we are sharing the same research interest and I wish my work can help you in achieving better success.

During my PhD study, I was full of ambition at the beginning and have been dreaming a lot about what to mention in this section. However, at this time when I have eventually finalized my study, most of the things do not matter that much anymore. Studying in PhD is difficult and lonely, very much like walking slowly in a desert and looking for an unknown oasis. Here, I would like to especially thank my two supervisors *Mr. Hongbo Du* and *Prof. Sabah Jassim*. They are like the torches in the night, that guided me in direction and helped me to walk forward in this dark desert firmly without any hesitation. I would like to thank them for supporting me and providing me with the greatest freedom in conducting my research.

I would also like to thank *Dr. Alaa Al-Zoubi* and *Prof. Dhiya Al-Jumeily* for their valuable suggestion provided for correcting this thesis. They have not only made this thesis more readable and scientific, but also inspire me in future research directions.

The other person I would like to thank is my high school physics teacher *Mr. Zhipeng Wu*. He is the person who inspired my interest in science. I would like to thank him for teaching me thinking critically and conducting research in scientific ways. The habit learned under his guidance has set a solid foundation for me and contributed to all my following research in higher degrees.

The thank also goes to my parents, *Mrs. Wendong Liu* and *Mr. Qiang Han*. They have provided me with a very good learning environment. I can still remember that my mother had once taught me about how rain is formed by putting a frozen ladle over boiling water so I can observe raindrops coagulated on the cold surface. My father’s job has also made it very convenient for me to familiarise myself with computers and other electronic devices at a very young age. They have inspired my curiosity about this world and trained me on pursuing the truth persistently. They never mocked me for my ignorance; instead, they always encourage me to discover new knowledge. What I have achieved today is inseparable from their long-standing support.

In the end, I would also like to thank every researcher that I have cited in this thesis. It is their research on the countless little mighty things that eventually cumulated and contributed to my success. As I have mentioned at the beginning of this section, I would also wish my little success can contribute to greater success for the others. Little by little, step by step, I believe this is exactly how modern science eventually achieved incredible things and made our life better. I cannot foresee the future of our life, but I am almost certain that it must be amazing and exciting. To the brighter future of humankind, salute!

Dedication

I would like to dedicate this thesis to my grandmother *Mrs. Huiqin Hu* and grandfather *Mr. Ying Han*, who passed away during the period of my PhD research. Unfortunately, I couldn’t accompany them at the end of their life due to studying abroad. They were born in a difficult era and have been working very hard in taking care of the whole family since then. I am pleased that they have finally rested in their long-deserved peace. I wish they would be proud of my achievement.

Abstract

This thesis generally investigated various aspects of designing and developing Clinical Decision Support Systems (CDSSs), but in particular exploited machine learning techniques in supporting medical diagnosis decisions. Having reviewed the fundamental functional components of existing modern CDSSs, it shows that most such systems were lacking a trusted decision evaluation module that provides reliable information about decision strengths. Therefore a refined CDSS system framework was first proposed, which centralises the concept of confidence-based classification by coupling eventual decision outcomes with a level of decision reliability. Based on measure theory, a unified *Decision Score* measure of the decision reliability was introduced, which combines the decision outcomes in terms of positive or negative signs together with the decision strength in percentage values.

Furthermore, the behaviour of the proposed decision score measure was investigated in more complex and diverse feature spaces of high dimensionality, where the challenges of the "curse of dimensionality" are encountered. Such challenge was handled by revisiting the problem under orthogonal projections of the feature space, and have developed a new measure in performing quantified evaluations on the decision score measure, known as the *Decision Sensitivity* measure. The key influencing factors for the sensitivity of decisions were found to include not only the dimensionality of the selected features, but also the standard deviation of each feature used in the transformed orthogonal space.

After the basic concept of the decision score measure is established, this thesis further extended the uses of the decision score measure in a multiple classifiers setting. This thesis first reviewed the principles and rationales behind various well-established information fusion schemes and tested their strengths and limitations in adapting the proposed decision score measure. Moreover, a correlation-based decision fusion scheme was proposed in maximising the potentials of the decision score measure in complex scenarios. Based on the evaluation results across different datasets, it proves that fusion schemes improve the robustness of the decision models while maintaining a good level of diagnostic accuracy in general.

As clinical decision making normally faces new unseen cases and unpredictable challenges, it is essential to maintain a degree of adaptivity in a CDSS for post-deployment robustness of the system. Therefore, the last piece of the research reported in this thesis focused on investigating possible ways to refine the CDSS decision scores model in a time-efficient manner, spontaneously. In particular, this thesis reviewed several commonly used metrics and methods for monitoring and refining prediction models, and further adapted these methods to the proposed decision score measure.

Table of Contents

Chapter 1. Introduction

1.1. Research Background and Problem Description

Healthcare is an essential service of modern societies. Effective healthcare is critical for improving life expectancy and life quality in both developed and developing countries. However, according to the newest published data from the Office of National Statistics, despite having a 2% drop in the death rate in the UK from 2018 to 2019 (Owen-Williams & Cornish, 2020), the year 2018 has hit the highest deaths registered in the history since 1999 (Patel, 2019). Among all mortalities accounted for recent years, after excluding the influence from the recent outbreak of the COVID-19 pandemic, cancer was counted as the most common cause of death worldwide (Sung, et al., 2021). A large proportion of cancer fatalities can be avoided or at least reduced with timely and effective treatments (Nardin, et al., 2020). Sadly, any unexpected pressure added to the healthcare system, such as the recent outbreak of the COVID-19 pandemic, causes delays to the treatment and leads to more death inevitably (Maringe, et al., 2020). It is for this reason that researchers throughout the world are constantly searching for more timely and easy to implement diagnostic techniques with robust accuracy. More timely and precise diagnosis will not only lead to more effective patient treatment, but also better prevention of the illness from deterioration, better forms of patient management, higher rates of patient survival, and better utilisation of resources.

The effectiveness of the diagnosis very much depends on the accurate identification of distinctive characteristics of the disease. Besides conventional medical tests such as blood, urine tests and X-ray examinations, medical checks using digital imagery devices have become commonplace in clinical practices. In the last two decades, technologies such as ultrasound, MRI, CT scan have advanced significantly, and these modern imagery devices provide information-rich images of different modalities that have greatly assisted the work of clinicians. The use of such images has growingly become a necessity need for most, if not all, of medical decision making. However, most of the time, radiographers, radiologists, and specialists manually examine the images, measure the target objects of interest within the images by applying machine functions, and record the readings automatically generated by the devices. The effective use of the images and their manually obtained image features very much depends on the knowledge, experience, and skills of the image examiner. It is well known that the supply of knowledgeable and skilled medical staff (including the image examiners) is always limited due to many years of costive training before members of such staff become qualified and start practising. According to the most recent clinical workforce report published by NHS Improvement in the UK in February 2016 (NHS Improvement, 2016), the shortage of qualified clinical staff has been reported as a great risk factor to public health services. Many medical-related occupations including medical radiographers appear constantly on the best-needed occupations shortlist issued by the UK government (UK Visas and Immigration, 2021).

To meet such constant and high demands, computer scientists have been trying to develop computer-based software systems to support clinical decision making such as diagnosis, and ease the pressure from the demand by speeding up the manual diagnostic process, and/or introducing a certain level of automation in processing image characteristics and results of other medical tests. In contrast to human domain experts, computer-based systems have many advantages in powerful and speedy data processing capabilities. Such strengths are particularly useful for a big data environment, which complements the conventional exhaustive information filtering process by clinicians. Along with the advances of machine learning technology in artificial intelligence, many computer-based approaches have been proposed to assist clinical diagnosis in the past two decades (see details in Chapter 2). These proposed methods and systems offer great potentials in assisting clinical diagnoses. However, the current state-of-art computerised solutions still largely stay within the realm of research purpose only. The ultimate effective solution lies in the integration of multiple predictive models augmented with knowledge and experiences from domain experts. Clinical Decision Support Systems (CDSSs) are set to achieve this goal.

Clinical Decision Support Systems (CDSSs) refer to computer-based systems that support decision making in clinics. Many types of CDSSs serving different use cases have been proposed in the past (Yang , et al., 2009; Tabesh, et al., 2007; Srivastava, et al., 2008; Kothari, et al., 2012; Basavanhally, et al., 2010). In recent years, machine learning techniques are increasingly deployed in medical diagnosis with promising results. A modern CDSS normally contains classification models as its core, but such models have constraints that are worth further investigation. Three important issues have not been fully addressed in the existing literature, or not fully studied together despite their intertwined natures. The first issue is concerned with the outcome of a typical diagnostic classification model, which is normally trained on various features extracted from medical images and observations based on medical tests using known discrete class labels. A class label can either be a categorical value such as benign or malignant (Yang , et al., 2009; Tabesh, et al., 2007) or a description of magnitude gradings (Tabesh, et al., 2007; Basavanhally, et al., 2010). In either case, the outcome predicted by a trained classification model is mainly the appropriate class label. However, in real-life medical applications (and perhaps in other applications as well), predicting the correct class label alone is often insufficient without any further support in terms of the level of adequacy in classification when there are vital risk implications of misclassification. For example, in a scenario of diagnosing the right type of ovarian tumour, it is important for doctors to know whether the tumour is benign or malignant and at the same time how much belief they have in that diagnostic decision. In a CDSS, this issue of measuring the level of belief becomes even more important given the potential ethical concerns regarding decisions made by machines. Unfortunately, most existing work in classification is mainly interested in the accuracy of the predicted class label rather than the reliability and strength of individual predictions. It must be noted that the decision strength is not the same as decision accuracy. The tested accuracy or the predicted confidence interval of a model can only provide a rough idea of the model's general reliability in distinguishing the predefined classes; it does not give any indication about the level of certainty of a specific decision. Therefore, evaluating the prediction decision strength with a score is within the realm of this research. How such a decision score should be defined and how such a score should be used in the input data feature space have become research questions of the interest.

The second issue is concerned with combining decisions made by multiple classification models. This issue is raised first because of a common practice in medical diagnosis. When an acute disease with exceptional characteristics occurs to a patient, it is often the case that multiple medical domain experts are consulted. Very often it is the consensus decision made by a panel of experts that is taken as the final decision. Due to various levels of knowledge and experience, such joint decision-making can be very complex without any existing rules to follow. In machine learning, although fusion of classification is a well-ploughed field of study, most existing fusion schemes are again mainly interested in final class labels and the joint decision accuracy (Ren, et al., 2016). The strengths of combined decisions by various classifiers are still non-trivial and remain an open question. Besides, practically deployed CDSSs in the foreseeable future are very likely to adopt an approach to combine machine recommended decisions and human expert decisions due to potential concerns on ethical grounds. Given these scenarios, the fusion of decision strengths should also be properly studied. In other words, the final decision is again not only just the most appropriate class label, but also a "combined" strength score for that decision.

The third issue relates to deployment beyond the training and testing stage. In medical decision making, it is desirable to continuously improve the decision quality and reliability. A medical domain expert does not become an expert only by training. More likely, an expert learns much more from his experience of making correct and incorrect decisions in his practice of medicine. Similarly, in machine learning, continual learning is the major school of thought regarding learning beyond the model development phase (Doorhof, 2018). However, the effect of learning beyond the development phase upon decision strength and reliability needs further investigation and understanding. Under certain circumstances, such learning may result in a marginal adjustment of decision strength, but under other circumstances, models may need to be retrained. The appropriate actions to take need further investigations.

All three issues are essential and critical for determining whether a CDSS is truly ready to assist medical doctors in making the right decisions.

1.2. Research Aim and Objectives

The overall aim of this research is to investigate the theory and application of a confidence-based classification decision-making scheme that underpins a typical CDSS, addressing all three issues raised. The term "*confidence*" here specifically refers to the level of belief in a classification decision made about a class label, which should not be confused with the concept of the confidence interval of a modelled result. In particular, the thesis is intended to achieve the following research objectives:

- To explore and define a sensible measure of confidence in a classification decision that reflects the level of strength of the decision for a specific class outcome, and to represent the level of confidence in a classification decision through a meaningful and valid decision score that combines the level of confidence and class outcome in a single quantity,
- To explore properties of decision score in low and high dimensional feature spaces regarding the level of sensitivity with a classification decision, i.e. how much and in what way the decision score was affected by the dimensionality of the feature used,
- To investigate existing fusion schemes and propose a new correlation-based fusion scheme that sensibly combines strengths of decision making from multiple classifiers into a final decision score,
- And to investigate and propose a method for continuously maintaining and refining decision scores in the deployment phase to reduce the need for model retraining.

The background setting of this thesis is the decision-making aspect of a typical CDSS concerning the three essential issues raised in Section 1.1. The context of this research is outlined in three intertwined and closely relevant areas regarding classification decisions, i.e. decision evaluation in assessing the decision strength and reliability of each CDSS decision model during the decision-making process, decision optimisation in enhancing the overall performance of a single or multiple CDSS decision models, and decision evolution in enabling and improving the adaptiveness of CDSS to unforeseen failures through continuous adjustment of trained models. Figure 1.1. illustrates a Venn diagram of the three areas. The aim and objectives of this research are set against the intersection (not the union) of the three connected areas. The common core is the concept of decision confidence. At the end of this research, the investigation from each topic area and the understandings gained are expected to consolidate each other and finally contribute to a sophisticated decision-making scheme behind CDSS.

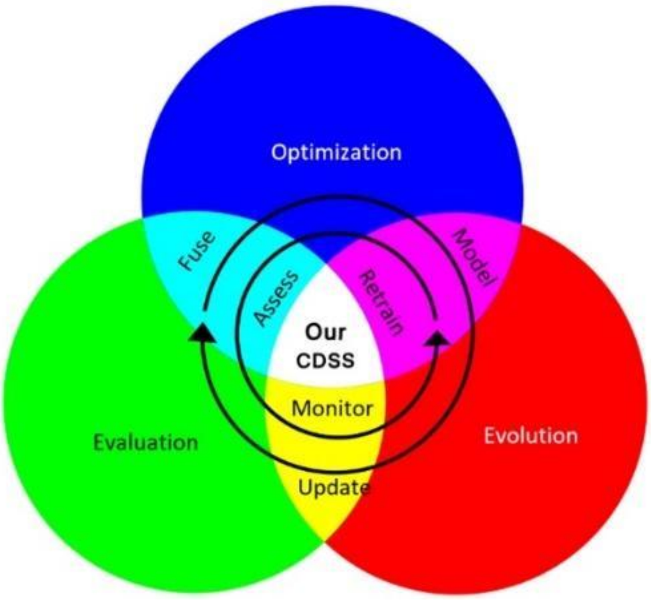


Figure 1.1: Context of This Research in a CDSS System Environment

As shown in Figure 1.1, On the one hand, the evaluation results regarding individual features and classifiers contribute towards a better understanding of decision-making, and such understanding provides a solid base towards better optimisation of classification models and possible fusions of the models. The improved individual and fused models can then be continually tested, assessed, and monitored for their performance during the deployment stage which in turn may update the evaluation of decision strength according to the online performance. On the other hand, the continuous evaluation of the decision performance can be seen as cues for continual learnings, which can then be used for determining the appropriate occasion for model updates and reconstructions to ensure the robustness of the CDSS decision-making. Retraining of the models can be triggered if a significant decay in the models' performance is observed. The duplex cycles among these three main areas of interest ensure that the CDSS adapts changes actively in the dynamic online environment, leading to a gradually maturing and improving CDSS.

By achieving the objectives listed above, this research is intended to answer the following research questions:

- What is it meant by decision confidence? How is the level of confidence measured into a decision score?

- What is the basic model for measuring decision confidence? How does the level of decision confidence change in low and high dimensional feature spaces? How is decision sensitivity controlled in low and high dimensional feature spaces?
- What is the best way of fusing the decision confidence levels when multiple classifiers are making a joint decision?
- How is the confidence measure adapted to a different type of data in the CDSS?
- How to make decision models in the CDSS adapting to data of high velocity?

1.3. Research Methodology and Approach Taken

This research follows the route of literature informed investigation. The main research is based on mathematical reasoning and statistical modelling approaches. Although the nature of the research is application-oriented, aiming to solve practically encountered issues in real-life CDSS, this research is not entirely intended as pure data-driven and experiment-based research. Data sets are indeed used to verify, support and constrain the mathematical models and schemes developed from sound theories. As indicated by the title of the thesis, Gaussian models and Bayesian classifiers play a central role in this research due to their soundness and directness in expressing the key concepts and reasoning schemes within this research.

The data sets to be used for evaluation and verification purposes are well-chosen to ensure that (a) the data sets are collected from real-life clinical settings in order to reflect the applicability of the research outcomes, (b) the data sets must be of different varieties to reflect the clinical reality, and (c) the data sets should be of different dimensionalities from very low dimensions to very high dimensions to test and validate the scope of the applicability of the derived theoretical models. It is also important that the research recognizes the constraints and limitations on the boundaries of the theoretical models. More details on the data sets used will be given in each of the key chapters later in the thesis. The data sets are acquired either from public domains under certain terms and conditions or from sources through collaborations with permissions of use.

The research application is interested in developing a sophisticated CDSS with a closed loop in providing accountable and reliable clinical decisions. In achieving successful applications, this research is going to study several fundamental elements including but not limited to (a) methods for assessing decision making process, (b) methods for optimising the preciseness of the decision made, (c) methods for generalizing information obtained from multiple decision makers and (d) methods for renewing rules learned from the past experience.

1.4. Ethical Approval

In this study, we have used two privately owned datasets collected from our collaborated partner hospitals and one publicly available dataset acquired from the online open access database. The detailed information regarding the three datasets is described in section 2.4.2 and 5.5.1. All the patient information within the three datasets was anonymised. This research and its data use were approved by the School Research and Ethics Committee, School of Computing, University of Buckingham, UK.

1.5. Research Contributions

Novel contributions achieved from this thesis are outlined as follows:

- The thesis defines a confidence measure based on posterior probability. The confidence is modelled on the basis of the Gaussian mixture model under a Bayesian classification framework. This work, in the form of a conference paper entitled "Towards a Confidence-Centric Classification Based on Gaussian Models and Bayesian Principles", authored by Dongxu Han, Hongbo Du and Sabah Jassim, was published at the 9th York Doctoral Symposium on Computer Science and Electronics in November 2016 (Han, et al., 2016), and received the best presentation award.
- The thesis conducts a thorough investigation into the behaviours of the proposed confidence measure in a high dimensional space. It evaluates the decision sensitivity of the proposed confidence measure in both low and high dimensional feature spaces. This work was published as a journal paper entitled "Controlling Sensitivity of Gaussian Bayes Predictions based on Eigenvalue Thresholding", authored by Dongxu Han, Hongbo Du and Sabah Jassim, was published at the EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, Vol.5, Issue 16 in November 2018 (Han, et al., 2018).
- The thesis adopts several existing fusion schemes for integrating multiple confidence measures into a single confidence score. The thesis further proposes a novel correlation-based classification fusion scheme based on confidence score correlations.
- The thesis proposes a continual learning scheme in automatic refining the confidence level for adapting online classification performances of a trained classification model.

1.6. Thesis organisation

The rest of the thesis is organised as follows. Chapter 2 introduces the fundamental background knowledge for understanding the theoretical models proposed. It also provides a

CDSS context for the main issues addressed by the thesis. Unlike a conventional thesis design, the background chapter of this thesis does not include a comprehensive literature review. Instead, the review of recent existing works in the literature will be given in each key chapter throughout the thesis. By doing so, the existing works in the literature and the topic of investigation by a key chapter can be closely coupled for ease of understanding. Chapter 3 describes the idea of a sound confidence measure for a classification decision, which is further embedded into a decision score. The properties of the confidence measure in low dimensional feature space are then studied with empirical evidence to verify the soundness of the proposed confidence measure. Chapter 4 discusses the integration of the proposed decision score measure across different measurable spaces, particularly in high dimensional spaces. Two critical factors in decision-making are then investigated. Chapter 5 presents the investigation of using confidence measures in a multi-classifier situation. The chapter outlines possible approaches to adapt the existing fusion schemes to a confidence-based decision-making process, and proposes a novel fusion scheme based on the correlations between classifiers' decision scores. Chapter 6 further investigates possible ways of evolving the proposed confidence measure to suit the dynamic online testing environment with spontaneously leaning capacity, aiming to produce a much more robust and trustable classification decision scheme in the core of a CDSS.

Chapter 2. Backgrounds

This chapter is designated to provide background and context for this research. It is against this background that the three important issues regarding classification decision making are raised. The chapter is divided into three main parts. In the first part, clinical decision support systems, as a practical application platform for medical decision making of various kinds, are first reviewed. A framework of a clinical decision support system with computer-based models at the centre is proposed. After that, some general knowledge regarding supervised machine learning is summarised, and the limitations of many existing classification systems and models are highlighted. In addition, measure theory, as the main underpinning theory behind confidence measure, as well as its various models of measurement are reviewed. In the final part of this chapter, the major datasets used for conducting experiments within this thesis are introduced, with a detailed explanation of the requirements for selecting these datasets.

It is worth reiterating that this chapter is not intended as a comprehensive review of literature where the most recent developments of existing work in the topic areas of this research are surveyed. It has been decided that such a survey will be presented in each key chapter of this thesis for close coupling between the existing work and any new work developed from this research. This particular chapter only paves the way for the rest of the thesis by providing the essential knowledge and understanding for the domain of the research. Knowledgeable readers may consider skipping this chapter if needed.

2.1. Clinical Decision Support Systems

2.1.1. Overview of Concepts and Principles of DSS and CDSS

The concept of a Decision Support System (DSS) has been first introduced in the early 1960s. It refers to a computer system (software, hardware, or hybrid of the two) that is intended to bring together data, information, and knowledge from various sources, which also analyse the collected information, and facilitate the evaluation of assumptions underlying the use of specific models in assisting the organisation to make complex decisions of different kinds (Sauter, 1997). In other words, DSS aims at assisting human decision-makers through efficient and effective uses of a large volume of data of various kinds. DSS systems should be distinguished from many other kinds of enterprise information systems ranging from human-based information processing and information management systems like a database management systems (Shobowale, 2020) to fully automated decision-making systems like autonomous vehicles (Shi, et al., 2021). As shown in Figure 2.1, Sauter (Sauter, 1997) illustrates the position of a typical DSS in contrast with the others in a spectrum of various kinds of information systems. At the left end of the spectrum are the machine-oriented systems which normally provide summary information extracted from well-structured data sets. Such systems are not meant to support decision making; decisions are still made by human users based on the provided information summary as evidence. At the right end of the spectrum are the human-oriented systems that are trying to mimic human decision logics and provide potential decision outcomes automatically based on a range of structured and unstructured data. A typical DSS is normally positioned between the two extremes and slightly towards the human logic side, which relies on structured data sets and may or may not produce human-like decision outcomes in the end. In other words, DSSs are intended to offer high-level and more abstract information or knowledge from structured databases that can be easily understood and used by humans to support decision making.

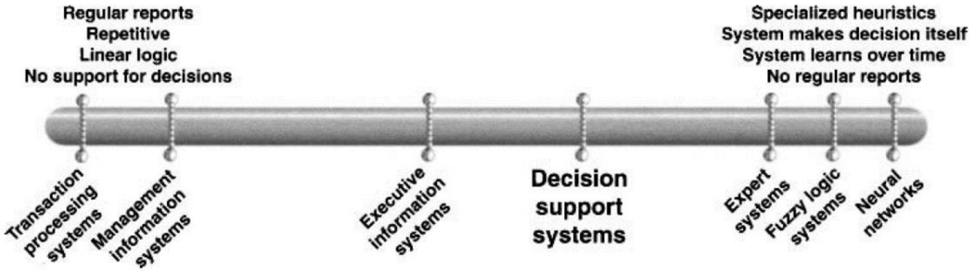


Figure 2.1 DSS in a Spectrum of Various Information Systems (Sauter, 1997)

In particular, a DSS should possess the following key characteristics (Holsapple & Whinston, 1996). First, a DSS should contain knowledge relevant to its stakeholders and intended purposes. Universal DSSs suitable for any application purposes do not truly exist. Second, a DSS should be capable of acquiring and maintaining knowledge of different kinds. Such knowledge should include not only verified information summarised from the collected raw data but also descriptive meta-data including hidden patterns discovered from the data. Third, a typical DSS, for supporting decision making, should also be capable of presenting knowledge in different and comprehensive forms according to user requests so that reasons behind decisions taken can be explained and become verifiable by human decision-makers. Last but not least, a DSS should be capable of retrieving knowledge and feature information from its database or information base for further logical deductions. It should also be capable and flexible in inducing new knowledge upon the demands of the decision-makers. Many other types of key functionalities have also been proposed (Turban , et al., 2004), but those listed earlier appear to be the most essential ones. Furthermore, different types of applications where a DSS is deployed must also require their own lists of application relevant functions.

Despite the lack of universally-recognised understanding of DSS functions, as a common consent, a DSS should provide the user with understandable information that is derived from low-level databases to enhance the human decision-making process in dynamic environments. With the recent advances of artificial intelligence, particularly machine learning, it is expected that more machine learning capabilities will be embedded into modern DSSs to handle complex and uncertain scenarios projected by large volumes of various data from multiple data sources.

Public health is always an important element for human well-being, there is a constant and urgent need to adapt DSS solutions in benefiting many stakeholders in health. Clinical Decision Support System (CDSS) is one type of DSS for health and deployed in clinics. CDSS can be generally divided into two main categories: knowledge-based CDSS and non-knowledge based CDSS (Berner, 2007). A knowledge-based CDSS aims at using computer programs to reproduce human logic and mimic the reasoning process of medical experts. The systems of this kind are most likely to be based on a set of well-defined rules that specify sophisticated and well-established clinical knowledge accumulated from the practice. IOTA Simple Rules and SRisk Calculator tool is such an example (Timmerman, et al., 2016). In the contrast, a non-knowledge based CDSS is intended to induce high-level knowledge from a large amount of practising data using machine learning techniques. Such systems may not always have an explicit and standard deductive reasoning process for decision making, instead, they rely more on the induced models to estimate outcomes of scenarios and make decisions. Depending on the areas of decision concerns, CDSSs can also be tailored and even completely developed from scratch for various purposes. Google Health's diabetic retinopathy screening solution can be considered as an attempt on such a CDSS (Gulshan, et al., 2016).

Based on intended application functionalities, CDSSs can be further classified into the following three different categories (Musen, et al., 2013). The first type of CDSS, known as the management CDSS, mainly aims for enhancing the speedy information searching capability to boost diagnosis accuracy indirectly. Such a system requires a well understood clinical knowledge database as the foundation, and then applies effective searching algorithms for relevant information and knowledge for quick reviews. There is no standard requirement on the style of presenting the knowledge retrieved, which can be either enumeration of individual cases or summarised reviews. Domain experts would conduct a clinical diagnosis by referring to the knowledge presented by the CDSS (Wagholikar, et al., 2013). Another type of CDSS, known as the focusing CDSS, is designed to supersede trained clinician for simple and repetitive works, e.g., counting the number of cells in a tissue slice, making the clinician work more effectively and efficiently on other tasks that require his/her knowledge and reasoning abilities. Such a system is commonly related to image processing, where clinical images are used as input and image summarisations and annotations are produced as output. An example of the extracted knowledge may be annotations such as the width and height ratio of a lesion area (also known as the Region of Interest) within an input image, which may indicate malignancy of the lesion. Such knowledge is then used by the doctors towards the final diagnosis of the disease (Doyle, et al., 2012). The third type of CDSSs, commonly known as the diagnostic CDSS, acts as a recommendation system that may automatically provide diagnostic decisions for human consideration. The automatic decisions can be seen as a complement to the clinical decisions made by experts and therefore increase the accuracy of the diagnosis. The system is usually built in an architecture where a central data repository forms the data kernel, feature extraction functions produce abstract feature information from the raw data, and the extracted feature data are fed into trained machine models to predict the possible outcome. Although the final knowledge provided may vary from system to system, in general, the system should produce an understandable description about the predicted circumstances of a given scenario (Srivastava, et al., 2008; Kothari, et al., 2012).

This research uses diagnostic CDSS as the background context due to the following reasons. Firstly, among all relevant decisions in medical and health applications, early and accurate diagnosis of a disease has paramount importance, particularly for cancer diagnosis. Secondly, the institution where this research is conducted has a long history of collaborating with clinical doctors within the UK and abroad with a large amount of accumulated experience which provides the convenience for the research outcomes to be verified against the real-life clinical practice. Thirdly, although topics within this research scope are also of interest for DSS in general, having a too wide scope of application in mind will lead to difficulty in conducting empirical studies to verify the soundness of any theorems. Based on those reasons, the CDSSs mentioned in this thesis will refer to cancer diagnostic CDSSs in particular.

Although accurate and precise diagnosis is always an essential requirement for a CDSS, errors do inevitably occur during testing and deployment influencing the performance of the CDSS in practice. As a solution, many types of optimisation techniques have been deployed to reduce errors and their associated risks. Generally, such optimisations can take place in three different stages of a CDSS operation. Pre-phase optimization refers to improving the quality of data before they are fed into the CDSS. The goal is to diminish potential errors contained in the raw inputs and therefore to make the CDSS yield a more accurate prediction. Examples of these techniques can be Welch's t-Test for feature selection (Song, et al., 2007), Gabor Filter for image enhancement (Hong, et al., 1998) and Principal Component Analysis for Dimension reduction (Wold, 1987), etc. However, the effect of these techniques varies due to the complex nature of the error involved, which cause the optimisation being unreliable in many circumstances. Therefore, we would like to investigate a more robust optimisation flow to deal with different types of errors dynamically.

In-phase optimization focuses on using different decision-making mechanisms to reduce overall chances of getting decision errors. Examples can be using Bayesian Networks (Boutell & Luo, 2004), random forest (Wang, et al., 2020) or XGBoost (Ogunleye & Wang, 2020) for classification accuracy enhancement. However, combining (also called fusing) the knowledge between different decision models can be challenging due to a large number of possible combinations and their various natures. A set of possible features F and a set of possible classification algorithms C would produce $F \times C$ possible classifiers. However, not all of them are appreciated since the nature of certain features may not be appropriate to all the classifiers. As a result, it would be unreliable and also costly if the final decision outcome is summarised among all the possible classifiers in real-time. Therefore, feasible solutions need to be proposed for an efficient and effective approach to obtain reliable results while minimising the unnecessary and redundant classification efforts, making the diagnosis in real-time more feasible, robust, and reliable.

Post-phase optimization aims at adding additional bias to the final decision outcome to reduce any negative eventuality. The optimization algorithms often follow a regression approach in transferring the derived in-phase decision into the expected decision, in which the parameters are set once trained and hardly changeable, but still are heavily relied on. For example, a rigorous threshold on evaluating the reliability of the decision outcome can be used to reduce the false classification of miscarriages and hence the unnecessary negative impact of odinopoeia on women. Determining and tuning these parameters optimally are always critical and challenging to the system performance. So far, this aspect of the system has not been properly studied, and further research can lead to more effective schemes for post-phase optimization.

2.1.2. A Proposed Framework for Hybrid Machine Learning and CDSS

In order to introduce the main topics concerning this research, a proposed framework for a hybrid machine learning and expert knowledge-based clinical decision support system is first given here. The main components of the proposed system are based on a general understanding of existing CDSSs with a specific emphasis on diagnostic decisions. Figure 2.2 shows a block diagram of the framework for such a system. At the centre of the system is the **central data repository** that stores various data details including patient demographics such as age and gender, descriptive data such as medical history, medical tests conducted such as blood tests, heart tests, various modality images like X-ray and ultrasound, extracted features from the images like histograms of local binary patterns, geometric features, fractal dimensions, etc. The repository pulls the data from different medical centres’ data sources and integrates them together into complete data objects and instances held in their entirety. The schema for the data repository must be open in nature so that newly found data details, as well as newly identified feature vectors, can be added at any time with ease. An example schema is the NoSQL key-value scheme (Sivasubramanian, 2012). The management software maintaining the data repository should provide facilities to integrate data, enforce consistency, and upon request pull a complete data set in supporting training and testing of machine learning models.

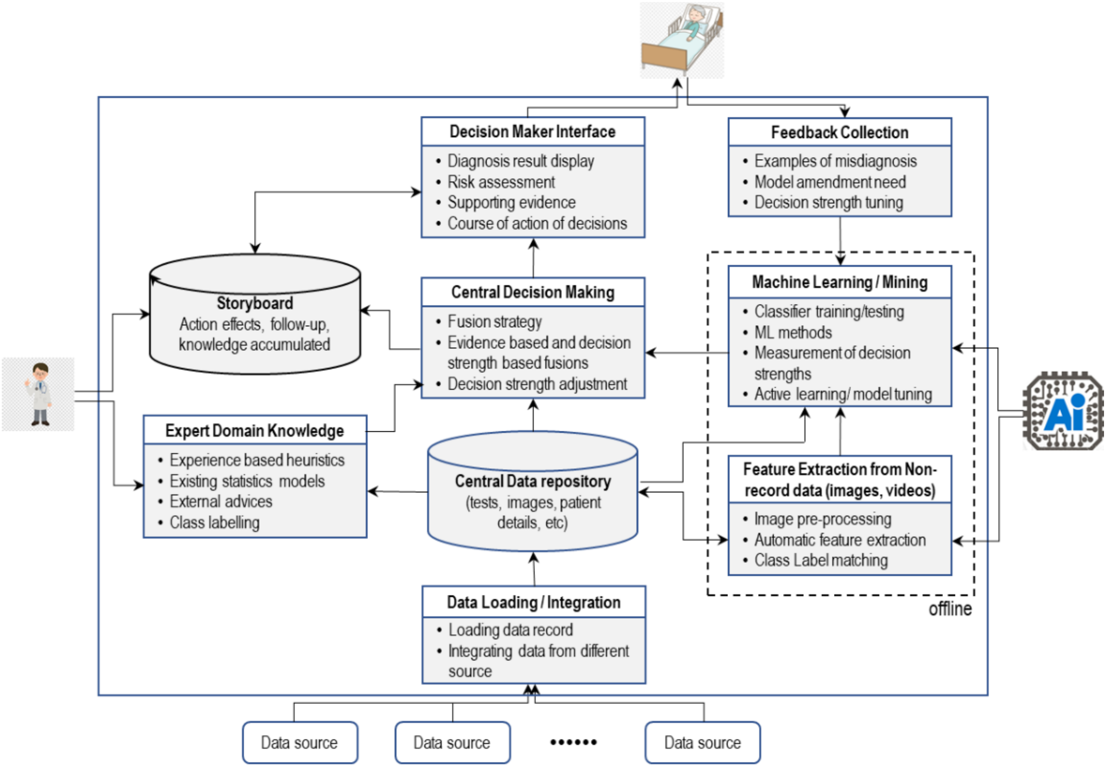

Decision Maker Interface
• Diagnosis result display
• Risk assessment
• Supporting evidence
• Course of action of decisions
Feedback Collection
• Examples of misdiagnosis
• Model amendment need
• Decision strength tuning
Storyboard
Action effects, follow-up, knowledge accumulated
Central Decision Making
• Fusion strategy
• Evidence based and decision strength based fusions
• Decision strength adjustment
Machine Learning / Mining
• Classifier training/testing
• ML methods
• Measurement of decision strengths
• Active learning/ model tuning
Expert Domain Knowledge
• Experience based heuristics
• Existing statistics models
• External advices
• Class labelling
Central Data repository
(tests, images, patient details, etc)
Feature Extraction from Non-record data (images, videos)
• Image pre-processing
• Automatic feature extraction
• Class Label matching
offline
Data Loading / Integration
• Loading data record
• Integrating data from different source
Data source
Data source
••••••
Data source
Ai

Figure 2.2 Framework Architecture of a Proposed CDSS

The **machine learning (data mining)** function and associated feature extraction functions are meant to perform the development of predictive models at the back-end of the system in an "off-line" mode. In other words, any training and modelling/remodelling do not interfere with the real-time (online) use of the decision-making part of the system. The modelling functions apply machine learning and data mining techniques to descriptive and feature data to construct the initial predictive models. The models should not only produce a class outcome but also an indication of the prediction decision strength. The predictive models should be regularly updated with clinical feedback received from the clinical feedback part of the system. Therefore, the model's learning algorithms should be iterative in nature to modify the existing models in light of any prediction errors. Therefore, a continual self-learning process needs to be embedded to automatically retrain the models to adapt newly appeared data for making the diagnosis more accurate. This module can be seen as two individual parts which involve online and offline mechanisms respectively. In the offline stage, different kinds of evaluation and optimisation techniques are implemented that targeting at enhancing the features and classifiers used for making decisions, as well as amending decision model structures from a system efficiency improvement perspective. In the online stage, testing examples that are not well understood from the training database would lead to a re-training to the decision network to make the system being more adaptive to the unknown occurrences and therefore boost up the system performance.

The **central decision-making unit** is the place where the fusion of predictive decisions from all existing models takes place, combining decisions made by individual classifiers or predictive models (including non-machine learning models and even predictive models in the domain experts' minds based on their extensive experiences) into a final diagnostic decision. This unit is the core of the whole decision support system. Strategies for decision support, fusion strategies, overall assessment of prediction strength as well as any "post" adjustment of weightings for the predictive models according to their performance should be embedded into the unit. The unit may deploy a series of steps of evaluations over a period regarding the models used for decision making. The final decision together with the response action of the decision is eventually sent to the **storyboard** component. This component is built to match evaluation results from the decision-making unit with associated treatment advice which have been developed over time according to the effectiveness of the treatments in the past. This advice part is editable by the domain experts. In the future, the recordings of treatment history and the associated diagnoses can be treated as another source of potential data mining or machine learning at another higher level. Both the decision results from the central decision-making unit and the treatment advice retrieved from the storyboard are presented to the end-users through a user interface unit. This unit displays the outcomes of the decision support system in an easy to understand and interactive manner. The unit may need to pull other related information from other data sources.

2.1.3. Three Key Concerns for this Research

From the proposed CDSS framework, it is clear that three key issues of central concerns exist in the core parts of the CDSS. These are: decision strength, decision fusion, and continual learning. Decision strength should reflect the level of confidence of a diagnosis besides the class label. This is in fact a common practice in clinics. Doctors always give a diagnostic result with some level of confidence. For minor common diseases such as flue where symptoms are obvious, a diagnosis comes with a high level of confidence. However, for acute diseases such as cancer, most, if not all, diagnoses come with some degree of uncertainty. For some tricky cases, even the doctors find them difficult to decide and the prediction outcome is closer to a random guess with a low level of confidence. Having a properly defined measure of confidence that accompanies a class outcome decision is not only natural but also realistic and may even influence any further decision making such as fusion.

Decision fusion is another key issue of concern. It is about combining outcomes of individual classifiers or predictive models. Fusion normally offers opportunities for more robust outcomes of decisions. Again, decision fusion is very common in medical diagnosis. When a doctor encounters some difficulty in diagnosing disease, it is a common practice for the doctor to consult other experienced doctors to make a joint decision. In machine learning, fusion is also often used, and various fusion schemes like majority voting have been developed. However, fusion is not only concerned with *how to* combine decisions but also how to combine the levels of decision confidence from the individual classifiers. The latter is not very well researched so far.

Continuous learning is also a central issue as well in a CDSS. No predictive models can be 100% accurate all the time. In fact, doctors quite often make mistakes in diagnoses. It is from those mistakes that the doctors accumulate useful experiences and improve their skills and their art in medicine. Similarly, a CDSS must also provide the necessary mechanism for predictive models to adjust themselves. This can be done through retraining the predictive models. However, frequent retraining of the models is very time-consuming (particularly for deep learning solutions) on the one hand and may even lead to unstable models on the other. This research is intended to investigate how to adjust levels of confidence of the models instead of retraining the models, which can save a huge amount of time and maintain the stability of the models.

2.2. Machine Learning for Computer-Aided Diagnosis

2.2.1. Process Overview

An important part of a CDSS is a Computer-Aided Diagnosis (CAD) component where machine learning, as well as computer vision techniques, may be heavily involved (see the *offline* part of Figure 2.2). This is where a classification decision is made by the machine-based models in assistance for a hybrid system in the central decision-making component of the CDSS. The process behind a CAD follows a sequence of operations for different purposes. The sequential process starts with data acquisition where data of a variety of forms such as numbers, text descriptions, 2D or even 3D images or video sequences are taken through either manual, semi-automatic or even automatic means from various sources such as medical centres, clinics, hospitals and health departments. Large health systems such as the patient information system of National Health Services (NHS) in the UK (McCracken & Edwards,

2017) often pull patient-related data from different medical centres like GP surgery and hospitals and share the information among the different parts of the services. With the assistance of modern information systems and big data platforms, data from different sources can now be integrated, resulting in high dimensional data records describing various aspects of patients in their entirety. This powerful data integration offers opportunities to analyse the data in different ways for different purposes.

Once the data are collected and integrated, the data need to be processed before they are used to train predictive models. This is because data may require treatments in order to maintain desirable levels of quality in terms of accuracy, precision, completeness, consistency and a minimal amount of unnecessary redundancy. Data may also need to be prepared in a certain format before any machine learning tools and functions can be applied. Known as pre-processing, unnecessary noises are removed, missing values may be imputed, samples can be taken from a large data population for different roles and/or efficiency, and certain data are transformed from one domain to another in order to discover useful and valuable information to assist disease diagnosis.

Once the data are in a properly prepared form, it is ready to go through the mining step for hidden patterns. However, for certain types of unstructured data of large sizes such as text, sounds, images and videos, it is not desirable to use the raw data directly for mining due to extremely large data sizes and huge amounts of redundancy in data. An operation, known as feature extraction after the pre-processing step is applied to extract useful information from the raw data and represent it in a concise and well-formatted form. This step can also be seen as a transformation that converts the raw data of variable lengths into a more abstract feature vector of fixed length. For instance, a 2D greyscale image of any size, i.e., X·Y pixels, can be represented by a histogram (table of frequencies) of 256 bins. An image may go through a transformation using local binary patterns (LBP), and then a histogram vector of the resulting LBP image is obtained as a representation of the texture feature for the image. More details on image feature extraction will be given later.

The feature data, either extracted or recorded, are then used to develop a classification model that maps the descriptive features to the outcome of specific pre-defined classes. Normally, there are two phases involved in this step of the process: training (or development) and testing (or deployment). In the training phase, a set of training examples with the extracted input features together with the known class labels are fed into a supervised machine learning algorithm which then builds a model that maps the input features to the appropriate class label. Through this training process, the resulting model should perform well as far as the training examples are concerned. In the testing phase, one or a collection of testing examples with the known class labels is used to go through the same steps of pre-processing and feature extraction as training examples went. Instead of building any new model, the extracted input features of the testing examples are fed into the trained model, and the model provides a predicted class label for each test example. The predicted class labels for all testing examples can then be used to check how accurate the model's predictions are. When the model is rolled out to be used in the field (i.e., being deployed in a real setting), any examples that the model is predicting will not have known class labels, and the predicted class labels are taken as the labels for such examples. The two phases are outlined in Figure 2.3.

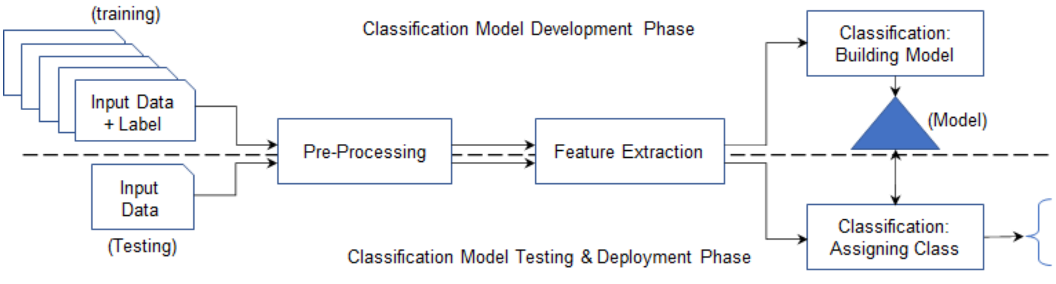


Figure 2.3: Two Phases of Developing, Testing and Deploying a Classification Model

The final step of the process is to evaluate the efficacy of the developed classification models. The evaluation result may determine whether the resulting model can be accepted for deployment or a new model needs to be redeveloped. There exist many model performance indicators among which the model accuracy is ultimately important, showing if the model is effective in the disease diagnosis or not. Accuracy is normally measured by testing the model on a set of independently sampled test examples that are well separated from those training examples used for developing the model. This is to test whether the model can be generalized to unseen testing data records. After all, the model's real use is to predict unseen data rather than re-classify an example that the model was trained from. The common practice of measuring accuracy is to measure the following main metrics:

- True positive rate (TPR): the ratio of the known positive class examples being classified correctly as positive. This ratio is also called Recall rate.
- True negative rate (TNR): the ratio of the known negative class examples being classified as negative.
- False positive rate (FPR): the ratio of the known negative examples being classified as positive.
- False negative rate (FNR): the ratio of the known positive examples being classified as negative.
- Positive precision rate (PPR): the ratio of the known positive class examples to the total examples classified as positive.
- Negative precision rate (NPR): the ratio of the known negative class examples to the

total examples classified as negative.

Note that in medical data analysis, recall rate is also known as sensitivity and TNR is also known as specificity. Of course, the intention is to develop models that have high TPR and TNR and low FPR and FNR. Other accuracy metrics such as precision, Area Under Curve (AUC), F score, etc. can be derived from the basic metrics mentioned earlier. Other performance indicators include model training and classification speed, model overfitting, model comprehensibility, model maintainability, etc. Some of these indicators will be explained when they are used later in the thesis.

Model evaluation normally follows an evaluation protocol that ensures the separation between the training and testing examples unless designated training and testing sets are already available. The simplest protocol is known as the single-split where a certain percentage of examples from the available data set is used for training and the rest for testing. Normal splits used include 2/3 training vs 1/3 testing, 80% training and 20% testing, etc. This protocol has the disadvantage that the random factor may affect the performance of the model when the testing examples have significant differences from the training ones. To reduce the random effect, random sub-sampling is an improvement by randomly repeating the single-split a certain number of times as a random experiment such as a *t*-test. The *t* statistic becomes stable when the number of trials becomes high enough. However, this protocol still has its disadvantage in that some examples are used as testing examples repeatedly and others may have never played the role of testing examples such that there is a chance that the model's test accuracy does not truly reflect the test results from all examples in the data set. That is why a more rigorous protocol, known as *k*-fold cross-validation is widely and commonly used in machine learning. The process can be explained as follows. Given a set of examples, the samples are firstly random shuffled. The set is then divided into *k* equal size partitions. Then in an iterative process, one partition is held out as the test set and the remaining partitions are used for training. Once a model is learnt from the training examples in the remaining partitions, the examples in the test partition are used to test the performance of the model. Once the iterative process is complete, *k* models are built and *k* test accuracies are recorded. The average of the test accuracies is then taken as the indicator of any models that are learnt from the data collection. In practice, *k* is often set to 5 or 10. A special case, known as leave-one-out, is the situation where $k = n$, the number of examples in the collection.

It is worthwhile to note that in medical science and research, a different testing protocol from those mentioned above, is often practised. It is often the case that clinical research is conducted in one medical centre first. Based on the single split (or hold out) principle, the collected data from the centre are split into training and testing examples. Such a test is often known as an internal test. A separate and prepared data set from the same centre can also be used to conduct another internal test. The aim of the internal test is to establish that the trained model first works well with the data from the same medical centre. After that, a separate data set from another medical centre is collected as the test set which is then used to test the model accuracy, known as an external test. The purpose of the external test is to examine the applicability of the model on data from another different medical centre. Similar to internal tests, external tests can also be conducted with data sets from different medical centres. If the model continues to perform well, it means the model is generally applicable, showing the robustness of the model and the increased scope of applicability of the model. In this research, both k-fold cross-validation and single-split protocols are used with justifications.

In the remaining part of this section, various issues at different stages of the CAD process will be raised and discussed briefly.

2.2.2. Issues in Data Acquisition

Medical data covers various aspects of patient health. The data tend to be in a variety of forms such as text, numbers, hand-written notes, diagrams, images, videos or sound signals. Many data can be found with time such as clinical histories, video clips of image frames, etc. Various challenges for data processing may exist. First, the dimensionality of data can be extremely high due to the various forms that data are represented. For a single patient, the dimensions can be in their thousands or hundreds of thousands to cover a range of medical data over time.

Another biggest issue is data quality. First, the accuracy and precision of data cannot always be assured. This is not entirely due to measurement error in data acquisition, but most of the time the fuzzy nature of certain measures and medical predictions. For instance, describing the margin of a lesion as regular or less regular can be extremely subjective despite the availability of established international guidelines such as TI-RADS (Tessler, et al., 2017). Measurement of the size of the lesion depends very much on where the radiologist or radiographer places the calibre markers on the border of the lesion. Well known as intra-observer variations, the same doctor measures the same lesion on different occasions will give different measurements. On the other hand, known as inter-observer variations, different doctors can examine the same medical image and give completely different observations and measurement results. This author was involved in a cancer sign detection work where it was found that when three doctors give the readings on margin smoothness for thyroid lesions, out of 20 ultrasound images, they only agree on 5 images, i.e., 25%. This makes machine learning difficult because some of the data are used as class labels. These labels are treated as “ground truth” when either a supervised learning algorithm tries to build a classification model, or domain-knowledge based algorithms are developed. Yet, such subjectivity in labelling means that these ground truths are not solid but soft, which leads to ill-defined models or algorithms which perform unsatisfactorily.

The precision of measurement may not be sufficient nor consistent across the recordings. Body weights may be precise to two digits after the decimal points in some measurements while others may be a whole number. Recordings from one medical centre can be taken as ordinal categories whereas those from another medical centre can be measured in scales numerically. This can be even more so for medical images. While MRI and CT scans provide high contrast and sharp images, ultrasound images are blurry by nature due to the presence of speckle noises in the images. Indeed, noise in medical data can be a serious problem. Again, taking ultrasound images as an example, noise is not only just speckle that makes the image poor contrast and not clear, but also man-made artefacts such as calibre markers, measurement results like lesion diameters, text such as patient names, black ribbon areas created by ultrasound scan actions. For certain analyses and diagnoses, there are regions of the image that offer no relevant information to the decision making. For instance, in determining if a lesion is malignant or not by examining the internal issues within a lesion, muscle structures, bones, and skin in nearby regions are not useful and hence can be considered as noise. However, these surrounding structures may have similar patterns of texture, making the detection of the true lesion region, known as the region of interest (RoI) extremely difficult.

It must be said that some medical data can be seen as very sensitive. Collecting such data may be prohibited by law or ethical codes of conduct even if they are very useful. Although not a technical issue, these issues nevertheless may prevent the collection and acquisition of useful information for disease diagnosis, investigation and effective treatment of patients.

2.2.3. Issues in Pre-processing

The main purpose of pre-processing is to prepare the acquired data into proper forms and overcome the quality issues raised in the previous section if possible. A range of operations can be performed in the pre-processing stage. Noises such as speckle noise in ultrasound images can be suppressed. As a result, the ultrasound images are enhanced without losing important information. For medical images, noise is reduced or filtered by using various filters such as median filter, adaptive median filter, wiener filter, bilateral filter and so on. Domain knowledge is often used to remove another type of noise, i.e., irrelevant artefacts like those mentioned earlier.

Region of interest may be selected or segmented. In most existing CAD systems, this task is performed through manual or semi-automatic means because accurate segmentation of RoI is still a very challenging topic of research. Automatic RoI segmentation itself may require machine learning solutions. Another operation required specifically for medical image data is image normalisation and enhancement because the images may be generated from imagery machines from different manufacturers with different device settings on intensity value ranges, frequency ranges, and zooming in/out, etc. Operations such as gamma transformation and histogram equalization can be performed for these purposes.

2.2.4. Issues in Feature Extraction and Feature Extraction Approaches

As explained before, besides readily available data in categorical and numeric values in a record form, many patient data are unstructured in nature such as text, image, diagrams, graphics, videos, and sounds. For these unstructured data, representative features are extracted from the original raw data using specially designed algorithms. In this research work, feature extractions from medical images are of particular interest. In this section, several different types of image-based features are explained, and their extraction approaches are outlined.

The first type of image-based feature is known as a morphological feature. Such features refer to signs and amounts known to domain experts and doctors. Examples of morphological features include mean sac diameter for gestational sac, area of calcification (micro or macro) inside a thyroid nodule, composition of breast lesion in terms of various echogenicity, etc. Without CAD systems, doctors normally observe the images and derive descriptions of the features based on their knowledge and training experience over the years. Sometimes, they rely on manual use of some functions to extract the features like the diameters of a lesion on different planes such as sagittal and transverse planes for a gestational sac inside a womb. Algorithms are developed to automate the measurement and extraction of such features in order to reduce the workload on doctors and avoid inter and intra-observer variations (Ibrahim, et al., 2016). Once the features are extracted, they can be used to train a classification model to make predictions of a certain disease. The advantage of this type of feature is that the features are easily explainable and understood by doctors and can be used to automate operations such as writing a clinical review report for a patient. The disadvantage is that the classification models built on such features can hardly exceed the level of diagnostic accuracy of clinical doctors because the algorithms only extract those known morphological features.

Another type of feature is more concerned with the image content. Algorithms are developed to analyse either the whole image or region of interest within the image from colour and intensity of pixels, colour frequencies, regular and repeated changes of intensity values of a collection of pixels known as texture, lines, curves and shapes. Such features may have a direct mapping to the morphology, but other features may not have such a direct mapping, but rather a certain form of pattern conveyed by the image itself. Therefore, some features of this kind might not be easy to interpret like the morphological features. In the past two decades, many extraction algorithms for image colour, texture and shape features have been developed.

The simplest is a histogram of pixel intensity values. Given a greyscale image, a histogram can be taken as a table of frequencies of pixel intensity values. The intensity value of image pixels is divided into 256 bins, and for each bin (or a specific intensity value) the number of pixels or ratio of pixels that have that specific intensity is calculated and stored. Therefore, a histogram can be seen as a vector of 256 component values. As for colour images, the histogram can be a simple concatenation of the histograms for the three colour channels, forming a feature vector of 256×3 = 768 components. Other histogram-based features can also be found. For instance, statistic moments such as means, standard deviation, entropy, etc. can be further calculated from the histogram, creating a feature vector of much lower dimensions, at the expense of losing some information details.

Another texture feature of an image is local binary patterns (LBP) (Çamlica, et al., 2015). By scanning through a 2D greyscale image, under a small sliding window of a certain size, e.g., 3·3 pixels, a central pixel is compared with its neighbouring pixels on their intensity value differences, and a sequence of binary bits is formed accordingly as a result of the comparison (if a neighbouring pixel has a higher intensity than the central pixel, 1 is assigned to that neighbouring pixel; otherwise, 0 is assigned). The sequence is also known as an LBP code. Once the scanning is complete, each pixel in the image has a corresponding LBP code. Then a histogram of all the LBP codes forms a feature vector: each component of the vector represents the frequency of the specific LBP code occurring in the image. Collectively, the LBP codes represent texture patterns such as edges, corners, curves, flat plains, etc. Among all 256 LBP codes, there is one type of code within which there are two transitions maximum between 0 and 1. These LBP codes are known as Uniform LBP (ULBP) which happen much more frequently than the other LBP codes. There are 58 possible ULBP codes, which means that a default 256-bin LBP feature vector may be replaced by a 58-bin ULBP histogram feature vector without losing too much information. An alternative compromise practice is to include an extra bin in the histogram feature vector for ULBP codes to represent all non-uniform LBP codes, creating a 59-bin histogram feature vector.

LBP is only one of many types of texture features. Another widely used one is the grey level co-occurrence matrix (GLCM) (Nguyen, et al., 2021). Based on certain patterns of intensity value differences between two pixels along a specified direction, a GLCM captures the frequencies for such patterns to occur in the image. Rather than using the matrix directly as a feature vector by flattening, some statistic summary over the frequencies (similar to statistic moments) is obtained from a GLCM and used as the feature vector. GLCM feature vector represents the frequencies of patterns such as lines, edges and their orientations inside an image. Depending on the number of angles involved and the summary moments included, a GLCM feature vector can be of high or even extreme high dimensions.

2.2.5. Issues over Classifiers Used

Classification is a well-studied area of machine learning. Many methods of learning a classification model from a set of training examples have been developed (Verma, et al., 2017; Désir, et al., 2012; Ren, 2012). According to the representation of the model, the existing methods can be categorised into main approaches. The *k nearest neighbour* (kNN) methods produce a model in the form of a memory space of selected training examples as templates. Using a suitable similarity function, the classifier calculates the degree of similarity from the unseen record to each of the templates and finds the *k* closest neighbours of the unseen record. By exercising voting or scoring policy (such as majority voting), the class of the unseen record can be determined collectively by the neighbours together. The effectiveness of the classifier is determined by the similarity measure, the representativeness of the selected templates and the scoring or voting scheme adopted. Another approach for classification is to construct a *decision tree* where internal nodes are the tests on attribute values and leaf nodes are the class labels. Making a decision using the classifier is a process of traversing the tree from the root towards a leaf. During the traversal, a sequence of tests is conducted on the values of the attributes (or feature variables). If an attribute takes a specific value or a range of values, the branch concerned in the tree is followed. Decision trees are best suited for categorical variables although working also for the continuous variables. A version of the decision tree induction, known as *Random Forest*, is to create a large number of trees constructed out of randomly selected training records and variables and then make a joint decision based on majority voting. Similar to decision tree induction is a category of methods known as *rule-based classifiers* that produce a sequence of IF..THEN rules, where the appropriate class labels appear in the consequent part of a rule, and testing on input variable values or ranges of values, occur in the IF part of the rule. Once the classifier tries to assign a class label to an unseen record, the variable values of the unseen record have to match the conditions of certain rules. If the condition is true, then the class label is then assigned to the unseen record. If not, the next rule will be tried until the final default rule is applied to assign a default class to the record. Other classification methods include *support vector machines* (SVM) that fit a hyperplane between examples of the known classes for separating one class from another, *artificial neural networks* (ANN) that consists of layers of artificial neurons each of which combines all input data values into a single weighted sum and transformed it into another value via an activation function over the layers, and statistical models that best capture the value distribution characteristics of one class from other.

Among the existing approaches for classification, one simple and explainable approach is the *Bayesian classifier*. Based on the Bayesian theorem, the classifier makes a classification decision by calculating the probability that an unseen record belongs to each of the pre-defined classes. This posterior probability is based on the prior probability of each class occurring in the training set and the conditional probabilities of input variables taking certain values given the known class label of them. The *Naïve Bayes* principle assumes that every input variable is independent statistically from the other variables, making the estimation of the posterior probability the result of products of prior probabilities. The simple but effective framework of classification is known for its robustness and decision making without having a large training set, a quite useful point for clinical data analysis. Another advantage is that the classification decision is probabilistic; only the class label with the highest probability is taken as the final output. This characteristic will later be exploited in this research.

2.2.6. Issues Arising in Evaluation

In Section 2.2.1, some key performance metrics were described. Aiming at a classification model that is sufficiently accurate is an ultimate requirement, specifically in CDSS when human life is at stake. However, understanding of model accuracy must be realistic. First, it must be said that the tested accuracy, no matter how high it is, does not mean the real accuracy of the model in the deployment phase. The concept of confidence interval (CI) is used to anticipate the real level of accuracy based on the central limit theorem in statistics (Khalili , et al., 2020). Second, it has been noted that many systems of classification are interested in the final class label outcomes instead of the degrees of belief in that class prediction. As outlined in the introduction chapter, clinical decisions are full of uncertainties. Doctors normally predict a likelihood of disease; a prediction of disease with only 50% confidence should be less certain and reliable than a prediction of the disease with 95% of confidence. Decisions of a class label without confidence are insufficient for risk analysis in clinical environments. Accuracy of classification and the level of confidence of the classification are two related but different issues. Confidence in classification decisions is one fundamental issue to be investigated by this research.

Accurate and robust classification models must be capable of detecting subtle differences between examples of different classes on the one hand and not to be too sensitive on the other. The models that fail to tell even big differences in feature values that will separate the different classes are called *underfit* models, which often fail to classify even obvious cases. Some models may give completely different class outcomes when there is even a small difference in the input feature values between examples of the same class. This can be often caused by the *overfitting* of the model where the model remembers too many specifics of the training examples, but fail to recognize the common characteristics between the training and testing data. Models that are overfitting tend to be very sensitive and unstable whereas models that are underfitting tend to be idle, inactive, and eventually useless. The issue of model sensitivity will also be investigated as the main topic in this research.

2.3. Measure Theory Approach

As outlined in the research aim and objectives, classification strength plays an essential role in a CDSS. Therefore, the proper calculation of such classification strength becomes critical for this research. To be shown in detail in Chapter 3 later, there exist various ways for calculating classification strengths. However, most existing methods were interested in the calculations under specific conditions and assumptions without careful and systematic reflection on the space and principles used behind the calculation. This research is intended to revisit the calculation of classification strength under a formal setting of measure theory, addressing the issue of the soundness of a classification strength measure before considering how to measure the decision strength.

As a characteristic of machine learning in general and supervised learning in particular, the population of data under study is mostly unknown. Classifiers can only be learnt from a sample or subset of the population. In mathematics, such analysis over a known subset of the population is known as the σ-algebra, which the analysis method is commonly referred to as a measure and the analysed subset is referred to as the measure space. Measure, measure space and σ-algebra lay the foundation of the Measure Theory.

A measure on a set is a systematic way to assign a number to each suitable subset of that set in relates to certain property of the subset measured. Intuitively, a measure can be considered as a generalization concept of the property of an object such as the length, area, volume and so on. Formally, let X be a set and Σ be a σ-algebra over X. A measure can then be defined as a function μ from Σ to the extended numerical space, if it satisfies the following properties:

- Null empty set property: the amount of measurement is 0 for no event, i.e.

$$\mu(\emptyset) = 0$$

- Non-negativity property: every event E in Σ is measurable with a non-negative outcome, i.e.

$$\mu(E) \geq 0$$

- Countable additivity property for all countable collections $\{E_i\}_{i=1}^{\infty}$ of pairwise disjoint sets in Σ: the measurement of a whole set is the sum of the measurements of all individual events in that set

$$\mu\left(\bigcup_{k=1}^{\infty} E_k\right) = \sum_{k=1}^{\infty} \mu(E_k)$$

In general, defining a measure can be very difficult to guarantee that all listed axioms above on all subsets are upheld. This problem is commonly resolved by defining measures on a predefined sub-collection of all subsets, where this sub-collection is also referred to as the measurable subsets. Following such a concept, if we define the pair (X, Σ_X) and (Y, Σ_Y) as two measurable spaces, the members of Σ_X and Σ_Y as the measurable sets, then a function $f: X \rightarrow Y$ is called measurable if the inverse image is X-measurable for every Y-measurable set B ∈ Σ_Y, i.e., $f^{-1}(B) \in \Sigma_X$. Under such definition, the composition of measurable functions is also measurable, making the measurable spaces and measurable functions a category, with the measurable spaces as the objects and the set of measurable functions as the identity arrows.

In the following parts of this section, we are going to review some of the commonly used measures with their possible applications in machine learning.

2.3.1. Counting Measure

Mathematically, counting measure is one of the most straightforward and simplest ways to measure a given set. Because of its simple nature, the counting measure is one of the very few measures that can be defined on any given set while still satisfying the axioms of the measure theory. However, it is mostly used on countable sets. The counting measure simply returns the number of elements in the subset if the subset has a finite number of elements; otherwise, it returns ∞ if the subset is infinite. As a formal definition, given any set E in a measurable space by taking the σ-algebra Σ of measurable subsets that consist of all subsets of E, the counting measure μ_{count} on this measurable space (E, Σ) is then defined as a positive measure $\Sigma \rightarrow [0, \infty^{+})$ as

$$\mu_{\text{count}}(A) = \begin{cases} |A| & \text{if } A \text{ is finite} \\ \infty^{+} & \text{if } A \text{ is infinite} \end{cases}$$

for all $A \in \Sigma$, where $|A|$ denotes the cardinality of set A.

In a clinical setup, counting measures can be widely applied from data preparation (e.g., counting the number of red cells in a blood test) to statistical evaluation (counting the sample size or a number of observations). More specifically, the counting measure is also commonly used in evaluating the strength of the classifier decision making. For example, the k-NN classifier utilizes the counting measure for counting the number of nearest neighbours and the decision tree classifier also adopt the counting measure for counting partitions in calculating the information gain at different levels of the tree. These measures will be further discussed in Section 3.1.2.

2.3.2. Lebesgue Measure

Simple measures such as counting measures can offer nice properties over countable sets. However, most of the sets in the real world, such as a subset of real numbers, are unfortunately uncountable, which limits the usability of the counting measures. For example, the Euclidean space is one of the most commonly used vector spaces in linear algebra, which has a finite-dimensional inner product space over the real numbers, where counting measure is not so useful when measuring such a space.

As a solution, the Lebesgue measure is one of the commonly used measures in measuring a subset in *n*-dimensional Euclidean space. The Lebesgue measure was introduced by utilizing the concept where the set of intervals within a real number space is countable. It simply assigns the length, area, and volume of Euclidean geometry to the suitable subsets within $\mathbb{R}^1, \mathbb{R}^2$ and $\mathbb{R}^3$, where the concept further extends to an *n*-dimensional volume in $\mathbb{R}^n$. However, it is worth noting that non-measurable sets do exist within the real number space when the set does not satisfy the Carathéodory criterion. An example is the Vitali sets, where the Lebesgue σ-algebra is strictly contained in the power set of $\mathbb{R}$ (Petrovai, 2019). Therefore, the Lebesgue measure is mostly studied in its outer form, referred to as the Lebesgue outer measure. The Lebesgue outer measure is defined on a domain which no longer consists of all subsets within the space $\mathbb{R}^n$. Instead, it is defined on a σ-algebra of subsets within $\mathbb{R}^n$. In generall, sets that can be assigned a Lebesgue measure are called Lebesgue-measurable, otherwise, not Lebesgue-measurable. The formal definition of the Lebesgue outer measure is presented as follow:

Given a subset $E \subseteq \mathbb{R}$, with the length of interval I = $[a,b]$ and its length function by $\ell(I) = b - a,$ the Lebesgue outer measure $\lambda^*(E)$ can then be defined as an infimum by utilising the principle of countable additivity as

$$\lambda^*(E) = \inf\left\{\sum_{k=1}^{\infty} \ell(I_k)\right\}$$

where $k \in \mathbb{N}$ and $I_k\ is\ a\ sequence\ of\ open\ intervals\ with\ E \subseteq \bigcup_{k=1}^{\infty} I_k$. To fulfil the Carathéodory criterion, the measurable set E has to satisfy that (Folland, 1999) for every $A \subseteq R$

$$\lambda^*(A) = \lambda^*(A \cap E) + \lambda^*(A \cap E^c)$$

For any set in the Lebesgue σ-algebra, its Lebesgue measure is given by its Lebesgue outer measure: $\lambda(E) = \lambda^*(E)$.

The Lebesgue measure has a wide application in machining learning, which is mostly involved in analysis within feature dimensions of different kinds. An example can be using the Lebesgue measure for measuring the distance from the position of the observation to the decision hyperplane in the SVM classifier, which is very important for understanding the bias and strength in decision making (this problem will be further discussed in Section 3.1.1 in more detail).

2.3.3. Probability Measure

As introduced so far, a measure μ can be used for measuring the designated property of a measurable set E in the defined measure space (E, Σ), where the counting measure and the Lebesgue measure are two such examples. The range of these two measures from 0 to $\mu(E)$ very much depends on the size of the measurable set defined. The problem is that $\mu(E)$ can be of any positive values within the constraint of the measure defined. The variation in the range causes the measured results incomparable across different measurable sets, which limits the usability of the measure defined. As a solution, the probability measure is commonly used for unifying the range measured across different measurable sets. The fundamental difference between a probability measure and the measures introduce previously is that the probability measure is defined in the probability space with a total measure of one, i.e., $\mu(E)$ = 1. More formally, a probability measure is a real-valued function defined on a set of events in a probability space while satisfying the essential properties of the measure theory. More specifically, the probability measure p must return 0 for the empty set (null empty set property), return results within the interval [0, 1] for the none-empty set (non-negativity property) and return results of the sum of the probabilities of each disjoint events for the union of them (countable additivity property).

As a probability measure provides a uniform way of measuring sets of different kinds, it has various cases of use. One of the common applications is used for Naïve Bayes classifiers, which was based on a slightly altered form of the conditional probability measure on the intersection of events as:

$$p(B \mid A) = \frac{p(A \cap B)}{p(A)}$$

which still satisfies the probability measure requirements as long as $\mu(A)$ is not zero (Gray, 2010) (this problem will be discussed in more detail in Section 3.1.3).

Fuzzy measures can be sometimes confused with probability measures as both of them involve likelihoods of different kinds. However, it is worth noting that not all measures that involve likelihood are probability measures. As a good example, the fuzzy measures do not count as a type of probability measure as it does not enforce a total measure of one and the countable additivity property is replaced by a set inclusion-based order relation.

2.3.4. Signed Measure: Measure with Justified Definition

Finally, it is worthy to mention a special type of measure, known as the signed measure. The most special characteristic of such a measure is that it does allow the appearance of negative values on the result measured (which disobey the property of non-negativity). The advantage of such a measure is obvious as it offers more diversity to the result representation, which in many cases contributes to clearer definitions. Nevertheless, it also raises a lot of debates such as whether the measure should include infinite values or not (Kesavan, 2019), which requires careful design in constraining the measure within the reality. In a formal definition, similar to what we have defined for the other measures previously introduced, given a measurable space (E, Σ), a signed measure $\mu^{\pm}$ can then be defined as a function as

$$\mu^{\pm}: \Sigma \to \mathbb{R} \cup \{-\infty, \infty\}$$

such that $\mu^{\pm}(\emptyset) = 0$ and $\mu^{\pm}(\bigcup_{k=1}^{\infty} E_k) = \sum_{k=1}^{\infty} \mu^{\pm}(E_k)$. That is, the signed measure produces results in the range of $[-\infty, \infty]$ or $]-\infty, \infty[$ (depends on definition) while still satisfying the property of null empty set and countable additivity. For more detailed use cases, we will be further discussed in Section 3.3.1.

2.4. Data for Supporting this Research

As outlined in the research methodology in Section 1.3, besides rigorous modelling, ideas and theorems are also to be tested using real-life data sets collected in a clinical setting throughout this thesis. This section, therefore, outlines some fundamental requirements for the data sets to be used, even in the context of issues with medical data as described in Section 2.2.2.

2.4.1. Data Requirements

The data requirements are outlined according to the main properties of the data set. First of all, the data sets should have a variety of different sizes in terms of the number of observations in each data set. Within the scope of this research, we consider medical centres of different scales and aim to have data sets of hundreds or thousands of samples in each set.

The next relevant data requirement is the dimensionality of the data set. This research is interested in both data spaces of low and high dimensions. Therefore, data sets with only a few input variables and data sets of more than one thousand input variables are targeted. To satisfy this requirement, we will consider data sets with morphological feature variables (low dimensional feature space) and the data sets with extracted texture features from medical images (high dimensional feature space).

In terms of data types of the variables, it is understood that both categorical and numeric variables are very likely to be involved in describing medical data in reality. However, in order to focus on the main topics of research and avoid going through any process of transforming categorical data into numerical ones through encoding schemes and the potential issues associated with such encoding, this research work is only concerned with numeric variables, namely, the data sets for supporting the research work are data matrices. The spread of data should be complete, covering all possible eventualities.

Data quality is always an issue of concern as outlined in Section 2.2.2. Most existing data sets are retrospectively collected and therefore it is impossible to address data quality at the point of collection. Therefore, we may have to be realistic in terms of the data used. The first point is regarding the given labelling data. The labelling data may appear in two possible ways: manually collected measurements given and class labels assigned to the data examples. We will select data sets where the labelling data are as solid as possible, the so-called golden ground truth. So, either the labels are obtained on the basis of biopsy or assigned by experienced domain experts specialized in the relevant medical domain. The second point is regarding noises and outliers. Unfortunately, since we have no control over data collection, we cannot rule out the existence of noise and outlier data objects which may create difficulty for modelling. Having said that, some pre-processing operations may be applied in order to reduce the effects of noise and outliers. The third point is about data completeness. This research assumes that the given data are complete with values for all input variables and possess the associated class labels. The fourth point is regarding data precision, the closeness to the true value. We may have to make the assumption that the given data are precisely measured and correctly recorded because this research does not involve collecting data directly from participating patients.

Data granularity can be of different levels. Data details can be the same as being given, and aggregations of data of various forms can be seen as summarization and further abstraction of the given data. Data transformed from one domain to another domain for feature extraction purposes are also possible and permitted.

2.4.2. Selected Data Sets

With the requirements listed above in mind, we carefully selected two data sets. The first data set is of low dimensionality, obtained from the Early Pregnancy Department, NHS Queen Charlotte's and Chelsea Hospital, Imperial College London. The data set is concerned with manual measurements of gestational sac sizes taken by gynaecologists from 2D ultrasound images of the womb for detecting signs of miscarriages in the first trimester of early pregnancy. The data set was provided in two separate batches. The first batch has 94 measurement records together with the known class labels: 15 records of miscarriage (MC) and 79 records of Pregnancy of Unknown Viability (PUV), a phrase used for pregnancies not considered as miscarriage declared at the point of the ultrasound scan even miscarriage may be declared in subsequent scans. The second batch contains 90 measurement records: 11 cases of MC and 79 cases of PUV. For both batches, three manual measurements of gestational sac sizes, i.e., major and minor diameters of the sac taken from the sagittal plane and major diameter of the sac taken from the transverse plane. The diameters were recorded by the ultrasound machine after the gynaecologist placed calibre markers. The whole data collection also includes a derived variable, known as the Mean Sac Diameter (MSD), calculated as the average of the three diameters of the sac. This data set is chosen because of its simple morphological features and relatively low dimensionality. A clear medical understanding of the features also justifies using the data set.

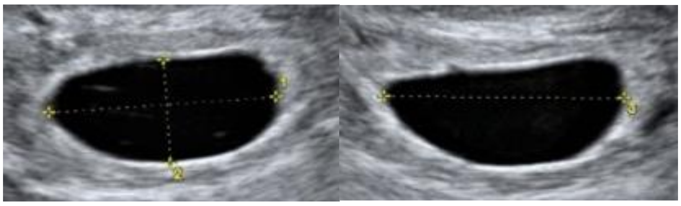

(a) (b)

Figure 2.4: Gestational Sac in 2D Ultrasound from the Early Pregnancy Dataset
(a) sagittal plane (b) transverse plane; major diameters are measured by yellow axis

Another dataset chosen for the research is the CBIS-DDSM dataset (Lee, et al., 2017) obtained from the public domain. The dataset contains 2,620 mammography images of breast lesions of two kinds: *mass* and *calcium* from the results of pathology reports. The relevant regions of interest, i.e., area of a breast lesion, are specified and verified by domain experts. From the region of interest mammography images, image-based textures will be extracted, which will provide flexibility and freedom to explore decision confidence measures in a high dimensional feature space.

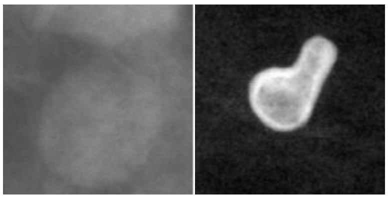

(a) (b)

Figure 2.5: ROI Images in Mammogram from the CBIS-DDSM Dataset (a) mass (b) calcium

Chapter 3. Measuring Classification Confidence

After providing the context for this research, this chapter is intended as the first key chapter to address the core issue of classification confidence. Informally, classification confidence refers broadly to the strength and degree of certainty of a classification decision. It is an essential requirement for clinical diagnostic decision-making. Although there have been various forms of expressing the concept, this chapter provides a formal definition of the concept.

The chapter is organised as follows. We will first review existing methods in the literature for measuring classification strengths. Then the chapter will present a unified and formal definition of the concept of classification confidence based on the principle of measure theory. Following the definition, the chapter will propose a confidence measure in the probability space based on the principles of Gaussian distribution and Bayesian classifiers. The proposed confidence measure will then be evaluated with a clean clinical dataset of low dimensionality. Comparisons between different settings and modelling techniques will then be presented and discussed at the end.

3.1. A Review on Existing Measures of Certainty on Classification Outcomes

In recent years, evaluation of classification outcomes has increasingly drawn attention in order to satisfy the clinical requirements due to management of risk and obligation to explain. Conventional methods for evaluating classification outcomes focus primarily on measuring the overall accuracy of the classification model in terms of predicted class labels. The degree of certainty for each specific classification decision is often used as a complement of the decision rather than the essential focus of the decision-making. Several different measures for evaluating the certainty of a classification outcome have been proposed in the past, but most of them heavily depend on the nature of the classifier used. In general, the existing methods can be categorized into hypothesis-based, information-based or predictor-based measures, each of which will be explained in more detail in the following subsections.

3.1.1. Hypothesis-based Measures

Hypothesis testing is a well-established method in statistics for scientific research, which is primarily interested in the statistical significance regarding the difference between the means of two data samples. In principle, any classification problem can also be seen as a statistic problem, which intends to find an approximation of an unknown observation drawn from the population with the maximised amount of likelihood to a set of training examples provided. In other words, statistical models can be built for samples of each training class to mimic their real nature in the whole population. Any example drawn from the population is tested against these models to obtain the statistical significance and be accordingly classified. The statistical significance can be determined by using different hypothesis testing methods. For example, calculating the confidence interval p of a training class sample regarding an unknown test feature value x by using one-sample location Z-test for a specific class ω_i can be written as:

$$p_{\omega_i} = \int_{-\infty}^{\frac{x-\mu_{\omega_i}}{\sigma_{\omega_i}}} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \, dx \tag{3.1}$$

where μ is the sample mean, σ is the sample standard deviation. We assume the test statistics are approximately normally distributed according to the central limit theorem (Filmus, 2010). Based on this assumption, the cumulative distribution function is sometimes simplified into a measurement based on a scale of Z-score (Khazendar, et al., 2014):

$$\hat{p}_{\omega_i} = \left| \frac{x - \mu_{\omega_i}}{\sigma_{\omega_i}} \right| \tag{3.2}$$

where the reliability of the classification result is no longer measured by a numerical confidence interval but by a numeric distance based on a scale of standard deviation that indicates how much the observation is away from the standard mean. In other words, the reliability is inversely proportional to the distance between the observed value and the mean of the classified label.

A similar concept can also be applied to certain types of classifiers such as the Support Vector Machine (SVM) classifier that categorises the unknown test examples according to a decision hyperplane that best separates the training examples of different classes in the data feature space. Unlike the hypothesis testing methods previously introduced where individual models are created for each training class sample, the statistical models in SVM are combined and being replaced by a decision hyperplane. However, the same principle remains. The level of the statistical significance can be measured by calculating the *distance D* from the test example to the decision hyperplane (Li, et al., 2002) as

$$D(\vec{x}) = \frac{|f(\vec{x})|}{\sqrt{\upsilon \cdot \upsilon}} \tag{3.3}$$

where f is a function of the decision hyperplane and υ is a vector perpendicular to *f*. The hyperplane itself is the division plane separating training examples of one class (positive) from those of the other class (negative). Therefore, the distance indicates how far away the test example from the hyperplane surface and at the same time how close the test example from the sample mean vector of the training examples of one class. The smaller the distance is, the closer the test example to this “middle line” lying between the sample means of the two classes, indicating lower statistical significance since the test example is equally close to both the sample means of the two classes. On the contrary, a large distance refers to a high statistical significance since the test example is considerably close to the sample mean of one class but further away from the sample mean of another class. However, the application of such a method is very much limited due to the difficulty in defining a clear decision boundary, which is not always available for all kinds of classifiers. Khazendar et al. introduced a simplified method of categorising confidence level into low, medium or high depending on if the test example is within the half standard deviation, one standard deviation or more away from the hyperplane (Khazendar, et al., 2014). However, the boundaries between the categorical bands are determined empirically by a given dataset.

3.1.2. Information-based Measures

Information theory was proposed by Claude E. Shannon in 1948 for building reliable communication over an unreliable channel (Shannon, 1948), where "information" is referring to a set of possible messages. This theory has then been extensively applied in many fields of discipline including machine learning in the later years. From the communication perspective, the health status of a patient can be seen as a target signal of interest where the signal itself is encoded with environmental noises and finally be presented as clinical observations. In this context of understanding, CDDS is playing a role of a decoder that tries to recover the original signal from the clinical observations received and then presents it as a diagnosis decision. In such a “communication”, classifiers are playing a key role in recovering the most amount of information from the received inputs and filtering out most of the irrelevancies. The training of a classifier in such a circumstance can be seen as a process of maximising the information gained over a set of known signals and related observations. The purpose of a classifier is then to categorise a group of unknown observations into *n* partitions, where *n* is the number of classes defined. The information gain I is measured by the reduction of uncertainty achieved by partitioning the original data set Ω into subsets of $\{\omega_1, \dots, \omega_n\}$, which can be expressed as

$$I(\omega_1, \dots, \omega_n) = H(\Omega) - H(\omega_1, \dots, \omega_n | X) \tag{3.4}$$

where H denotes the entropy as a measurement of uncertainty in the data set. The entropy of a discrete random variable is calculated as

$$H(\Omega) = -\sum_{i=1}^{k} p_i \log p_i \tag{3.5}$$

where p_i is referring to the natural expectation of the i^{th} category within the total *k* categories. Therefore, the entropy of a partition can then be presented as

$$H(\omega_1, \dots, \omega_n|X) = \sum_{i=1}^{k} \frac{n_i}{n} H(\omega_i|\vec{x}_i) \tag{3.6}$$

where $\frac{n_i}{n}$ is the weight of the i^{th} partitions in entire *n* data samples within *k* partitions.

Following the same concept, the reliability of a classified label can be measured by the amount of information gained at a particular feature vector after applying the classifier. A larger amount of information gained indicates a more reliable classification outcome. This principle can be applied to many classifiers. The easiest classifier among all to apply would be the k-Nearest Neighbour (kNN) classifier. The kNN classifier is relatively straightforward in design, which does not require any training in advance. The test example is classified by a majority voting among *k* nearest neighbours within the training dataset, which are identified by using distance or similarity functions in succession. In this scenario, as the classifier is using a fixed set of training examples, the entropy of the training data set remains constant and the classification result would entirely depend on the partitions among the *k* nearest neighbours. Therefore, the certainty of the classified label at a given feature vector $\vec{x}$ can be measured by using the partition entropy only as

$$I(\omega_1, \dots, \omega_n|\vec{x}) = -\sum_{i=1}^{n} \frac{n_i}{k} H(\omega_i) \tag{3.7}$$

where *n* is the total number of the class labels, $\frac{n_i}{k}$ refers to the proportion of the i^{th} class among the *k* neighbours and entropy will share a negative correlation to the certainty. The partitional entropy calculated in this case is very much dominated by the majorly class, therefore the expression above is sometimes used in a simplified linear form (Xue, et al., 2006) as

$$I(\omega_1, \dots, \omega_n|\vec{x}) \sim \frac{n_{\max}}{k} \tag{3.8}$$

where $\frac{n_{max}}{k}$ is referring to the proportion of the majority among *k* nearest neighbour.

This information-based measure has provided a very clear intention on the strength of partitioning the data. However, the calculation of the entropy is dedicated to a finite set of discrete random variables, where real-life data are commonly found within the range of real numbers. Therefore, supervised discretization techniques are commonly applied to these records such as information-based discretization where it categorises the records into subfolders with minimised partitional entropy. Although some research has adopted this technique with reasonable experiment results (Dai & Xu, 2013), it is still arguable whether this approach is valued since the discretization profoundly changes the information distribution within the data set, and therefore alters the nature of the original data.

3.1.3. Predictor-based Measures

Conformal Prediction is a theory that was first introduced in statistics and more recently brought into machine learning. It determines the level of confidence in new predictions by referring to past experiences (Shafer & Vovk, 2008). In this approach, an error probability ε is introduced when a classifier is making decisions, and then the level of confidence is considered as $1 - \varepsilon$, i.e., the probability of correct classifications. Unlike the theories introduced previously, conformal prediction allows regressions over a set of real numbers by adopting continuous probability models, which provides much broad applicability.

Modelling the error rate ε is an essential objective in the conformal prediction framework. A variety of probability models has been proposed in relation to the different types of classifiers used. This review will summarise the basic models that have been proposed in different classifiers in the literature.

kNN Classifiers

kNN classifier is a very simple classifier as mentioned in the previous section. Besides assessing classification confidence by measuring information gain, the confidence of classification by a kNN classifier can also be measured under the conformal prediction framework (Wang, et al., 2006). In this approach, $\mathrm{P}(\omega_i|\vec{x})$, as a function regarding $\{\omega_i, \vec{x}\}$, measures the probability of selecting the class ω_i from the set of the k neighbours found around the test sample $\vec{x}$. It can be measured by calculating the proportion of the majority among k nearest neighbour, i.e., $\frac{n_{max}}{k}$, under the assumption that the difference between the distance from each nearest neighbour to the test sample is very small and can be seen as identical. The error rate ε can then be measured as:

$$\varepsilon = \sum_{j=0}^{\lfloor 2^{-1}k \rfloor} \binom{k}{j} \mathrm{P}(\omega_i|\vec{x})^j [1 - \mathrm{P}(\omega_i|\vec{x})]^{k-j} \tag{3.9}$$

However, modelling the error rate ε by simply following a Bernoulli distribution is too trivial and naive. It may fail to determine the precise level of confidence. When $k > 1$, the distance from each nearest neighbour to the test sample may differ significantly from each other. The larger the value of k is, the more likely this becomes, and hence toning the probability distribution accordingly has to be considered, which very much limits the usage of such a method in practice.

Naïve Bayes Classifiers

The Naïve Bayesian probability model is a well-used conditional model that assumes the value of each feature is statistically independent. According to the Bayesian theorem (Carlin & Loui, 2000), the conditional probability $\mathrm{P}(\omega_i|\vec{x})$ of having the predicted classω_i at a particular given feature vector $\vec{x}$ can be expressed as

$$\mathrm{P}(\omega_i|\vec{x}) = \frac{\mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i)}{\mathrm{P}(\vec{x})} \tag{3.10}$$

where $\mathrm{P}(\omega_i)$ is the prior regarding the natural expectation of the classified class, $\mathrm{P}(\vec{x}|\omega_i)$ is the posterior regarding the probability of having the observed feature vector $\vec{x}$ given that the class has been classified as ω_i, $P(\vec{x})$ is the expectation that feature vector $\vec{x}$ being observed. In more detail, the naïve Bayes predictor assumes an independent relation across the *n*-dimensional feature vector $\vec{x} = [x_1, x_2, \dots, x_n]$ where $\mathrm{P}(\vec{x}|\omega_i)$ can then be simply calculated as

$$\mathrm{P}(\vec{x}|\omega_i) = \prod_{k=1}^{n} \mathrm{P}(\vec{x}_k|\omega_i) \tag{3.11}$$

The form of the probability models would depend on the nature of the data set, which can adopt various kinds of statistic models such as Bernoulli, Gaussian or Multinomial, etc. In this scenario, the conditional probability $\mathrm{P}(\omega_i|\vec{x})$ is already indicating the confidence of the prediction, where the error rate ε can then be simply determined as

$$\varepsilon = 1 - \mathrm{P}(\omega_i|\vec{x}) \tag{3.12}$$

HMM Classifiers

In contrast to the Bayes predictor introduced previously, the Hidden Markov Model (HMM) is a classic statistic model that describes the nature of a set of dependent sequential data $\{\vec{x}_1, \dots, \vec{x}_n\}$. In this model, each of the observed random variables $\vec{x}_i$ in the data set would be reliant on a hidden variable z_j that contains hidden information regarding it, i.e., $\{z_j \to \vec{x}_i\}$. In addition, each hidden variable z_j is statistically dependent on the previously hidden variable z_{j-1}, i.e., $\{z_{j-1} \to z_j\}$. This transitive relationship over multiple sequential data is also known as the Markov chain. A good example of this kind of relationship is recognising handwritings (Hu, et al., 1996), where each of the individual characters written can be considered as the sequential random observations $\{\vec{x}_1, \dots, \vec{x}_n\}$, which each of the observed feature vectors would relate to an unknown character. In addition, each of these unknown characters is expected to be somewhat related to the previous character, e.g., there is more likely to observe a vowel after the character “d” instead of consonant based on the convention of English spelling.

Following this model, by providing a set of sequential observations $\{\vec{x}_1, \dots, \vec{x}_n\}$ with their related hidden variables $\{ z_1, \dots, z_n \}$, a probability $P(\vec{x}_1, \dots, \vec{x}_n, z_1, \dots, z_n)$ regarding this prediction can then be presented as

$$\mathrm{P}(\vec{x}_1, \dots, \vec{x}_n, z_1, \dots, z_n) = \mathrm{P}(z_1)\mathrm{P}(\vec{x}_1|z_1) \sum_{i=2}^{n} \mathrm{P}(z_i|z_{i-1})\mathrm{P}(\vec{x}_i|z_i) \tag{3.13}$$

Where $P(z_1)P(\vec{x}_1|z_1)$ is the conditional probability regarding the first term, $P(z_i|z_{i-1})P(\vec{x}_i|z_i)$ is the probabilities of the following term in relation to the previous observations. Like the Bayes predictor introduced earlier, the modelling of the probability distributions can take an arbitrary form depending on the nature of the training samples, where the error rate ε can then be determined as

$$\varepsilon = 1 - P(\vec{x}_1, \dots, \vec{x}_n, z_1, \dots, z_n) \tag{3.14}$$

which only applies under the assumption that the input vectors are statistically dependent on each other.

ANN Classifiers

Artificial neural network (ANN) is a classifier that has been first proposed in the 1940s (Papadopoulos, et al., 2007), which has recently drawn more attention due to its promising performance based on the growing computational powers. Unlike the other predictor introduced previously, ANN classifies labels by summarising the knowledge outputs among multiple layers of neurons where the weights attached to the neurons are fine-tuned through a backpropagation process. The last layer of the ANN refers to as the output layer, which contains n neurons that equals the number of class labels. In the output layer, each neuron contains a real value z_{ω_i} that associates to the specific label ω_i; the label with the highest associated value will be selected as the final classification decision. Under this framework, the bias of the classified label has been already reflected by the neuron in the output layer. These biases are normally regulated by a softmax function, which transfers the real number outputted into the range of [0,1] based on all the neurons in the output layer as

$$\sigma(z_{\omega_i}|\vec{x}) = \frac{e^{z_{\omega_i}}}{\sum_{j=1}^{n} e^{z_{\omega_j}}} \tag{3.15}$$

As $\sigma(z_{\omega_i})$ already has that matches probability functions, the error rate can then be simply defined as

$$\varepsilon = 1 - \sigma(z_{\omega_i}|\vec{x}) \tag{3.16}$$

In summary, different kinds of measurement have been proposed in the past to assess the reliability of the classification result based on various types of statistical theories. Measuring the reliability of the classification result by calculating the information gain provides solid theoretical support to the evaluation. However, the constraint of discrete value has limited the usability of this method. As a complement, the hypothesis test allows a regression over a set of real numbers, but it is felt that the confidence margin is still being very crude and can be very much improved. In addition, conformal prediction provides a very nice framework that can be applied to many classifiers for a precise estimation of the reliability of the classification result. However, the probability models used for determining the error rate are most likely depending on numeric values, which seems still to be short of accurate modelling over nominal data.

Table 3.1 Rule-based Fusion Performance on Miscarriage Dataset

Classifiers	Assumption on Feature Relationship	Limitation
kNN	No assumption	High computation cost on testing
Naïve Bayes	Independent	High requirement on data characteristic
HMM	Associated	High requirement on data nature
ANN	Inter-twined	Require large data for training

More importantly, as briefed in table 3.1, all of these measures introduced have their own assumptions and limitations based on the nature of the classifier used, which lacks a common definition under a universal criterion.

3.2. Measure Theory Perspective of Classification Confidence

In general, the confidence measure defined herein is a measure of the confidence of the classification decision made upon a specific given input feature, where the classification confidence is a quantitative representation of the strength or certainty under a given environment. Let $X = \{x_1, x_2, \dots, x_n\}$ be a countable set of features that can be extracted from the observed object $\mathcal{X}$, i.e., the patient in our case. The function $\mu_C: 2^X \to \mathbb{R}^+$ regarding the score of classification confidence is a measure function because:

1. $\mu_C(\emptyset) = 0$, i.e., there is no classification confidence (zero) without providing any observed features about the object to be classified;

2. For all x in X, $\mu_C(x) \geq 0$, since every feature derived should be measurable, and should contribute towards classification outcome greater than at random;

3. For all countable collections $\{x_i\}_{i=1}^{\infty}$ of pairwise disjoint sets in X:

$$\mu_C\left(\bigcup_{k=1}^{\infty} x_k\right) = \sum_{k=1}^{\infty} \mu_C(x_k)$$

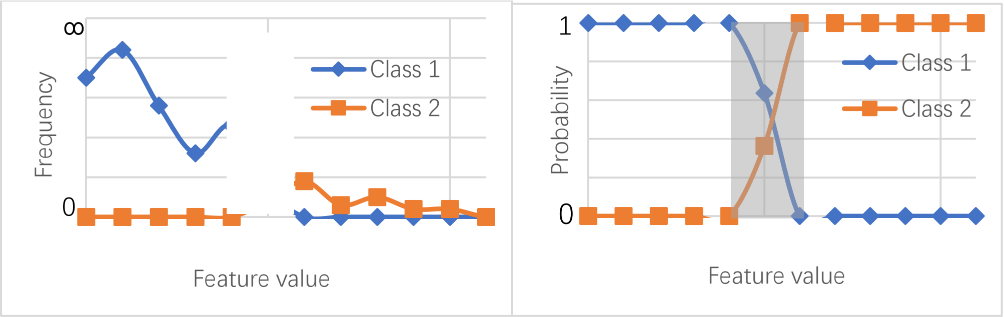
since the confidence measure of multiple features can be considered as the sum of the confidence measure on each individual of them. This definition will be further elaborated in more detail in Chapter 5.

In this context, all the measures reviewed in Section 3.1 can be reinterpreted under this universal measure space $(\mathcal{X}, X, \mu_C)$ as defined. These measures can be seen as a countable measure in universal space (such as the measure used for KNN classifier), a Lebesgue measure in Euclidean space (such as the measure used for SVM classifier) or a probability measure in probability space (such as the measure used for HMM classifier), etc. As we have already acknowledged, despite the different properties and behaviours of the confidence measures proposed in the past, they all share a fundamental principle; any kind of classification problem can be seen as a statistical problem that is trying to differentiate the training classes in the given measure space. Therefore, we would like to propose a generic confidence measure that does not rely on the type of classifier but the nature of the data distributed.

3.3. Proposed Confidence Measure

3.3.1 Measuring Classification Confidence

In a typical training data set, examples of the individual classes may be distributed differently in the corresponding feature space. Figure 3.1 presents a simplified view of distributions of a set of one-dimensional training examples of two classes and provides a conspicuous view of the strength of classification for each class. As illustrated by the frequency diagram in Figure 3.1(a), the two classes are very much distinct from each other when the data feature x has a value that is below a certain threshold x_a or above another threshold x_b due to the lack of examples from the opponent classes. However, conflicts of classification occur in a region between the two thresholds, where the feature values of the samples of two classes start to overlap. At the intersection point of the two curves, the overlapping occurs the most. Therefore, the overlapped region should be considered as a "zone of confusion" and the level of uncertainty in classifying samples should be maximised when the presences of the two classes are nearly equal.



(a) Frequency Distribution of Feature Values   (b) Probability of a Class over Feature Values

Figure 3.1 an Illustration of Value Distributions of Examples of 2 Classes

Based on this observation, it is logical to transfer the frequency-based reasoning as shown in Figure 3.1(a) into a probability-based concept as shown in Figure 3.1(b), where the likelihood of the presence of different classes is a good indication of the confusion caused in classification. As illustrated in the diagram, all the discussed characteristics regarding "confusions" are well preserved within a normalised scale. The range between the two probability curves on the y-axis indicates the magnitude of the overlapping between the two classes, which should be considered as being proportional to the level of decision confidence. Therefore, for a given finite set of classes, $\{\omega\} = \{\omega_1, \omega_2, \ldots, \omega_k\}$, the level of classification confidence μ_C can be presented as:

$$\mu_C(\omega_i|\vec{x}) \propto \left|\mathrm{P}(\omega_i|\vec{x}) - \left(1 - \mathrm{P}(\omega_i|\vec{x})\right)\right| \tag{3.17}$$

where $\mathrm{P}(\omega_i|\vec{x})$ is the conditional probability of predicting class ω_i based on a given feature vector $\vec{x}$ and therefore the aggregate probability of predicting into the rest of the classes will be $1 - \mathrm{P}(\omega_i|\vec{x})$. The second term in the absolute difference in (3.17) is indeed the classification error rate ε for class ω_i at the given data point, which can be simplified as:

$$\mu_C(\omega_i|\vec{x}) = |2\mathrm{P}(\omega_i|\vec{x}) - 1| \tag{3.18}$$

This definition is justified by an assumption that the level of the confidence of the classification is directly proportional to the difference of the two probabilities without any transition bias, i.e., the gradient is equal to 1. Formula 3.18 motivates the introduction of a generalised confidence-centric score function μ_D for the classified label ω_i as:

$$\mu_D(\omega_i|\vec{x}) = 2\mathrm{P}(\omega_i|\vec{x}) - 1 \tag{3.19}$$

In this setting, we further transform our original measure into a signed measure, in which the sign of the decision score indicates the belonging of the class. A positive value would indicate a confirmation of the chosen class ω_i and a negative value indicates a preference of the remaining classes. The absolute value of the decision score is the level of confidence in the decision made on the class belongings.

3.3.2. Modelling Decision Score

Here we introduce a Gaussian Bayes model for measuring the decision score defined in Formula 3.19. According to the Bayesian theorem:

$$\mathrm{P}(\omega_i|\vec{x}) = \frac{\mathrm{P}(\vec{x}|\omega_i)\mathrm{P}(\omega_i)}{\mathrm{P}(\vec{x})} \tag{3.20}$$

where $\mathrm{P}(\omega_i)$ and $P(\vec{x})$ are two priors that represent the natural incidence of the class ω_i and the expected observation probability of the feature $\vec{x}$, while $\mathrm{P}(\vec{x}|\omega_i)$ is known as a posterior of the feature $\vec{x}$ given that it belongs to the class ω_i. However, it is impossible to know exactly the priors in real-life scenarios due to uncertainty and randomness in the data population. We, therefore, estimate the parameters by using the training dataset.

Given a sample space $\Omega = \{[\omega_1],[\omega_2],\ldots,[\omega_k]\}$, where $[\omega_i]$ is the set of all samples that belong to the class ω_i, then $\mathrm{P}(\omega_i)$ can be estimated as the proportion of the interested class ω_i to the total number of samples, i.e.

$$\mathrm{P}(\omega_i) = \frac{|[\omega_i]|}{|\Omega|} \tag{3.21}$$

$\mathrm{P}(\vec{x})$ and $\mathrm{P}(\vec{x}|\omega_i)$ are the two probability functions describing the distribution of the feature $\vec{x}$, respectively within the overall population and the population of the class ω_i. Our proposed scheme assumes that both are Gaussian distributions. First, a simplified model based on a single Gaussian distribution is proposed as follows. Given the mean μ and variance $\sigma^2$for a univariate feature $\vec{x}$, we use the Gaussian probability density function:

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

for characterizing $\mathrm{P}(\vec{x})$ and $\mathrm{P}(\vec{x}|\omega_i)$ as:

$$\begin{cases} \mathrm{P}(\vec{x}) = \mathcal{N}(x \mid \mu_\Omega, \sigma_\Omega^2) \\ \mathrm{P}(\vec{x}|\omega_i) = \mathcal{N}(x \mid \mu_{\omega_i}, \sigma_{\omega_i}^2) \end{cases} \tag{3.22}$$

In many applications, data features normally exist in a multidimensional space. Therefore, it is essential that we expand the previous simple model into a multivariate Gaussian model to accommodate multidimensional feature vectors. For a given data set of *d* dimensions with the mean vector $\vec{\mu}$ and covariance matrix Σ, we simplify the standard Gaussian probability density function $\mathcal{N}(\vec{x} \mid \vec{\mu}, \Sigma)$ as:

$$\mathcal{N}(\vec{x} \mid \vec{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi^d|\Sigma|}} e^{-\frac{(\vec{x}-\vec{\mu})\,\Sigma^{-1}\,(\vec{x}-\vec{\mu})^T}{2}}$$

We derive $\vec{\mu}_x, \Sigma_x$ from $\Omega = \{\omega_1, \omega_2, \ldots, \omega_k\}$ and $\vec{\mu}_{\omega_i}, \Sigma_{\omega_i}$ from ω_i , and then $\mathrm{P}(x)$ and $\mathrm{P}(x|\omega_i)$ can then be characterized as:

$$\begin{cases} \mathrm{P}(\vec{x}) = \mathcal{N}(\vec{x} \mid \vec{\mu}_\Omega, \Sigma_\Omega) \\ \mathrm{P}(\vec{x}|\omega_i) = \mathcal{N}(\vec{x} \mid \vec{\mu}_{\omega_i}, \Sigma_{\omega_i}) \end{cases} \tag{3.23}$$

One concern of using a single Gaussian in the modelling is that it may not always be realistic. Real-life data may reflect a mixture of multiple Gaussians, each of which has its mean vector and covariance matrix. Therefore, we have chosen to further extend the model into a Gaussian Mixture Model (GMM). In the mixture model, each sub-Gaussian model has been given a parameter set $\theta = \{W, \vec{\mu}, \Sigma\}$, where W represents the weight of each sub-model in the mixture and the summation of the weight of all the models should be 1. Therefore, given a sequence of K parameter sets $\{\theta_{i=1\ldots K}\}$, i.e., K Gaussian sub-models, we can characterize the mixture model as:

$$\mathcal{N}(\vec{x} \mid \theta_{i=1\ldots K}) = \sum_{i=1}^{K} W_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i)$$

Therefore, we are able to derive a parameter set θ_{ω_i} for each class after the relevant class set $\{\omega_i\}$. The weight of each set is considered as its proportion in the whole training set, i.e., $W_i = \frac{|[\omega_i]|}{|\Omega|}$, which $\mathrm{P}(\vec{x})$ and $\mathrm{P}(\omega_1)\mathrm{P}(\vec{x}|\omega_i)$ can then be characterised as:

$$\begin{cases} \mathrm{P}(\vec{x}) = \mathcal{N}(\vec{x} \mid \theta_{\omega_{i=1\ldots k}}) \\ \mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i) = \mathcal{N}(\vec{x} \mid \theta_{\omega_i}) \end{cases} \tag{3.24}$$

3.3.3. Discussion: Decision Score Measure with Gaussian Models

Although we have proposed using Gaussian for modelling the confidence measure in Section 3.3.2, the confidence measure can be modelled based on probability models of any arbitrary kind. Having said this, the Gaussian models can still be seen as an optimal option for modelling confidence. From the definition in Section 3.3.1, μ_C is in fact a measure of the given class ω_i at the location $\vec{x}$ within the given measure space $(\mathcal{X}, X, \mu_C)$. Known that each class should have a feature vector $\vec{z}$ such that $\mathrm{P}(\omega_i|\vec{z}) = \max_{x \in X} E[\omega_i|x]$, where $\vec{z}$ is then considered as the expected feature vector of the given class ω_i in this case, i.e., the truth vector. With such understanding, any $\vec{x}$ can be considered as $\vec{z} + \varepsilon$, where ε is the random environmental error that has caused the actual reading variate from the truth vector $\vec{z}$. For a fictional example, a malignant breast tumour is expected to have n number of micro-calcifications for the most of time; where in this case, $\vec{z}$ equals to n if we define the number of micro-calcification as a one-dimensional feature vector in describing the malignancy of the breast tumour. As the tumour evolves differently through time and also impacts differently on different persons, the number of micro-calcification varies up and down naturally, which eventually cause the actual reading being away from the expected value $\vec{z}$ by ε units. Note that it is less likely to have the actual reading varying hugely from the expected value $\vec{z}$ but rather close to $\vec{z}$, where the likelihood of the observation decays along with the increase in the distance from the expected values $\vec{z}$. This decaying nature follows the principle of Gaussian distributions; the observations are more likely distributed around the expected value and are less likely to be further away. Following this line of argument, if we define the random environmental error ε to follow a Gaussian distribution as $\{\varepsilon\}\sim\mathcal{N}(\vec{z},\Sigma)$, i.e., to consider ε as a type of Gaussian noise, X must also follow a Gaussian distribution since $X \coloneqq \{\vec{z}+\varepsilon\}\sim\mathcal{N}(\vec{z},\Sigma)$, which explained why Gaussian models can be an optimal solution for measuring confidence.

However, one may argue that $\{\varepsilon\}$ might follow distributions of other kinds and therefore fundamentally change the nature of modelling. Indeed, if we reconsider the fictional breast tumour example provided, the integer nature of the feature derived in fact discretised the distribution of the observations; and pathologically many micro-calcifications might merge together and result into a visual image that very much like a macro-calcification and distorts the original distribution towards the negative side. All of these facts indicate that a Poisson distribution may suit better in this case for measuring confidence. However, these facts also indicate that the feature we derived is not accurate enough when describing the malignancy of the breast tumour. Alternatively, we can derive a better feature that combines the average size of the calcifications, the proportion of the calcified area in the whole nodule and their overall visibility to overcome the issues mentioned. With such an argument, following the concept of the Gaussian process, we can always derive a super feature vector $\vec{z}$ that best represents the studied object $\mathcal{X}$ under a specific high dimensional projection where every individual dimension follows a Gaussian distribution. Although we cannot mathematically guarantee that $\mathcal{X}$ can always be turned into a Gaussian process, we can certainly approximate them under a certain margin of errors. As a matter of fact, many machine learning research works have shown that random feature vectors of large dimensionality naturally converge into a Gaussian process (Sohl-Dickstein, et al., 2020), providing more ground for using Gaussian for modelling the confidence.

In addition, note that μ_C theoretically has a range of]0,1[when it is modelled on Gaussian distributions since $\mathcal{N}$ is always greater than 0 and less than 1. However, $\mathcal{N}$ may inevitably equal to 0 or 1 in practice when the numeric value surpasses the precision of hardware representation, which caused the range of μ_C becomes to [0,1]. In this case, any measure between -1 and 1 is reflecting the precise decision score at the given feature reading. A measure of 1 is reflecting absolute confidence in the decision made since the sum of the likelihood of the opposite classes is equal to 0 in this case. But such value with absolute confidence may also imply potential underfitting at the given feature reading, since it indicates some classes had very low support around the given domain. Regarding these discussions, a potential solution in improving computing precision based on PCA whitening will be discussed in Section 4.3.2. Issues regarding potential under fittings are further discussed in Section 4.5.2.

3.4. Confidence Measure Evaluation

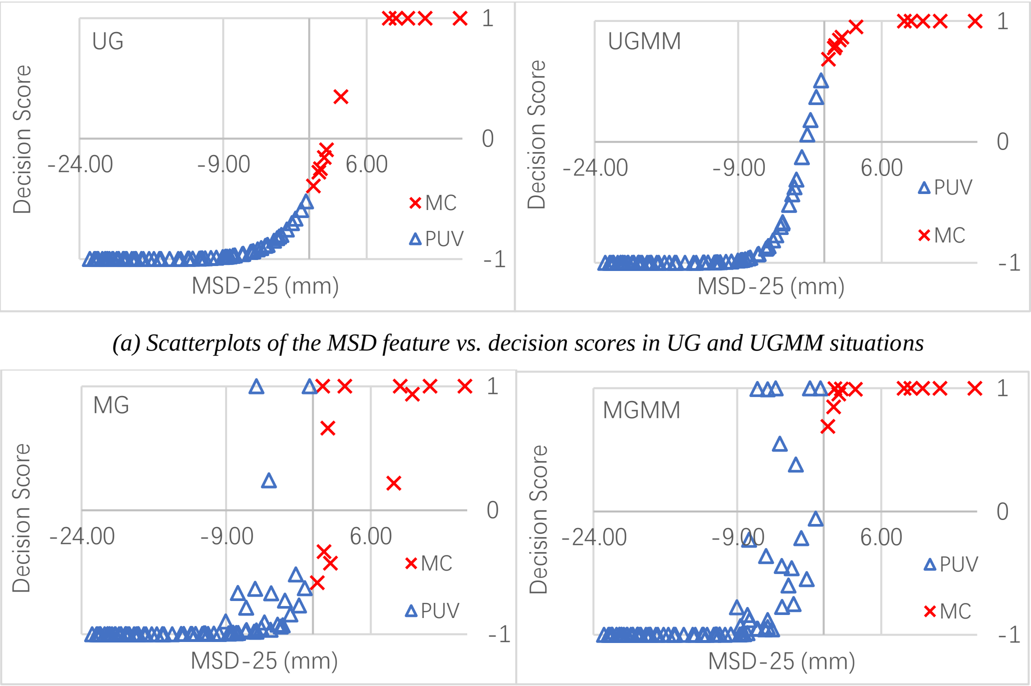
3.4.1. Dataset Used

In evaluating the usefulness of the confidence concept introduced in the previous section, this study conducted several experiments using the small early pregnancy dataset as introduced in Section 2.4.2. The whole dataset is divided into a training set of 94 examples (15 cases of MC and 79 cases of PUV), and a test set of 90 examples (11 cases of MC and 79 cases of PUV). This data set is chosen because of its simplicial nature with relatively low dimensionality and clear medical understanding, which provides a good start for testing the proposed methods. For more details on the dataset, see Section 2.4.2.

We first trained the proposed decision score model and derived their parameters based on the training dataset. We then applied the proposed models on each testing example from the test set and measured the decision score for each testing example. As introduced in Section 3.3, the decision score is within the range of [-1, 1], which can be seen as the decision confidence towards [PUV, MC] in this binary class dataset.

3.4.2. Evaluation of Zones of Confidence

Figure 3.2 presents the scatterplots of the decision scores against feature values along the single MSD dimension. According to the known literature in the related field of medicine, 25mm in MSD is a well-recognised threshold for separating PUV from miscarriage cases (Bourne & Bottomley, 2012). To better illustrate the change of predicted classes as the value of MSD increases, we have rescaled the MSD dimension by off-setting (25, 0) as the origin, then plotted the related decision score of each feature value in the test set for each model accordingly. The corresponding classes of the test examples are marked as a blue triangle (PUV) and red cross (MC) respectively. After the rescaling, the 1st, 2nd, 3rd and 4th quadrants in each scatterplot indicate the possible classification results, i.e., true positive, false positive, true negative and false negative respectively.



(a) Scatterplots of the MSD feature vs. decision scores in UG and UGMM situations

(b) Scatter plot of the feature values vs. decision scores in MG and MGMM situations

Figure 3.2 Illustration of Classification Confidence (UG: Univariate Single Gaussian on MSD, UGMM: Univariate Gaussian Mixture Model on MSD, MG: Multivariate Single Gaussian on three diameters; MGMM: Multivariate Gaussian Mixture Model on three diameters)

As shown in Figure 3.2, the confidence measured was very high for MC cases beyond 31mm and PUV cases below 16mm, where the zone of confusion covers the range between 16mm and 31mm with the maximum confusion close to 25mm. This finding itself is quite interesting due to the well-known fact that 16mm was a threshold previously practised in the USA that has only recently been revived to 25mm because of concerns of potential false positives (Bourne & Bottomley, 2012). This finding indicates that the confidence score does reflect the level of confidence in the diagnosis.

In addition, all the modelled data points were distributed according to a sigmoid pattern, which matches our expectation that the confidence would drop dramatically when it approaches the confusion point, i.e., the origin in the presented diagrams; otherwise, be stable at -1 or 1 when the feature value is outside the “confusion zone”. The scatterplot also shows that the use of GMM has resulted in data confusions being moved from the false negative region into the false positive region, since it better replicates the actual bias within the dataset. Figure 3.2 also shows the scatter plots of decision scores and feature values for multivariate situations. In clearly demonstrating the relationship between the decision scores and feature vector values, we purposely combined the three diameter components of each feature vector into a single average value (in fact MSD), and display the location of the data point along the

MSD dimension. At the same time, the decision scores are calculated using the original 3D feature vectors themselves.

Some general observations can be made from the scatterplots in both Figure 3.2 (a) and 3.2 (b). The confusion zone clearly exists between the two thresholds, and the best fitting curve through the confusion zone tends to be close to a sigmoid curve. The use of GMM also tends to move confusion cases from the false negative region into the false positive region, and at the same time increase the level of classification confidence in the true positive region. However, the confidence scores are more scattered in the confusion zone here than the scores for a univariate situation (a phenomenon to be further investigated in the next chapter). The second scatterplot shows increased cases of misclassification even with a high level of confidence.

In summary, the single Gaussian models tend to have a smoother fit to the sigmoid function, which reflected the nature of the proportional relationship between the MSD and the classification result. However, the performance of the multivariate Gaussian models shows that the data points are eventually made more distinguishable and pushed the classification results towards the two extremes, which may lead to the justification for multivariate fusion in differentiating classes in highly overlapped feature values.

3.4.3. Comparing Decision Model with Data Expectation

As demonstrated in the previous section, the proposed method provides a good indication of the range of confidence/confusion. However, does the decision score truly reflect decision confidence in real-life practice, and in precision? In fact, this is a very difficult question to answer. Unfortunately, such “real-life confidence” is not readily available in the training data set most of the time, nor easy to obtain. Even such confidence scores are available, they are normally based on subjective judgements by domain experts. Such a subjective judgement tends to be inconsistent, a problem known as “intra- and inter-observer variations”. In this section, we explore some alternative ways of modelling the “reality” and comparing our decision scores with such a modelled reality, and outline the limitations of these approaches.

One rudimentary solution is to map the decision score to absolute 1 or -1 according to the class label given in the training set, i.e., it is assumed that each of the decisions provided was made with absolute confidence. With this approach, we can evaluate the experiment result by calculating the difference between the projected decision score and the derived decision score from the proposed confidence model. This difference is within the range of [0, 2], where 0 indicates a perfect match, and 2 an absolute conflict between the two decision scores. However, this approach is too rudimentary, and the decision score is an oversensitive estimation, which does not reflect real expectation (the idea of the decision score sensitivity will be explored in the next chapter).

We can revisit the definition regarding the proposed confidence measure in Section 3.3.1. The measure $\mu_C(\omega_i|\vec{x})$ can be seen as a function that reflects the expectation of observing the class ω_i at the given feature $\vec{x}$. Therefore, if we collect a rich multiset of observations $X \in \vec{x} \to Z$ where Z indicates the ground truth of the class label, according to the law of large numbers (Dinov, et al., 2009), the expectation E regarding observing the class ω_i at the given feature reading $\vec{x}$ can then be estimated as

$$E(\omega_i|\vec{x}) = \frac{m_X(\omega_i)}{|X|} \text{ if } |X| \gg 1 \tag{3.25}$$

where $m_X(\omega_i)$ is the multiplicity function of the element ω_i in the multiset X, defined as

$$m_X(\omega_i) := \sum_{x \in X} \mathbf{1}_{\omega_i}(x) \tag{3.26}$$

Following this definition, the expectation of the Decision Score measure S_D can then be calculated as

$$E[S_D(\omega_i|\vec{x})] = E(\omega_i|\vec{x}) - E(\neg\omega_i|\vec{x}) \tag{3.27}$$

Consequently, we can derive a difference function $\Delta(S_D, E|\vec{x})$ as an evaluation method to compare the decision score we modelled and the real expectation value at the given feature point $\vec{x}$ as

$$\Delta(S_D, E|\vec{x}) = S_D(\omega_i|\vec{x}) - E[S_D(\omega_i|\vec{x})] \tag{3.28}$$

The difference value $\Delta(S_D, E|\vec{x})$ indicates the margin of error between the decision score we have modelled and the real expectation, in a range of [-2, 2], which should ideally equal to 0 if they are very close to the real expectation.

Unfortunately, obtaining such a large verification multiset X can be very difficult in the reality, which only a few individual readings of $\vec{x}$ at a different time can be obtained. Nevertheless, we can still estimate the result in a coarse manner by considering errors involved during data acquisition and feature extraction. In reality, errors inevitably exist during data collection and processing, where each reading in the verification multiset X at $\vec{x}$ are eventually drawn from $\{\vec{x}\}$ with error $\mathcal{E}$ instead of from $\{\vec{x}\}$ only, i.e.

$$X \in \{\vec{x} + \varepsilon_1, \vec{x} + \varepsilon_2, \dots, \vec{x} + \varepsilon_n\} \tag{3.29}$$

In this scenario, different readings can be considered as identical if their difference is within the maximum error margin, i.e.

$$X \in \{\vec{x}\} \text{ if } \bigwedge_{x \in X} |\vec{x} - x| < \varepsilon_{\max} \tag{3.30}$$

This condition grants more tolerances in obtaining a rich verification multiset.

In practice, there are two fundamental types of error occurring during data collection, which is systematic error (bias) and random error (accuracy). These two types of error can be affected by many factors but are mainly caused by the measuring instruments used and human observation error respectively. These errors are further evolved in the feature extraction stage of the pattern recognition process. Therefore, we can ideally calculate the theoretical error contained in feature reading by understanding how measurements are further used in the following experiments. However, data and methods may vary from application to application, which made it very difficult in measuring and quantifying the errors encountered. Nevertheless, there have been studies that summarized several commonly occurred error types in image-based experiments (Goldstein, 2000). In our experiment conducted in Section 3.4.2, we have used a human-labelled feature over the digital ultrasound image, which involves two major types of errors, as image pixelation error and cursor placement error. As the literature proposed (Goldstein, 2000), by combining the two types of errors, three potential error margins from ±0.064mm to ±1.8mm can be defined depending on the tolerance margin chosen in relating to different levels of uncertainty. Based on this study, we have computed the expectation at different feature readings based on the testing set and compared them with the decision score we derived from the SGMM model, which the result is shown in Figure 3.3.

From Figure 3.3, it is clear that the choice of error margin has an impact on the expectation computed, in which the range of the confusion zone is proportional to the margin of error. A small error margin may lead to under fittings since the number of elements within the margin is not satisfying the minimum requirement of the law of large numbers. On the other hand, a large error margin leads to a relative vague measure, which may not reflect the expectation at the feature reading precisely. In addition to these issues, a precise understanding of the errors contained regarding the feature used may not always be available, especially for some of the state of art features such as CNN. These defects add difficulties in evaluating the difference between our model and true expectation.

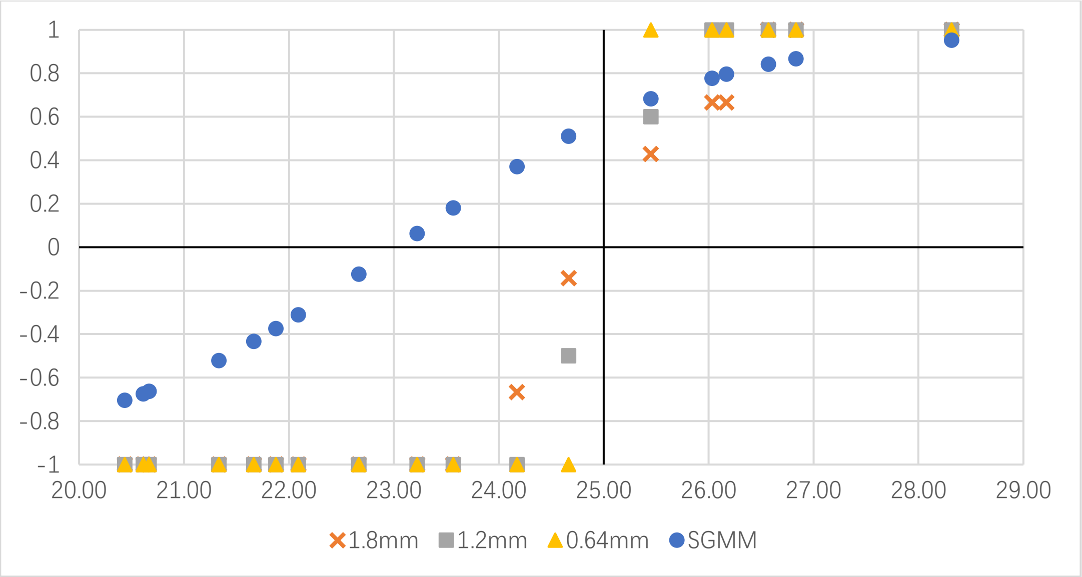


Figure 3.3 Expectation measured under different error margins

To avoid these defects, we can revisit the Formula 3.28 defined previously, which can be rewritten as

$$\Delta(S_D, E|\vec{x}) = |X|^{-1} \sum_{x \in X} \left[S_D(\omega_i|\vec{x}) - \mathbf{1}_{\omega_i}(x) + \mathbf{1}_{\neg\omega_i}(x) \right] \tag{3.31}$$

That is, the average of the sum of the difference/correction needed between the decision score computed and the absolute decision score based on ground truth, which can be easily computed on the fly. Following this, as $\Delta(S_D, E|\vec{x}) = 0$ indicates a perfect match at $\vec{x}$ between the two measures, the difference between the two measures must also equal to 0 at any region of the feature space if they match perfectly, i.e., for $\vec{x}_a, \vec{x}_b \in \{\vec{x}\}$

$$\int_{\vec{x}_b}^{\vec{x}_a} \Delta(S_D, E|\vec{x}) \, d\vec{x} = 0 \tag{3.32}$$

This again can be used as an evaluation method over a region, which the precise value can be calculated as

$$\frac{\vec{x}_a - \vec{x}_b}{|X|} \sum_{\vec{x}_i \in [\vec{x}_a, \vec{x}_b]} \sum_{x \in X} \left[S_D(\omega_i|\vec{x}_i) - \mathbf{1}_{\omega_i}(x) + \mathbf{1}_{\neg\omega_i}(x) \right] \tag{3.33}$$

That is, the average amount of difference/correction needed between the decision score computed and the absolute decision score based on ground truth within the region. The advantage of this measure is that it does not limit the region of interests, which can be as large and convenient as we needed in satisfying the law of large numbers. However, it is noticed that the measure becomes less representative along with the increase in the range of inspection, which sacrifices evaluation preciseness in exchanging expectation accuracy. In balancing the two, we can perform the difference measure as Formula 3.33 over several regions/bins of reasonable length when evaluating the performance of a decision model on a validation set. In performing such a piece-wise difference measure, it is needed to project the original data into a single-dimensional space to ease the computational cost in searching and sampling. That is, we are producing a set of projected validation samples $\{x'\} \in X'$, then sampling and evaluating the decision score model by defining regions/bins on the projected dimensions. Based on this, the *i*th difference measure $\int \Delta(i)$ of *k* regions/bins can then be calculated as

$$\int \Delta(i) = \int_{x'_{\mathrm{min}}+(i-1)(1-\mathrm{s})\ell}^{x'_{\mathrm{min}}+[i(1-\mathrm{s})+\mathrm{s}]\ell} \Delta(S_D, E|\vec{x})\, d\vec{x} \tag{3.34}$$

where ℓ denotes a unit length when dividing the projected space $x'_{max} - x'_{min}$ into *k* regions/bins with *s* stride, which calculated as

$$\ell = \frac{x'_{\mathrm{max}} - x'_{\mathrm{min}}}{[k(1-s)+s]} \tag{3.35}$$

in this definition, the final average of the difference measure among all divided regions/bins after disposed of repeated strides can be calculated as

$$\frac{1}{k}\sum_{i=1}^{k}\left[\sum_{n=1}^{\varsigma}\frac{(1-s)\int\Delta(i)}{n}+\frac{[1-(1-s)\varsigma]\int\Delta(i)}{\varsigma+1}\right] \tag{3.36}$$

where ς denotes the max amount of the strides that can be fitted in a unit length and calculated as

$$\varsigma = \lfloor\frac{1}{1-s}\rfloor \tag{3.37}$$

As defined, *k* and *s* are the two essential parameters that affect the sampling of the validation set during the evaluation. *k* can be any positive integer, but it is ideal to maximise the value of *k* in obtaining a much precise result while making sure that each of the regions/bins has at least a sample size that satisfies the law of large number. On the other hand, *s* can be any real number that is in the range of [0,1[, but it is ideal to have the *s* being relatively small and making $(1-s)\varsigma = 1$ in minimising the computation cost while again satisfying the minimum sample size required by the law of large number.

As a pilot experiment, we have used MSD as the projection method and evaluated the 4 types of decision score models proposed in this chapter with *k* = 5 and *s* = 0.2.

Table 3.2 The average reading of the difference measure on different models

	MGMM	*SG*	*SGMM*	*MG*
Δ	-0.1827	0.1007	-0.0939	0.0225
Stdev	0.2473	0.2115	0.1709	0.1874

The result in Table 3.2 shows that all 4 models had similar variations of the performance over the validation set with an average standard deviation around 0.2, which implies that they had a similar pattern in reflecting the real expectation of the dataset in different regions. This matches what we have observed in Section 3.4.2. In addition to this, MGMM had the worst performance with the largest difference of -0.18. This observation again matches the conclusion in Section 3.4.2, since we realise MGMM was the most complex model and therefore expecting potential under fittings/over fittings on a small training set. A good indication of this assumption is shown by the *Δ* result of SGMM and MG, they all had a significantly better result than MGMM despite they share similar principles. SGMM has integrated the original 3-dimensional feature into 1-dimensional space, which provides a more tolerable degree of freedom to the regression of model parameters. On the other hand, MG has combined 2 classes into 1 model in overcoming the lack of training samples on the malignant class. These factors very well explained the reason why SGMM and MG significantly outperform the MGMM in online testing by reducing the potential underfitting and overfitting respectively. However, despite the relatively poor performance on the miscarriage dataset, MGMM is believed to be the most potent model when facing complicated features as long as the size training sample size is large enough to overcome the underfitting issue, since MGMM has the most capability in representing complex distributions in precise.

3.5. Summary

In this chapter, we have first highlighted the essential needs of measuring the confidence of classification in the clinical environment. In fulfilling such needs, we have reviewed a list of potential methods for measuring the classification confidence and selected probability measure as the solution due to its universality on applications. In further refining the probability measure for solving specific requirements for CDSS, we have proposed a generic measure, referred to as the confidence measure. This proposed measure was modelled based on the Bayesian principle and Gaussian models of four various kinds, which later been transferred into a signed measure, referred to as the decision score measure, for a more simplified and informative representation. We have debated the applications of the proposed method and tested the proposed models on small scale real-world data with limited dimensionality. The experiment result showed that all variants of the proposed measure had a good reflection on the confidential nature of the experiment dataset used, where the UGM showed good fitness to the decision strength and MGMM showed the best discrimination power.

In addition to these findings, we have also proposed several methods that can be used for evaluating the fitness of the proposed confidence measure. Initial experiment results have also shown the measure's validity. we have further discovered that the proposed measure tends to be more sensitively to feature inputs in high dimensional space. Nevertheless, the dimensionality of the experiment dataset was not great enough in proving such an argument. Therefore, we will further study the behaviours of the measure proposed in high dimensional space, which is the main topic for the next chapter.

Chapter 4. Confidence Measures in High Dimension

In the previous chapter, we have proposed a generic confidence measure and its use in a signed decision score, as well as various forms of modelling the measure for the classification decision strength using the Gaussian Bayes principle. The proposed measure was tested on a dataset of three-dimensional features and their aggregated one dimensional feature (i.e. MSD) for detecting miscarriages. The test results have shown that the measure produces valid measurements and supports the known medical findings. However, data sets of such low dimensionality are not common in the intended application domain. In medicine and health, not only there are data sets with tens or even hundreds of variables representing medical test results measured and recorded manually, but also data sets with hundreds or even thousands of features that are extracted from medical images automatically by computer algorithms. In either case, a dataset can be of very high or extremely high dimensionality. High dimensional feature spaces encounter the issue of the “curse of dimensions” (see Section 4.1). Although various procedures and algorithms have been proposed to reduce dimensionality as a preprocessing step before machine learning, the dimensionality of most features passed into a trained classification model remains high. It is very important and necessary to understand how the proposed confidence measure behaves in high dimensional feature spaces, and its impacts.

In this chapter, therefore, we extend our investigation into the behaviours of the decision score measure introduced in the previous chapter in high dimensional spaces. To facilitate this investigation, we will consider data sets of automatically extracted features of high dimensionality from medical images. We will focus on two potential issues that affect the behaviour of the proposed decision score measure, in the context of *singularity* and *sensitivity*. These terms are first introduced and investigated rigorously and then illustrated with synthesized data. We then propose a solution by projecting the previously proposed confidence measure into an iteratively filtered eigenspace. Effects of the proposed method are then evaluated with real-life data sets of different dimensionalities through experiments. Afterwards, the decision score in high dimensional features is analysed. The result of this study together with the conclusions from the previous chapter will be further exploited in the proposed fusion schemes to be presented in Chapter 5.

4.1. Issues Arise from Curse of Dimensionality

Curse of dimensionality (Hongbo, 2010) refers to the fact that as the dimensionality of data space linearly increases, the amount of search space increases exponentially. Besides the fact that the time needed in the data processing increases, data points in a high dimensional space become more spread and dispersed, which then distorts the statistical nature of a modelled measure. The sparse range of the Euclidian space causes distance measurements and regression analysis of various types to become less meaningful. The Hughes phenomenon (Hughes, 1968) states a negative relationship between the predictive power of a model and the dimensionality of data when the number of training samples is fixed. This is because the lower coverage in high dimensional space leads to lower support for the classifier predicted outcomes. The smaller a training set is, the worse the impact of the Hughes phenomenon is. Consequently, as the data dimensionality increases, the number of training examples has to grow exponentially in maintaining model accuracy. Due to difficulties and costs for clinical data acquisition, the exponential increase in the demand of training examples can be seen as an unrealistic luxury.

The curse of dimensionality has significant impacts on our proposed decision score measure that are manifested in a variety of ways beyond the efficiency of computations. The curse of dimension has been dealt with in the literature by a variety of dimension reduction techniques such as PCA (Nasution, et al., 2018). Precisions of computation, in relation to the covariance matrix associated with the dimension reduction procedure, usually impose certain limitations that are mathematically measured by two matrix properties identified as *singularity* and *sensitivity*. Next, we shall describe these two issues and discuss their impacts.

4.1.1. Singularity

The concept of singularity is a major matrix algebra that describes the solvability of a linear system A***x***=***b*** when A is a square matrix. Such a matrix system has a unique solution if and only if A has an inverse or equivalently, i.e., det(A) $\neq 0$. If A has no inverse it is said to be singular.

The proposed decision score measure has its root in the Bayesian probability model. Although many studies that have adopted the Bayesian approach (such as Naïve Bayes) assume conditionally independent features (Wood, et al., 2019), which is not strictly required by the Bayesian theorem. However, the covariance matrix Σ calculated from each class must be positive semidefinite in the modelling function in Formula 3.23, where positive semi-definite matrix refers to a symmetric matrix M with a real number $z^T Mz$ that is positive or zero for every nonzero real column vector z. If the covariance matrix Σ was not positive semi-definite, the terms for $|\Sigma|$ and Σ^{-1} renders the formula computationally undefined. In fact, if the determinant $|\Sigma|$ is a very small non-zero the computing inverse matrix Σ^{-1} results in underflow, i.e., Σ is computationally ***singular***.

The covariance matrix of an n dimensional dataset is a $n \times n$ matrix whose diagonal terms $\{\mathit{\Sigma}_{i,i}\}$ are the *i*th variances ${\sigma_i}^2$ among the *n* dimensions. The rest of the terms $\left\{\Sigma_{j,k} \middle| \, j \neq k\right\}$ can be defined as a linear transformation of the product of the *j*th and *k*th standard deviations $\sigma_j\sigma_k$ among the *n* dimensions with a gradient of the pair wised Pearson correlation coefficient $r_{j,k}$, i.e.

$$\Sigma = \begin{bmatrix} {\sigma_1}^2 & r_{2,1}\sigma_2\sigma_1 & \cdots & r_{n-1,1}\sigma_{n-1}\sigma_1 & r_{n,1}\sigma_n\sigma_1 \\ r_{1,2}\sigma_1\sigma_2 & {\sigma_2}^2 & \cdots & r_{n-1,2}\sigma_n\sigma_1 & r_{n,2}\sigma_n\sigma_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r_{1,n-1}\sigma_1\sigma_{n-1} & r_{2,n-1}\sigma_2\sigma_{n-1} & \cdots & {\sigma_{n-1}}^2 & r_{n,n-1}\sigma_n\sigma_{n-1} \\ r_{1,n}\sigma_1\sigma_n & r_{2,n}\sigma_2\sigma_n & \cdots & r_{n-1,n}\sigma_{n-1}\sigma_n & {\sigma_n}^2 \end{bmatrix}, r \in (-1,1) \quad (4.1)$$

It is a well-known fact that singularity occurs if there are linear relations of any kind across any two or more different dimensions in the training data, i.e., $|r_{j,k}| = 1$. This is due to the fact that correlated vectors have the exact same direction under any projections. Any raw data acquired may naturally cause a singularity threat. When dimensionalities are very high, it is more likely to contain duplicates on extracted features, yielding a singular Σ.

4.1.2. Sensitivity

The sensitivity of a decision model refers to the rate of change in the decision score predicted in terms of change in feature values. This statement is very similar to the definition of condition numbers (Belsley, et al., 1980), which is a measure of how much the output value changes in relation to a small change in the inputs. A large condition number implies a significant change in the output with a small change in the input, which refers to as the ill-conditions. On the contrary, a small condition number implies a minor change in the output with a small change in the input, which refers to as the well-conditions. In theory, well-conditions imply a much predictable and stable outcome of a system, whereas ill-conditions imply a much sensitive system. Next, we shall be studying the concept of sensitivity by first looking at the definition of the condition number.

Definition: The condition number of a matrix *A* is defined as its norm multiplied by the norm of its inverse, i.e.

$$cond(A) = \|A\|\|A^{-1}\| \quad (4.2)$$

where *A* is commonly expressed by a non-singular square matrix. Although condition number does not limit the type of norm being used for calculation, however, when the conventional

Euclidean norm is used, the condition number can be expressed as the ratio of the largest singular value of A to the smallest one.

Condition number has its limitations and may not be very suitable for studying the sensitivity of our proposed measures. In our study, complicated functions such as the MGMM model usually have nonlinear characteristics, which creates difficulties in applying the measure of condition numbers. In addition, computing the condition number of a matrix can be much complex in real life where A is neither guaranteed to be non-singular nor being square. Although it is still possible to use its pseudo-inverse of such a matrix, the calculation may well be complex and limited. Most importantly, the singularity challenges still exist as what we have discussed in Section 4.1.1, given that $\lim_{\det(A)\to 0} cond(A) = \infty$. Therefore, we need another appropriate type of measure for sensitivity. Known as the *decision sensitivity measure*, it is defined as follows:

Definition: Given a classification system, let S_D be its decision score function. The sensitivity of this decision score function can be defined as the gradient (i.e., first-order derivate) of S_D, i.e.

$$\mathrm{S}_D{}'(\omega_i|\vec{x}) = \mathrm{S}_D(\omega_i|\vec{x})\frac{d\mathrm{S}_D}{d\vec{x}} = \mathrm{S}_D(\omega_i|\vec{x})[\frac{\partial \mathrm{S}_D}{\partial x_1}, \ldots, \frac{\partial \mathrm{S}_D}{\partial x_n}] \tag{4.3}$$

The high decision sensitivity of the systems results in major changes in the predicted decisions with respect to a minor change in feature values, which indicates potential overfitting of the trained decision model. On the other hand, very low sensitivity results in only marginal differences in the decision score as a result of a marginal change in the input feature vectors, indicating potential underfitting of the trained model because of its indistinguishable classification results. In supervised learning, the modelling of sensitivity heavily depends on the training data used and the chosen classifier. As discussed in Chapter 2, for medical diagnostic systems, it is important to consider an acceptable application-dependent level of sensitivity to have some margin of tolerance to noise (or data errors).

In general, an ideal classifier is expected to be sensitive enough in distinguishing different classes while remaining insensitive among data objects of similar cases. Consequently, the decision score model should be sensitive within the confusion zone where decisions over different classes are made, but insensitive outside the confusion zone where the belonging of classes is more settled (see Section 3.2.1 for the definition of confusion zone). With a proper definition, the decision sensitivity measure can be used as a good indicator of the reliability of the decision score model.

The sensitivity of the decision score model at a given feature vector input can be measured using Formula 4.2 in an ideal situation. However, as we discussed in Section 3.4.3, noises exist inevitably in any real-life measurements. Therefore, it is reasonable to measure decision sensitivity over an interval around the feature vector value readings in practice. If an ideal error margin ε of a feature value reading is introduced, the average decision sensitivity at a given feature value point within the error margin can then be measured as the definite integral, obtained from Formula 4.3:

$$\frac{1}{2\|\varepsilon\|}\int_{\vec{x}-\varepsilon}^{\vec{x}+\varepsilon} \mathrm{S}_D{}'(\omega_i|\vec{x})\, d\vec{x} \tag{4.4}$$

which can be simplified to or approximated by 4.4：

$$\frac{1}{2\|\varepsilon\|}|\mathrm{S}_D(\omega_i|\vec{x}+\varepsilon)-\mathrm{S}_D(\omega_i|\vec{x}-\varepsilon)| \tag{4.5}$$

The decision sensitivity measured in this way has a range of $[0, \|\varepsilon\|^{-1}]$. The decision score predicted is considered as certain when the sensitivity measured equals 0 since the model was producing consistent estimation within the error margin around the given feature point. On the contrary, the decision score predicted is considered as uncertain when the decision sensitivity measured equals to $\|\varepsilon\|^{-1}$, since the entire zone of confusion was enclosed within the error margin in this case. To avoid cases where the decision sensitivity measurement equals to $\|\varepsilon\|^{-1}$, we can optimise the decision score model so that the range of the confusion zone is wider than the error margin. That is, if we define $\vec{x}_{min}$ as the feature reading at the start of the confusion zone and $\vec{x}_{max}$ as the end of the confusion zone,

$$\|\vec{x}_{\mathrm{max}}-\vec{x}_{\mathrm{min}}\| > 2\|\varepsilon\| \tag{4.6}$$

However, such a requirement may fail in a high dimensional space due to inconsistent distribution across different dimensions and potentially overfitted trained models. These factors will be further investigated in Section 4.4.2.

4.2. Influencing Factors of Decision Score Measure

In the previous section, we noted that the singularity issue raises concerns about the applicability of our proposed model for the decision score measure in a high dimensional feature space. In addition, the potential use of decision sensitivity parameters in assessing our decision score model is also highly influenced by the dimensionality of the feature space. Therefore, controlling/reducing feature dimensionality becomes necessary for building reliable decision score measures.

Various methods for selecting or aggregating subsets of highly correlated features (i.e., coordinates) into a new set of reduced number of features have been commonly used. However, such methods require an in-depth understanding of the correlation among a large number of features and hence are deemed infeasible for feature space of significantly high dimensions. Unsupervised dimension reduction methods such as Principal Component Analysis (PCA) (Nasution, et al., 2018) can be effective alternatives. PCA in fact has been widely used for reducing feature dimensionality while keeping the information with the most discriminating power. It is used to project data from the original high dimensional space into a new space of a selected set of eigenvectors, along which the variance of the data is maximised. The number of the eigenvectors to keep depends on the required tolerance of information loss.

The PCA dimension reduction method is data-dependent. The output projection matrices are therefore closely related to the variation of the training sample from the assumed probability distribution of the overall infinite data population. Consequently, determining the essential influencing factors in relation to the singularity and sensitivity issues highlighted above depends on the assumed probability model of the data population. Our general assumption here is that the data feature space are sampled from a Multivariate Gaussian Model (MGM) with given mean and standard deviation matrices μ and Σ, as defined in Section 3.3.2.

As unprocessed features can be highly skewed and hard to observe, it is considerably easier for studying the behaviours of MGMs in high dimensional space if we standardise them first. To achieve this objective, we need to ensure first that MGMs in any dimensionality is standardizable. Let $X \sim \mathcal{N}(\vec{\mu}, \Sigma)$ for some $\vec{\mu} \in \mathbb{R}^n$ and Σ be a real $n \times n$ positive semi-definite matrix, then there must be a matrix $B \in \mathbb{R}^{n\times n}$ such that

$$\mathrm{Z} = B^{-1}(X - \mu) \sim \mathcal{N}(0, I)$$

Here, Z is considered to be a collection of independent standard normal random variables, i.e., $Z = \{Z_1 \dots Z_n\} \sim \mathcal{N}(0, I)$. Following this, the above expression can be simplified by the linear expression $X = BZ + \mu$, which states a linear relationship between the original MGM and its standardised version (Do, 2008). In other words, any random variable X with a multivariate Gaussian distribution can be interpreted as the result of applying a linear transformation $X = BZ + \mu$ to a collection of n independent standard normal random variables Z. Therefore, any MGM $\mathcal{N}(\vec{x}|\vec{\mu}, \Sigma)$ can be transformed into a product of independent Univariate Gaussian Models as

$$\begin{cases} \mathcal{N}(\vec{x}|\vec{\mu}, \Sigma) = \displaystyle\prod_{i=1}^{\dim \vec{x}} \mathcal{N}\big(\Lambda(\vec{x})_i \big| \Lambda(\vec{\mu})_i, \lambda_{i,i}\big) \\ \Lambda(\vec{x}) = \nu^{-1}\vec{x}\nu \end{cases} \tag{4.7}$$

where ν is a matrix whose columns are the corresponding right eigenvectors of a diagonal matrix of eigenvalues λ of the original covariance matrix Σ so that $\Sigma\nu = \nu\lambda$.

As previously shown in Formula 4.1 we can simply present the Σ as a transformation of correlated standard deviations. Similarly, λ can also be presented by using standard deviations as the eigenvalue and standard deviation are interchangeable. As λ has already been projected into an independent (orthogonal) space, the correlation factor r^2 equals to one in the diagonal of λ and the equals to zero in the remaining parts. As a result, the diagonal of λ is a set of pure variances as $\langle {\sigma_1}^2, {\sigma_2}^2, \ldots, {\sigma_n}^2 \rangle$ with the reminding element in the matrix equals to zero, which the Formula 4.3 can then be rewritten as

$$\begin{cases} \mathcal{N}\left(\vec{x} \middle| \vec{\mu}, diag(\lambda)\right) = \prod_{i=1}^{\dim \vec{x}} \frac{1}{\sqrt{2\pi {\sigma_i}^2}} e^{-\frac{(\Lambda(\vec{x})_i - \Lambda(\vec{\mu})_i)^2}{2{\sigma_i}^2}} \\ \Lambda(\vec{x}) = \nu^{-1}\vec{x}\nu \end{cases} \tag{4.8}$$

Note that Formula 4.8 always has a maximum reading when $\vec{x} = \vec{\mu}$. At this point, a density peak is formed and can be simply computed as:

$$\prod_{i=1}^{\dim \vec{x}} \frac{1}{\sqrt{2\pi}\sigma_i} \tag{4.9}$$

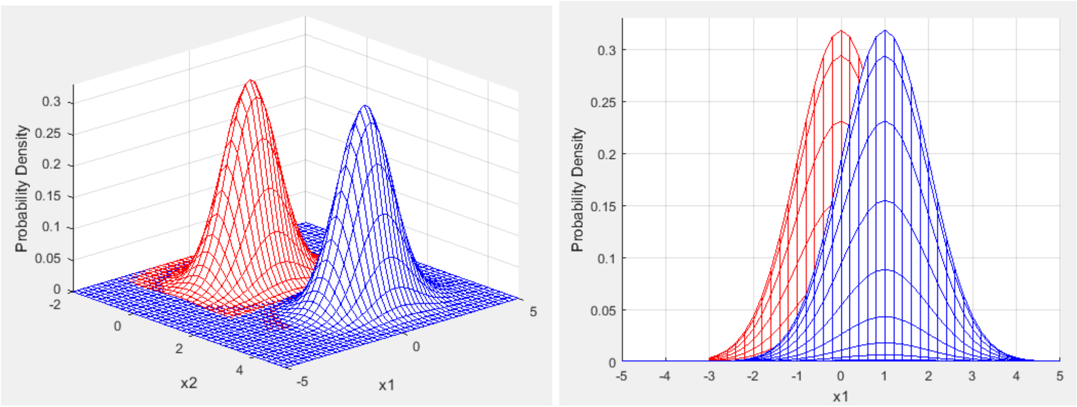
Consequently, the range of the density function can be estimated as $\left[0,\ \prod_{i=1}^{\dim \vec{x}} \frac{1}{\sqrt{2\pi\sigma_i}}\right]$, which implies that the variation of the density value is directly proportional to $\prod_{i=1}^{\dim \vec{x}} \frac{1}{\sigma_i}$ with a coefficient of $\frac{1}{\sqrt{2\pi}}^{\dim \vec{x}}$. Also, it should be noted that the product of these eigenvalues has a minimum over any non-random matrix, in which the value supposed to be closer to the minimum eigenvalues, i.e.,

$$|\sigma_{max} - \sigma_{max}\sigma_{min}| \geq |\sigma_{min} - \sigma_{max}\sigma_{min}| \tag{4.10}$$

which implies that the result can be very much dominated by σ_{min} and inversely proportional to the variation.

In summary, we have noticed that the range of the decision score measure depends on the dimensionality $\dim \vec{x}$ and standard deviation values σ_i. As the modelling sensitivity reflects the range of values in the feature domain, the dimensionality and the modelling variances are considered as the two critical factors in controlling the sensitivity of a decision score measure.

The dimensionality of feature space is expected to have an impact on the discrimination power between different classes, and hence affects the sensitivity of a decision score measure. The two figures presented below may assist the understanding of the link between dimensionality and model behaviour over individual classes. Figure 4.1(a) shows a pair of clearly separated classes modelled in a 2D feature space (x1, x2). However, as shown in Figure 4.1(b), the projection into one of the two dimensions (x1) lead to significant overlapping of the two classes. Nevertheless, dimension reduction can be achieved differently by firstly transforming the feature space and then selecting fewer coordinates of the transformed feature space.



(a) (b)

Figure 4.1 Probability Models in 2 and 3 Dimensional Space

The second influencing factor is the standard deviation of the feature used in each dimension, where this factor can be highly influenced by dimension reduction procedures. PCA outputs the eigenvectors and their associated eigenvalues in descending order, which helps the user pruning the dimensions by keeping the projected features with larger eigenvalues. Therefore, in studying this second influencing factor, it is essential to understand how our proposed decision score measures change along with the change in eigenvalues (standard deviations), where we have made some illustrations using a simplified model under different conditions.

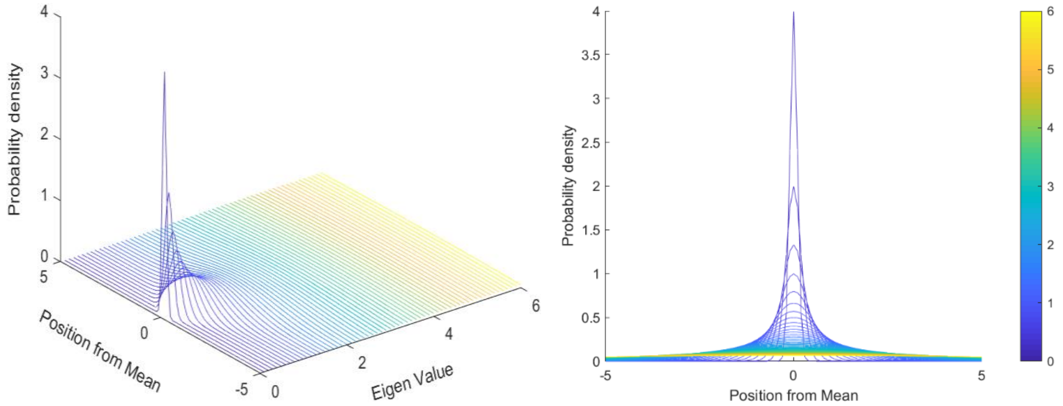


Figure 4.2 Change in Probability Density Under Different Eigenvalues

Figure 4.2 shows how the univariate Gaussian probability density function varies in terms of different input values sampled under different standard deviations, where the standard deviation is the same as the square root of the eigenvalues under such a setting. From the illustration, we can observe that a small eigenvalue leads to a very steep probability distribution. The probability density predicted with a small eigenvalue heavily depends on the feature value, where it produces extremely high readings around the mean of the distribution, but the reading drops dramatically as the feature value moving away from the mean. In other words, the model tends to be very sensitive when it was built on small eigenvalues. On the contrary, the probability density predicted with a large eigenvalue maintains relatively independent to the change of feature value, where it produces very steady readings under a wider spectrum. In other words, the model tends to be very insensitive when it was built on large eigenvalues. Therefore, as the Gaussian probability model serves a fundamental role in our decision score measure, we conclude that the eigenvalues inevitably influences the sensitivity of the decision score measure, where the level of sensitivity tends to be inversely proportional to the scale of eigenvalue.

Nevertheless, it is also important to understand that our proposed decision score measure does not only involve one Gaussian probability model as illustrated in Figure 4.2, but multiple Gaussian probability models with different means and standard deviations. Therefore, we further illustrate how decision score measures vary in correspondence to different value inputs under different standard deviations in Figure 4.3. To keep the illustration clear, we have simplified the setting where only two univariate Gaussian Models representing one class each are presented, both variables have the same standard deviation, and their means are 3 units away from each other.

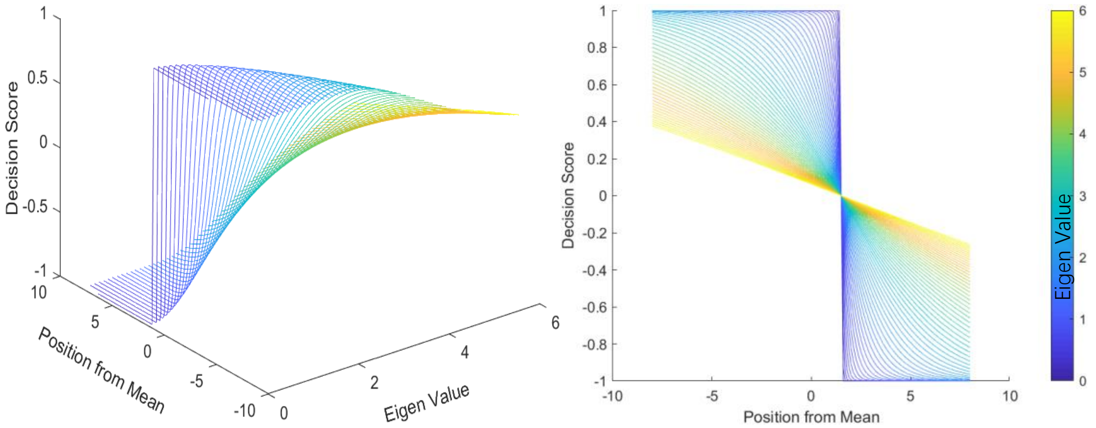


Figure 4.3 Change in Decision Score Under Different Eigen Values

As the figures show, a small eigenvalue causes the decision score measurement to perform very much like a step function, where a marginal difference in the input value results in a dramatic change in the score predicted. On the contrary, a large eigenvalue causes the decision score measurement to perform very much like a linear function, where the decision score changes in a constant manner along with the change in the input values. Therefore, we can clearly verify that the sensitivity property was well inherited from Gaussian models into our proposed decision score measure. It is worth noting that the behaviour of decision score measure very much depends on the properties of each Gaussian model used, which may result in slightly different characters than what appears in Figure 4.3. However, the principle of the negative relationship between the eigenvalues and the sensitivity of the decision score measure remains.

4.3. Mitigating the Constraints of the Influencing Factors

As mentioned in Section 4.1.1, the potential correlations across different dimensions very likely lead to a singular covariance matrix and cause failures in measuring the decision confidence. Hence, our ultimate goal is to minimise the potential correlation across different feature dimensions for improving the robustness of the decision score measure. As a solution, it is sensible to filter (or project) raw features into an altered feature space, which removes or replaces the perfectly correlated dimensions and produce a filtered semidefinite positive covariance matrix Σ', i.e.

$$\Sigma' = \{\Sigma \,|\, |r_{j,k}| \neq 1\} \tag{4.11}$$

Perfectly correlated dimensions rarely occur in real-life data sets. It is more common to have a correlation $r_{j,k}$ being close to but not exactly equal to 1, which may cause computational errors and precision flaws in computerized systems. Therefore, the filtering condition of the correlation is better to adopt a threshold $\varepsilon \approx 0$ to indicate the maximum error tolerance margin, rather than looking for exact matches. Thus, we can reformulate Formula 4.11 as follows:

$$\Sigma' = \{\Sigma \,|\, |r_{j,k}| < 1 - \varepsilon\} \tag{4.12}$$

4.3.1. Correlation-based Filtering of Dimensions by Selection

A naive solution to the problem can be simply de-selecting the dimensions that correlate to the others by using exhaustive searches through all possible combinations of different dimensions. This naïve solution can be simply implemented using multivariate linear regression analysis, as correlation by definition is nothing but the linear regression fitness of one dimension to another. More specifically, the target dimension should be de-selected if the regression fitness is greater than a predefined threshold. With this method, the raw data will be filtered by only selecting the associated dimensions that are not considered as perfectly correlated (under the tolerance margin ε) to any other dimensions. The result is a pruned feature space that guarantees to produce a semi-definite covariance matrix. The advantage of such a method is that it has a minimal change to the original features and their meanings. In other words, pruning is only made to redundant and replicate dimensions that may have little or no effect on class discrimination. In addition, the reduction of dimensionality using this method also results in a smoother confidence prediction and diminishes the risk of potential misclassifications as the predictions became less sensitive.

Despite those advantages as mentioned, the computational cost of this method can be a huge deficit, especially when the original features are of very high dimensionality. Although the computational cost is meant to reduce through the iterations of the deselection process, where the best-case complexity can be in the order of Ω (*n*), the worst-case complexity of the solution is still in the order of $O(2^{n-1})$. More importantly, it is worth noting that regression analysis already has a computational cost of at least $\Omega(n)$, which the naive solution by feature selection will only act as an amplifier of this cost and render the implementation extremely inefficient or even infeasible.

4.3.2. Correlation-based Filtering by Principal Component Analysis

Although PCA has been primarily used for dimension reduction, it can be adapted for solving the singularity issue arising from the covariance matrix. The PCA scheme produces an orthogonal linear transformation of the original dataset, which decorrelates any possibly pre-existing correlations and provide a new coordinate system so that the maximum variation is maintained over the least amount of dimensions involved. The PCA transformation is conventionally implemented through Eigen Value Decomposition (EVD), but occasionally it can also be done using Singular Value Decomposition (SVD). The major difference between the two is that the EVD can only be applied to a square matrix but SVD can be applied to any rectangular matrix. In the practical application of using PCA for dimension reduction, two matrix representations of the data records can be constructed whereby the columns (or the rows) are the vector representation of the data samples. Hence, if the dimension of the data records is different from the number of samples, then both matrices are rectangular. There are two ways of generating square covariance matrices, by multiplying either records matrix by its transpose once on its left and once on its right. The SVD is simultaneously applied on the two data matrices (after subtracting the computed average vector), while the EVD is applied on one of the covariance matrices. Despite the fact that the result between EVD and SVD should not differ significantly, we prefer using the SVD method over the EVD method in this research because SVD is more numerically reliable when computed over a symmetric positive semi-definite matrix (the covariance matrix) (Datta, 2010).

More specifically, for SVD, any row data matrix A of size $m \times n$ can be uniquely represented as

$$A = USV^T \tag{4.13}$$

where U and V^T are $m \times m$ and $n \times n$ singular vectors respectively, which being orthogonal, i.e., $UU^T = I$ and $V^T V = I$; S is a matrix with the same size of A that contains zero except along its main diagonal for the corresponding singular values. Following this definition, a special case appears when $m > n$, where U becomes a large $m \times n$ matrix with the last $m - n$ columns being considered as unnecessary fields. In this case, people normally adapt it into an economy-sized SVD (MathWorks, 2013), which becomes more memory efficient in real practices. In this form, the original U and S are pruned by only preserving the first n columns, which are eventually reduced to $m \times n$ and $n \times n$ matrices respectively; V^T remains the same. In the economy-sized SVD, the projected feature vector$\vec{x}_{\text{PCA}}$ of the original feature vector $\vec{x}$ can then be simply defined as

$$\vec{x}_{\text{PCA}} = U\vec{x} \tag{4.14}$$

By the end of the process, $\vec{x}_{\text{PCA}}$ acts as an independent feature within the orthogonal space. Consequently, the potential singularity threat is resolved. Nevertheless, the singularity issue can also be solved by projecting the original data into other spaces, as long as the dimensions in the projected space are minimally correlated. However, the advantage of using PCA is that it produces an independent multivariate Gaussian model as we mentioned in Formula 4.7, where the multivariate Gaussian model can be formed as a product of n univariate Gaussian models from each projected dimension and therefore reduce computational cost.

Computing PCA over large matrices can be very expensive in real life. Therefore, an iterative approach is normally adopted. One of the most commonly used and well-established iterative solutions is the NIPALS-PCA algorithm (Risvik, 2007). However, one important issue regarding this algorithm is that the projected features may eventually lose orthogonality due to the errors accumulated in each iteration, especially when the data computed has very high dimensionality. This problem can be mitigated by applying the Gram-Schmidt orthogonalization method to correct the non-orthogonal principal components computed by the NIPALS method (Andrecut, 2008).

Although the dimension reduction schemes suggested above can help mitigate the singularity problem, we still need further modifications to reduce the influence from the sensitivity issues. As discussed in Section 4.2, the standard deviation σ plays a very important role in controlling the sensitivity of the decision score measure. We shall next discuss methods to deal with this problem.

4.3.3. Dimension Controlling based on Eigenvalues

The sensitivity can be controlled by modifying, via pruning, the selection of the eigenvectors of the PCA/SVD schemes by considering the respective variance (eigenvalues) σ^2 of each independent dimension. As we have explained, projected dimensions with large eigenvalues provide most of the information and ensure the trained classifier being robust, whereas projected dimensions with smaller eigenvalues provide the finest discrimination power and contribute to more precise classification (we shall further discuss this comment in Section 4.5). To optimise the use of the PCA and resolve the potential sensitivity issue, we first define ${\sigma_{min}}^2$ as the minimum eigenvalue required for computing decision scores in providing the essential discrimination power, and define ${\sigma_{max}}^2$ as the maximum eigenvalue required for computing decision scores with tolerable consistency. With the two parameters defined, the independent covariance matrix S, also known as the singular value matrix in the economy SVD, can then be further pruned as

$$U' = \{U_i \mid {\sigma_{min}}^2 < S_{i,i} < {\sigma_{max}}^2\} \tag{4.15}$$

By further pruning, any projected dimensions with eigenvalues below the minimum threshold are considered invalid as they may produce over sensitive decision score measures. Similarly, eigenvalues beyond the maximum threshold are also considered invalid as they are producing very general information and may not contribute to any discrimination power. This process will not only reduce the dimensionality of the feature used, but also adjust the eigenvalues so the optimal sensitivity can be achieved. We shall further test and validate these comments with experiments in Section 4.5.

4.3.4. PCA Whitening

The whitening transformation, also known as the sphering transformation, is a linear transformation that project random variables into a new dimensional space where the new covariance of the projected variables is equal to the identity matrix, i.e., uncorrelated and each dimension has a variance of 1. The transformation is called "whitening" because it changes the input vector into a white noise vector (Mobasseri & Lulu, 2021).

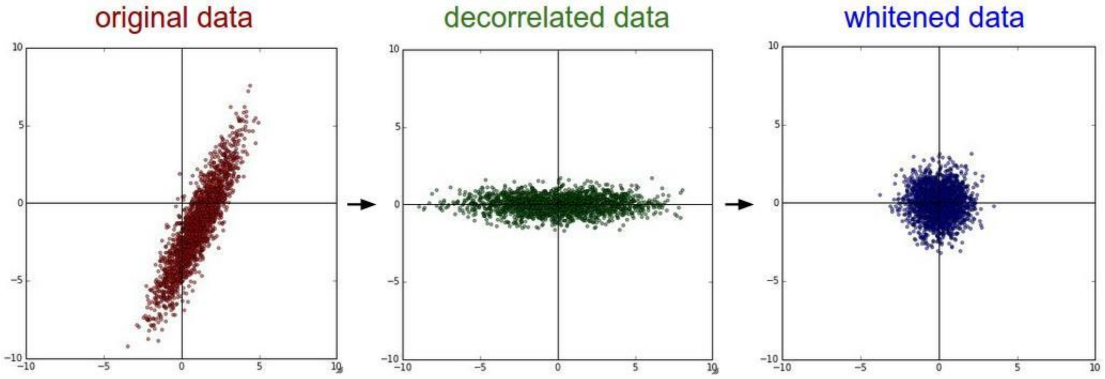


Figure 4.4 PCA Transformation and Whitening

As the Gram-Schmidt PCA already decorrelates the original data, the whitening process in this case simply becomes a standardization on the data. The variance of the decorrelated data has already been computed in PCA as the eigenvalues $S_{i,i}$, which the whitened data vector $\vec{x}_{\text{PCA white}}$ can then be computed as

$$\vec{x}_{\text{PCA white}} = \frac{\vec{x}_{\text{PCA}}}{\sqrt{diag(S)}} \tag{4.16}$$

where $diag(S)$ represents the diagonal of the eigenvalue matrix *S*.

In principle, the whitening process is not necessary for our study since it does not change the topology nature of the data and therefore does not influence the score eventually computed. However, it does play an important role in practice since whitening standardizes the eigenvalue in each dimension and therefore eases the management. More importantly, it also improves the computability of the probability model in practice, since the standardisation lowers down the precision requirements when dealing with extremely small numerics.

Having discussed the various modification of the adopted dimension reduction scheme(s) that helped mitigating the singularity and sensitivity problems, now we are in a position to formulate the correlation-based filtering task. As discussed in Chapter 3, the continuous probability models always yield a measurement of $\mathrm{P}(\omega_i|\vec{x})$ in a range of (0,1). Following the definition in Section 3.3.2, if we present $\mathrm{P}(\vec{x})$ as a probability mixture model $\sum_{i=1}^{n} \mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i)$, the conditional probability measure $\mathrm{P}(\omega_i|\vec{x})$ can then be defined as

$$\mathrm{P}(\omega_i|\vec{x}) = \frac{\mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i)}{\sum_{i=1}^{n} \mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i)} \tag{4.17}$$

where the conditional probability $\mathrm{P}(\vec{x}|\omega_i)$ regarding a feature reading given each class may all be very close to 0^+ in rare cases. Consequently, formula (3.20) can be reformulated as follows:

$$\lim_{\mathrm{P}(\vec{x}|\omega_i)\to 0^+} \frac{\mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i)}{\sum_{i=1}^{n}\mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i)} = 0 \tag{4.18}$$

However, the real-world computation always suffers from imprecise fraction numbers presentation, and hence formula (4.17) may occasionally cause errors due to division by zero. In addition, the algorithm may also produce a faulty value of 1 due to the indistinguishable difference between the numerator and the denominator. These errors appear more often when the projected feature dimensions have very small eigenvalues, as many outliers may frequently laying outside the measurable range. With the help from PCA whitening, it enlarges the measurable range of the confidence model by scaling up small eigenvalues with multiplication operation and scaling down large eigenvalues with division operation. In addition to the traditional whitening process, a scaling parameter r can be introduced as

$$\vec{x}_{PCA\,white}{}' = \frac{\vec{x}_{PCA}}{\sqrt{r \cdot diag(S)}} \tag{4.19}$$

where the measurable range is inversely proportional to r. Note, a very small r may also cause the overflow issue, which again makes the computation becoming invalid in reality. In principle, r should be set and optimised according to the intended application, however, we found that $r = 1$ is considerably a reliable value in general based on experimental experience, which provides robust performance while maintaining the simplicity of computations.

4.4. Assessing the Quality of Decision Scoring Measures

In practice, our proposed decision score measure produces real values in a range of [-1,1], which determine the class label and strength of the decision made by referring to the distribution learnt from training examples. In our definition, the magnitude of the decision strength is proportional to the level of certainty, where great values indicate high certainties on classification, and low values indicate uncertain predictions. To assess the quality of decision scoring measures, we need to distinguish between class prediction at the training stage and at the evaluation stage. In the training phase, decision score models are created by maximizing the likelihood of each class, where ideally matched class labels are awarded positive values and mismatches are awarded negative values. This strategy can be simply assessed by counting the number of positive matches of each class, as an accuracy measure, where the ultimate measure equals 100% to indicate ideal adherence to the rules/heuristics defining the respective class labels. This kind of assessment can also be applied to the validation phase, but 100% accuracy would never be expected in real practice.

Despite accuracy can be used for assessing the quality of the decision score model trained, it is important to understand that the accuracy measure is a type of discrete measure and does not suit the numeric nature of the decision score perfectly. Therefore, it is also important to evaluate the strength value of the predicted decision score instead of just the sign of it. As a naïve solution, we can assess the strength of the decision scores in the validation phase by first dividing the test results into two groups, as the samples that being correctly and incorrectly classified. For an ideal classifier, high confidence values in the correctly classified cases and low confidence values in the miss-classified cases are to be expected, respectively. This desirable trend can be visualized by drawing a histogram of each respective group. As Figure 4.5 illustrates, the ideal histogram for correctly classified samples should be concentrated at the two ends, forming a U-shape curve, whereas the ideal histogram for miss-classified (or incorrectly classified) samples should be concentrated in the centre, forming an A-shape curve. This is desirable as we want the level of confidence for correctly classified examples as high as possible, and the level of confidence for incorrectly classified examples as low as possible.

By using the visualisation as shown in Figure 4.5, we can gain good understandings of the decision score measured. However, the subjectiveness of the evaluation became a major deficiency when applying this kind of evaluation protocol in practice. The lack of quantification readings during the evaluation made it very difficult to compare results in real life, especially when the compared models were being very similar to each other. Therefore, we urgently need a method for assessing the strength of the decision score with appropriate quantifications, which will be further discussed in this section.

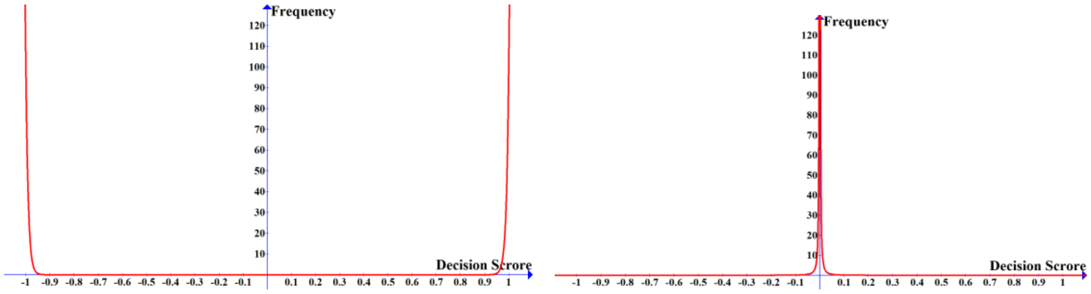


Figure 4.5 Ideal histogram plots of correct and miss-classified decision scores

4.4.1. Measuring sensitivity of a Decision Score

In Section 4.1, we highlighted two main influencing factors on any decision score model, i.e., the singularity of the covariance matrix and the sensitivity of the decision score function under the concept of the rate of change in the decision score measured. The measurement of sensitivity can be used as an appropriate tool for decision score quality assessment with some adaptations, as the sensitivity partially reflects how the trained decision model reacts to unseen data examples.

As we have presented in Formula 4.3, sensitivity can be essentially studied by measuring the gradient of the decision score function at different feature readings. However, determining the exact estimation of a decision score function gradient can be very costly, especially in high dimensions. Therefore, in our study, the gradient of the decision score function is calculated using the following approximation method:

$$\nabla S_D = \frac{S_D(\omega_i|\vec{x}+h) - S_D(\omega_i|\vec{x})}{||h||} \tag{4.20}$$

where h is considered to be an extremely small number and has been set to 10^{-15} in our experiments (see Section 5.4). Under such approximation, the decision score function can be seen as a black box and the output of which is computed without prior detailed knowledge of the functional characteristics. As a result, it does not only decrease the computational cost on the calculation, but also improves the adaptability of the sensitivity measure for quality assessment.

For high dimensional feature vectors, the gradients of our proposed decision score measures are obtained post projecting the original feature vectors $\vec{x}$ into the eigenspace, which simplifies the characteristics of the problem. In such a case, the elements $\{x'_1, x'_2, \dots, x'_n\}$ in the projected feature vector $\vec{x}'$ are distributed as a list of independent Gaussian variables in the projected orthogonal space. Each of the partial derivatives calculated, under such condition, is coming from independent univariate Gaussian variables, and therefore simply expressed by the orthogonal Jacobian matrix:

$$\mathrm{S}_D\left(\omega_i|\vec{x}'\right)\frac{\partial \mathrm{S}_D}{\partial \vec{x}'} = \begin{bmatrix} \frac{\partial \mathrm{S}_D}{\partial x'_1} & \cdots & \frac{\partial \mathrm{S}_D}{\partial x'_n} \end{bmatrix} \tag{4.21}$$

Since we are more interested in the changes of quantities instead of directions when measuring sensitivities, it is better to combine the gradient vectors into a scalar by simply taking the inner product of the vector with itself as follows:

$$\nabla S_D = \begin{bmatrix} \frac{\partial \mathrm{S}_D}{\partial x'_1} & \cdots & \frac{\partial \mathrm{S}_D}{\partial x'_n} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathrm{S}_D}{\partial x'_1} & \cdots & \frac{\partial \mathrm{S}_D}{\partial x'_n} \end{bmatrix}^T \tag{4.22}$$

which is simply the Euclidian norm of the Jacobian matrix. Using such a method, we can then easily obtain a set of sensitivity measurements $\nabla S_D = \{\nabla S_{D1}, \nabla S_{D2}, \dots \nabla S_{Dn}\}$ for any given testing set $X' = \{\vec{x}'_1, \vec{x}'_2, \dots, \vec{x}'_n\}$. Since the decision score function S_D by nature having sigmoid characteristics, then its first derivative must follow Gaussian characteristics, where the breadth of the Gaussian distribution is a direct reflection of the range of the confusion zones.

Consequently, the variance of these sensitivity measurements $\nabla\sigma^2$ can then be treated as a good reflection of the change in sensitivities at different test readings, defined by:

$$\nabla\sigma^2 = \frac{\sum_{i=1}^{n}(\nabla S_{Di} - \overline{\nabla S_{\mathcal{D}}})^2}{n-1} \tag{4.23}$$

In practice, the proposed evaluation method will be applied and compared across features of different dimensionality. However, the gradients measured in this case is proportional to the range and dimensionality of the feature used, where gradients measured from high dimensional feature with wider ranges can be higher than the ones measured from low dimensional features with narrower ranges. In drawing a fair comparison across features of different dimensionality, we have normalised the computed $\nabla\sigma^2$ through a transfer function in obtaining a coefficient ∇c regarding the sensitivity measurements of the test set, provided by the formula:

$$\nabla c = \sqrt{\frac{\nabla\sigma^2}{3\sqrt{\dim(\vec{x})} + \nabla\sigma^2}} \tag{4.24}$$

where ∇c is a real number that has a range of [0,1). As we increase the projection dimensionally, the range of calculated scalar values ∇S_D changes due to the fact that the score function gets steeper and yield a much larger Euclidian distance in the final calculation. Therefore, we have introduced another normalization factor $3\sqrt{\dim(\vec{x})}$ within the normalization function, where it is designed to normalize the sensitivity output into 0.5 when the variance of each dimension was equal to 1 individually. To highlight, this normalization factor is set by following the conventional rule of thumb of normal distributions, but it should not be constrained to such value only. Other alternatives are also allowed as long as it is proportional to the $\sqrt{\dim(\vec{x})}$. With such a measure, the decision score model is considered as sensitive to a set of testing samples if ∇c is close to 1, and be considered as insensitive if ∇c is close to 0.

4.5. Empirical Evaluation and Result Analysis

To evaluate and analyse the effects of dimension reduction on classification sensitivity, we conduct experiments on the CBIS-DDSM dataset (Lee, et al., 2017) using PCA-based methods. The overview of the data set can be found in Section 2.4. For this study, the experiments are conducted on 1,872 images of *calcium* type tumours that consist of 1,199 benign cases and 671 malignant cases, as calcium type tumours has relatively more distinctive looks that ease the requirement of image enhancement. The *mass* type of tumour is excluded from the data for this experiment as they require more sophisticated features for classification. We have made this pruning to the experiment dataset as our work is more interested in discussing the decision score model behaviours than proposing the best feature and building very good classifiers. These images have been further randomly organized into 10 individual patches to implement a 10-fold cross-validation process. Nevertheless, this random sampling process was stratified to ensure the original prior of the dataset remained undistorted. In the other words, each of the sampled patches is designed to maintain the same ratio between benign and malignant classes, which was set at about 1.78:1 as the initial ratio in the dataset before partitioning.

The well-known Grey Level Co-occurrence Matrix (GLCM) texture-based feature has been used in many classification studies about mammography (Majeed, et al., 2013) (Elshinawy, et al., 2011). In GLCM, the matrix is computed based on pixel neighbours to reflect the frequency of the occurrence of certain patterns, where the pattern can be identified in different distances and angles. After the raw data matrix has been computed, statistical measurements are normally extracted in forming high-level descriptors with uniform dimensions. In this study, we follow the widely used practice as proposed by Haralick (Haralick, 1979). The GLCM matrix, in this study, is computed for three different distances (1, 2 and 3) and four different angles (0°, 45°, 90°, and 180°). Then 13 statistical moment measurements (excluding the maximal correlation coefficient) are extracted from the raw matrix, resulting in a feature space of $3 \times 4 \times 13 = 156$ dimensions.

The model adopted for this study is an MGMM with a Naïve Bayes classification scheme for simplicity. Besides, we used the PCA method presented in Section 3.1 to project the original data for building the MGMM model into an orthogonal space, which does not only ensure the independence requirement of the Naïve Bayes classifier, but also simplified the original MGMM into a Univariate Gaussian Mixture Model (UGMM). As the coordinates in the orthogonal space are linear combinations of multiple coordinates in the original space, the data points in the projected space are therefore reflecting information from multiple dimensions, especially when the projected coordinates correspond with large eigenvalues. Therefore, modelling each projected dimension with a UGM only is no longer sufficient, which is considerably more desirable if we model the projected data points with Univariate Gaussian Mixture Models (UGMMs). Consequently, the class models derived from the projected multidimensional space are eventually taken as a mixture of UGMMs. However, the creation of UGMM on each projected lower dimension can no longer follow the approach mentioned in Section 3.3.2 due to the ambiguity of projection from different dimensions. As a solution, UGMM on each projected dimension was created by adopting the Expectation-

Maximization (EM) method (Kumar, et al., 2009), where the threshold of the log-likelihood has been set to 10^{-5}.

4.5.1. Number of Mixtures and Feature Dimensionality

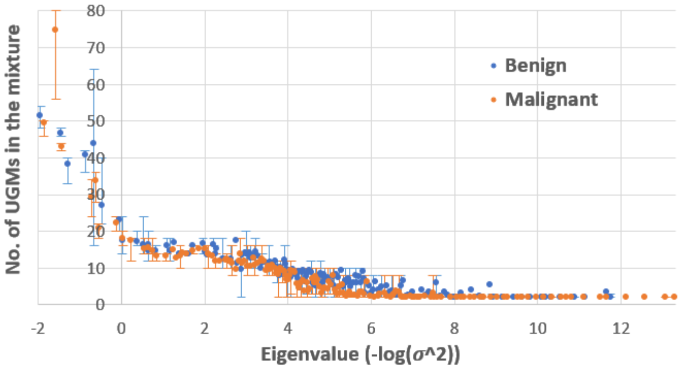


Figure 4.6. Number of UGMs in the mixture in each projected dimension with different eigenvalues

In our experiment, we have first investigated the number of UGMs in the mixture on each projected dimension in relation to the eigenvalues. The average of the 10 test patches is plotted in Figure 4.6, where the error bars are indicating the maximum and minimum readings among the 10 test patches. The scatter plot shows a clear positive relationship between the number of UGMs in the mixture and their eigenvalues on the dimension projected, where the dimensions with larger eigenvalues tend to require more UGs in the mixture to describe the behaviours of the class. This does match our expectations since the dimensions with larger eigenvalues tend to contain more information and therefore yields a more complex projection. We have further observed that the number of UGMs required on the projected dimensions falls significantly as the corresponding eigenvalues decrease; however, this trend becomes stabilized after the eigenvalue has fallen below 1. This fact can be seen as experimental evidence of the "eigenvalue-one criterion", which states that the projected dimension with an eigenvalue that is less than 1 can be dropped due to the relatively small information gain from them (Cardoso & Cruz-Almeida, 2016).

4.5.2. Altering Decision Score Measure Using GLCM Features

Following this analysis, the overall classification accuracy under different thresholds on maximum eigenvalues ${\sigma_{max}}^2$ and minimum eigenvalues ${\sigma_{min}}^2$, as explained in Formula 4.15, are recorded and plotted in Figure 4.7. The scatter points in these plots represent the average of the cross-validation results and the error bars reflect the best and worst readings among them. The initial seed used for generating the 10-fold random samples was fixed. Therefore, the testing environments under different eigenvalue thresholds are identical and comparable.

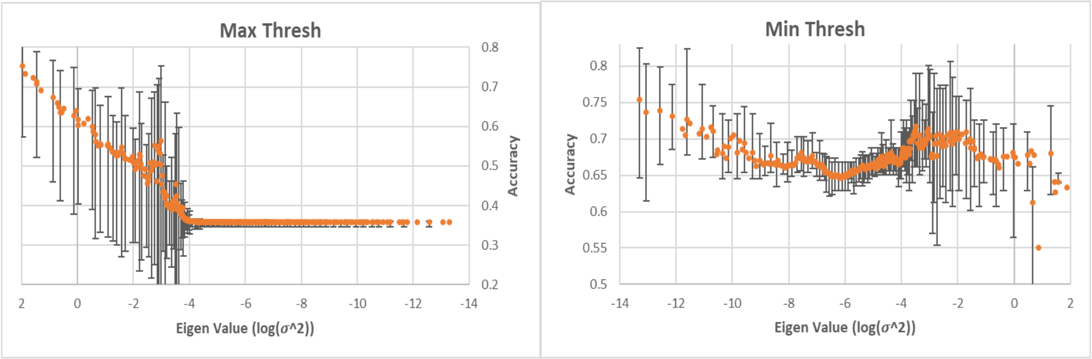


Figure 4.7. GLCM Accuracy in Relation to Different Eigenvalue Thresholds

As shown in Figure 4.7, thresholding on maximum eigenvalues had a clear and significant impact on the overall accuracy, which reached a minimum and remained stable at $10^{-3.9}$. This rather linear impact was caused by the strong proportional relationship between the scale of eigenvalues and the amount of information gained from them. Abandoning dimensions with large eigenvalues directly reduces the information gained by the classifier and therefore impairs the classification accuracy, the impact of which appears to surpass the ambiguities contained within these dimensions. The stabilized reading after the decay indicates that the remaining dimensions no longer provide sufficient amounts of additional information to support further classifications, causing the predictions to bias one of the classes consistently. Furthermore, thresholding on maximum eigenvalues appears to have a consistently large error margin. This can be caused by the eigenvalues on the minimum side, as discussed in Formula 4.10, where small eigenvalues tend to have a more dominating influence on the predictive model and therefore making it extremely sensitive. These effects will be further discussed in more detail in the next paragraph.

In contrast to the previous readings, thresholding on minimum eigenvalues had a moderate effect in general, whereby accuracy decaying initially, then followed by a steady increase after 10^{-6} and finally ending with another significant decrease. The initial decrease in accuracy reading is very much understandable since the reduction in dimensionality causes more ambiguities in the lower-dimensional subspace. Meanwhile, as we have mentioned in Formula 4.10, smaller eigenvalues should have more dominant effects compared to large ones. This fact is more likely to cause the decision model to be overfitted in the high dimensions. This explains the reason for the increase between 10^{-6} to $10^{-3.5}$, implying that the projected dimensions with eigenvalues less than 10^{-6} can potentially cause classification overfitting. Removing these eigenvalues essentially makes the classifier more robust and improves testing accuracy. Evidence that supports this argument is the error bars reflected on the scatter plots. The error bars remain consistently large at the beginning of the plot and then starts to decrease in size along with the increase in accuracy, which indicates that the initial classification results were very sensitive and unstable to different test sets but subsequently become more and more robust along with the pruning of dimensions with small eigenvalues. At the end of the plot, the continuous pruning of dimensions starts to have an escalating effect on the information loss and eventually causes the classifier to predict more errors. This is reflected by the drop in accuracy and by the increase in error margins.

Regarding the sensitivity measurements, the method introduced in Section 3.3 reflects the sensitivities of a group of samples, which makes the evaluation results being strongly dependent on the testing data. Consequently, it yields a large variation in the test readings. Nevertheless, the average reading among the testing patches will still be a good indication of the overall sensitivity under different thresholds. The diagrams in Figure 4.8, show that the averages of the sensitivity plot have clear negative sigmoid trends on both maximum and minimum threshed diagrams, reflecting that our sensitivity function defined matched our initial expectation.

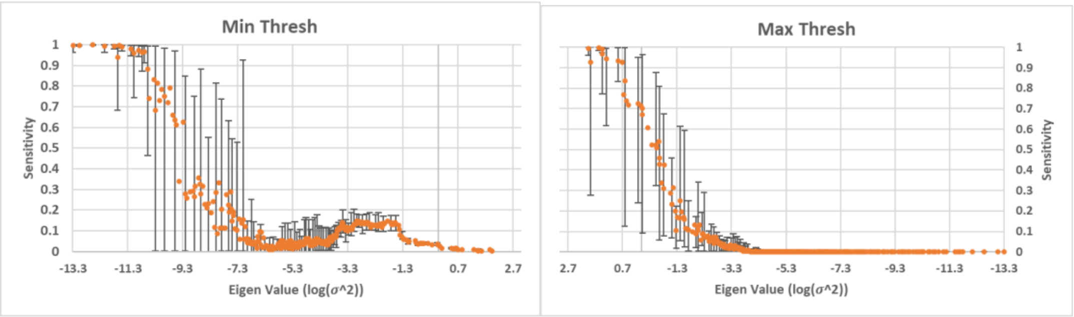


Figure 4.8. GLCM sensitivity measurements in relation to different eigenvalue thresholds

Thresholding on minimum eigenvalues had a clear impact on the classification sensitivity. This again reflects our expectation since small eigenvalues tend to generate abrupt MGMM decision models that are very distinct from the others, which cause the measurements to be susceptible. As the experiment results show, cohering to the observations from Figure 4.7, sensitivity decreases significantly in the beginning and then eventually reached a minimum at 10^{-6}, and finally ended with another drop after $10^{-1.4}$. The initial decrease in sensitivity essentially demonstrates the reduction in classification overfitting along with the reduction in dimensionality, which reached a floor eventually between 10^{-6} and 10^{-4} as the robustness of classification was established in testing. After a robust classifier is built, further pruning on the eigenvalues results in an additional reduction on the information gained from the discriminative factors and therefore force the classifier to focus more on the ubiquitous factors. On one hand, this helped classifiers gaining better knowledge over comparative factors and therefore produce more sensitive predictions. On the other hand, however, this also causes the classifier to focus more on much ambiguous features that in return reduce the power of the classifier in discriminating different classes. Finding the best balance between robustness and the discrimination power of a classifier is a challenge. The entangled relationship between robustness and discriminating power was shown as another positive sigmoid curve between 10^{-6} and $10^{-1.5}$ on the sensitivity plot, where the classifier remains insensitive in the beginning and then transit rapidly to reach another ceiling and being relatively sensitive. This showed that the classifier was initially benefited from removing the over discriminative factors but soon reached a maximum since the remaining factor starts being more and more ambiguous and couldn't contribute more to the classification. It can be potentially harmful when pruning beyond this local maximum, since the remaining factors may not be sufficient in discriminating different classes. Such effect can be clearly observed on the diagram as the ceiling was only maintained for a short period then followed by a second decrease in sensitivity immediately, indicating that the consistent errors made started to cause the classifier bias towards one of the classes and therefore led to insensitive predictions. In general, the clear drop in sensitivity at the beginning of the scatter plot was an indication of possible model overfitting, but the significant change in sensitivity at the end of the plot was reflecting possible model underfitting. An ideal threshold should be a value that positions at the beginning of the second peak within the plot, where the classifier with pruned dimensions maintains the best trade-off between robustness and discrimination power. This ideal threshold is observed at $10^{-3.5}$ in our experiment.

Compared to thresholding on minimum eigenvalues, thresholding on maximum eigenvalues has shown a much consistent impact on the sensitivity. However, this does not imply that thresholding on maximum eigenvalues affects sensitivities similarly. The constant reading of extremely sensitive results at the start of the plot was essentially an observation of the dominating effects from the lower eigenvalues. The significant decrease in sensitivities after $10^{-1.1}$ was again caused by the decrease in classification accuracy, where the bias generated by classification error eventually yields insensitive predictions. Therefore, maintaining minimum eigenvalue unchanged eventually preserve the high sensitivity yield by the overfitted prediction models, which cause the effect of thresholding on maximum eigenvalues to become less obvious and noticeable. We have again discussed this relation with Formula 4.10 as the minimum eigenvalues have a greater impact on the decision outcome than the maximum values.

4.5.3. Altering Decision Score Measure Using LBP Features

To validate the findings, we have also tested with the Local Binary Pattern (LBP) features, which are of a different kind of texture feature from GLCM, on the same dataset. LBP is a very popular local texture feature used in computer vision, which re-encodes each central pixel on the original image into one byte binary code depending on the relative contrast of the surrounding pixels to the central pixel. The histogram of these embedded codes for the whole image becomes a descriptive feature of image texture in 256 dimensions. Unlike the GLCM feature, dimensions in the LBP feature extracted are considerably less correlated, since each one of them is representing a distinctive pattern of the order relationship between the pixel and its 8 neighbouring pixels. We then present our findings in Figure 4.9.

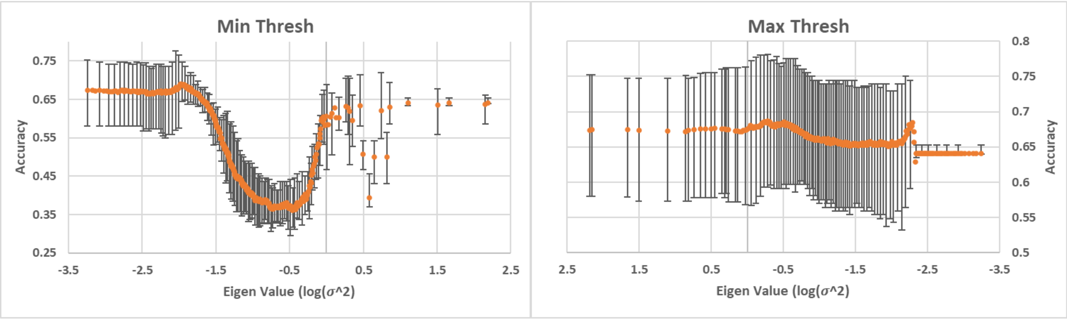


Figure 4.9. LBP accuracy measurements in relation to different eigenvalue thresholds

As the test result shows and as we expected, the LBP feature extracted has a much lower correlation compared to the GLCM feature used previously, where the LBP had a minimum eigenvalue at 10^{-3} compared to the GLCM which had a minimum eigenvalue at 10^{-13}. The LBP feature has a relatively worse performance compared to the GLCM feature, achieving only 77% accuracy in the best case. Nevertheless, here, we are more focused on the overall effects of the eigenvalue thresholding but not on the precise accuracy value, despite the accuracy being only slightly better than the expectation. In this experiment, we still observe the down-up-down trend pattern when thresholding the minimum eigenvalues, which further validated our expectation. Besides, we have also observed improvement by thresholding maximum eigenvalues for the first time. We believe this is due to the less correlated feature extracted having weakened the dominance of the small eigenvalues, making the effect of maximum eigenvalue thresholding becoming much more observable. As Figure 4.9 shows, thresholding on maximum eigenvalues has boosted the average accuracy by 1% but were still relatively less effective in comparison to thresholding on minimum eigenvalues. Furthermore, thresholding on maximum eigenvalues had consistently large error margins, inherited from the small eigenvalues. Both observations again confirmed our expectation on eigenvalue thresholding.

The sensitivity plots of the LBP feature have shown much less variations compared to the GLCM features. As Figure 4.10 shows, the sensitivity of the LBP feature remains consistently low despite its high dimensionality. This is eventually a reflection of the poor performance of LBP feature in classifying breast tumours, which was not sensitive to the variation of input features. However, we are still able to pick up a few valuable information when zooming into the charts. As we have highlighted in the plot of thresholding on minimum eigenvalues, the sensitivity measure still variates slightly between $10^{-2.3}$ to $10^{-1.2}$. Within the range, we are able to observe an increase in the sensitivity measure, which indicates that thresholding on minimum eigenvalues was still able to improve the performance of the classifier by making it more sensitive to different feature inputs. In addition, we also observe two peaks at $10^{-2.05}$ and $10^{-1.94}$ respectively, which again matched our discoveries with GLCM features but only within much smaller ranges. Similar to the same rationale as we have explained in the experiment with GLCM features, thresholding on minimum eigenvalues was meant to improve the classifier performance and the sensitivity measures started to change at $10^{-2.3}$, where all the redundant eigenvectors were removed. The change soon reached the maximum and then was followed by decaying, that is when the entanglement between dimension reduction and over ambiguity start to appear. Following the same principle, as we have discussed in the last experiment, the best threshold for the minimum eigenvalues is expected to be found at the start of the second peak, which was $10^{-1.94}$ in this experiment. As expected, this threshold does not only being optimal on the sensitivity plot, but also reflected with more robust performance and the best accuracy on the accuracy plot, which again confirmed the clear potential of using sensitivity measures for optimising eigenvalue thresholds for classifications.

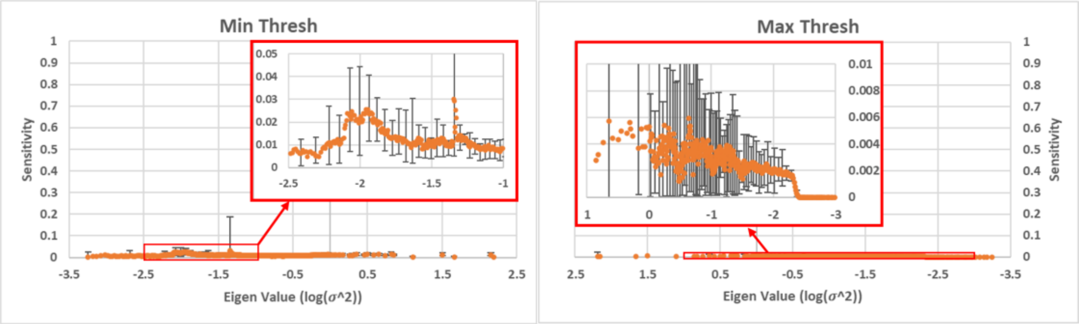


Figure 4.10. LBP sensitivity measurements in related to different eigenvalue thresholds

On the other hand side, thresholding on the maximum eigenvalues again showed us inferior performance compared to thresholding on minimum eigenvalues. Despite the insensitive sensitivity readings, we were still able to see the decaying sensitivity measure along with the reduction of dimensionality when zooming into the image. These foundings were again confirmed our expectations as we have discussed in the previous experiment.

4.6. Further Discussions on Issues and Challenges

Our preliminary experiments were conducted on controlled variables, where one of the two thresholds always remain constant at the maximum/minimum evaluations. However, it is still desirable to further test our hypothesis in a fully variable environment to reveal the relationship between the two thresholds. In the remaining parts of this chapter, we shall highlight and discuss some aspects of the challenges in carrying out these tests.

4.6.1. Finding Optimal PCA Thresholds

Tuning the eigenvalue thresholds on both maximum and minimum sides in a consistent manner can be challenging due to the different magnitudes of information contained on each side. A practical solution in determining the appropriate thresholds may rely on the use of Confidence Interval (CI). In statistics, the confidence interval is a type of estimation that defines the probability (or likelihood) of observing a certain event within an interval for a certain level of confidence (Kragten , 1994). In a two-tails test, the confidence level is always bounded with an upper limit and a lower limit, which can be adapted as the minimum and maximum thresholds of the given observation on the computed eigenvalues. In practice, a probability distribution model can be first created from the eigenvalues observed. The CI analysis can then be applied to this distribution. As a result, we should be able to obtain the maximum and minimum eigenvalue thresholds as the upper and lower limits of the CI at a specified confidence level. In this way, the eigenvalue thresholds can be determined and tuned in the context of confidence levels, which became an empirical method depending on the environmental requirements.

However, CI analysis is assuming a normal distribution of the data sample, where the symmetric characteristic of the distribution should ease the modelling of the analysis. Unfortunately, eigenvalues do not follow a normal distribution. As we observed in Figure 4.6, most of the eigenvalues computed are relatively small and the frequency of the observation decreases along with the increase of eigenvalues. Therefore, it would be better to define the eigenvalue distribution as a positively skewed distribution. Currently, the essential form of the eigenvalue distribution has not been fully understood. Most of the theories regarding the distribution of eigenvalues can only be supported by inductive approximations and massive computing simulations (Pastur & Shcherbina, 2011) (Liu, 2000). As a result, validating the method proposed in this section can be too ambitious and infeasible due to the ongoing debates on the newly proposed hypothesis. Besides, the unsymmetrical property of the eigenvalue distribution causes the computation of the CI to be very difficult. Determination of the appropriate CI can only be done through massive computing using the Monte Carlo method (Rubinstein & Kroese, 2016) or approximation in controversial kinds (Patil & Kulkarni , 2012). Therefore, at the current stage, we continue to recommend thresholding the eigenvalues with predefined and constant values. However, approaches based on CI can be further tested and validated in the future along with the growing understanding of eigenvalue distributions.

4.6.2. Open World vs Close World Situations

Following on the previous discussion on PCA whitening, which is aimed to reduce the effects of outliers, regarding computations of the confidence model. The concerning issues are caused by deficient coverage of the training samples, which provides no support regarding a specific prediction outcome at a certain point in the feature space.

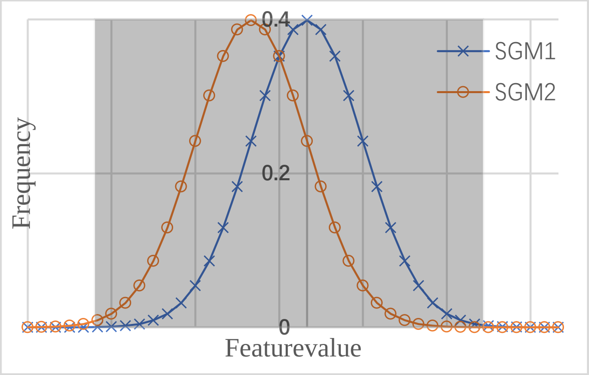


Figure 4.11 Illustration of An Ideal One Dimensional Data Sample

As Figure 4.11 illustrates, if we define $\vec{x}_{min}$ and $\vec{x}_{max}$ as the two boundaries with the near-zero support, then we would be able to derive

$$\mathrm{P}(\vec{x}|\omega_i) \approx \mathrm{P}(\vec{x}) \ \approx 0 \text{ if } \vec{x} \notin [\vec{x}_{\mathrm{min}}, \vec{x}_{\mathrm{max}}] \tag{4.25}$$

Under a close world assumption where the class labels are limited only to the labels given in the training sets, we present the conditional probability measure $\mathrm{P}(\omega_i|\vec{x})$ as Formula 3.20, where the outlier indeed results as Formula 4.25 in theory. However, under an open-world assumption (which will be further discussed in Chapter 6), there can be infinite numbers of different classes that cover different perspectives. It is never possible to obtain the complete data population to cover the open-world assumption in real life. Therefore, a practical solution is to limit the decision score to be undefined if the support from the training sample was too low to be defined. Reasonably, the range $[\vec{x}_{\mathrm{min}}, \vec{x}_{\mathrm{max}}]$ regarding the minimum support required for computation can be bounded with a relative significance level α. The function S_D regarding decision score can then the further refined as

$$\begin{cases} \mathrm{S}_D(\omega_i|\vec{x}) = 2\mathrm{P}(\omega_i|\vec{x}) - 1 & \text{if } \int_{-\infty}^{\vec{x}} \mathrm{P}(\vec{x}|\omega_i) \in [\alpha, 1-\alpha] \\ \mathrm{S}_D(\omega_i|\vec{x}) = \text{undefined} & \text{else} \end{cases} \tag{4.26}$$

Note that, checking whether $\int_{-\infty}^{\vec{x}} \mathrm{P}(\vec{x}|\omega_i) \in [\alpha, 1-\alpha]$ or not can be very costive in real-world computation especially when $\vec{x}$ has a very high dimension. Therefore, a simplified method can be proposed by observing the linear transformation of the original $\mathrm{P}(\vec{x}|\omega_i)$. In this way, $\mathcal{N}(\vec{x}|\vec{\mu},\Sigma)$ can be dominated by a minimum threshold

$$\varphi(\vec{x},\vec{\mu},\Sigma) = \min_{i=1\dots\dim\vec{x}} \mathcal{N}\left(\Lambda(\vec{x})_i \middle| \Lambda(\vec{\mu})_i, \lambda_{i,i}\right) \tag{4.27}$$

where it guarantees $\mathcal{N}(\vec{x}|\vec{\mu},\Sigma) \leq \varphi(\vec{x})$. Following this, (4.26) can be computed with ease as:

$$\begin{cases} S_D(\omega_i|\vec{x}) = 2\mathrm{P}(\omega_i|\vec{x}) - 1 & \text{if } \int_{-\infty}^{\vec{x}} \varphi(\vec{x},\vec{\mu},\Sigma) \in [\alpha', 1-\alpha'] \\ S_D(\omega_i|\vec{x}) = undefined & \text{else} \end{cases} \tag{4.28}$$

where $\alpha^{'} \propto \alpha^{\frac{1}{\dim\vec{x}}}$, since considerably $\mathcal{N}\left(\Lambda(\vec{x})_i \middle| \Lambda(\vec{\mu})_i, \lambda_{i,i}\right) \gg \prod_{i=1}^{\dim\vec{x}} \mathcal{N}\left(\Lambda(\vec{x})_i \middle| \Lambda(\vec{\mu})_i, \lambda_{i,i}\right)$ in real practice when the dimensionality is very high.

4.6.3. Precise Result Analysis in High Dimensional Space

In Section 3.4.3, we have explained the principle of the “difference to expectation(DiffEx)” measurement in detail, which requires choosing a projection method. In this experiment, a well-defined projection method such as MSD is not available for the image-based feature we derived. Since the “difference to expectation” is theoretically valid in any dimension projected, we decide to choose one of the projected PCA dimensions for analysis. Despite the projected PCA dimension can be ideally chosen randomly, it is still critical in real practice. The eigenvalue of each projected dimension indicates the variance of the data distributed. A projection on the dimension with large eigenvalues overspreads the data points, which may leave a gap when sampling and cause computational failures. On the contrary, projecting on dimensions with very small eigenvalue causes the projected data points to mostly concentrate on a small region, which again leads to computational failures. As a solution, following the eigenvalue one criterion, we finally decided to project on the PCA dimension that has an eigenvalue that is closest to 1, which is believed to be a good balancing threshold when choosing the dimension. The DiffEx analysis was conducted with 5 bins and 0.6 strides in better achieving the law of large numbers and minimising the possible computation failures. In our experiment, the pattern obtained from the DiffEx analysis was highly identical to the accuracy measure, which is understandable since the accuracy measure is considered a standardised DiffEx measure with 1 bin only. As also expected in Section 3.2, thresholding on eigenvalues indeed affects overall accuracy in the experiment. In general, it is expected that the accuracy decreases as the dimensionality reduces. However, thresholding minimum and maximum eigenvalues have shown more specific behaviours.

4.6.4. Eigen Vectors: Criteria, Precisions and Nonuniqueness

In Section 4.2, we have discussed a potential method in simplifying the MGM into UGM by applying linear projections $\Lambda(\vec{x})$ to the feature vector $\vec{x}$. We have further discussed, in much more detail, the method of projecting based on an orthogonal matrix U with SVD in Section 4.3.2, whereby U consists of the eigenvectors of the input feature set $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\}$. However, it is worth noting that U by definition is not unique, despite all the possible candidates corresponding to the same eigenvalues (but only being seen as different rotation on the projection). This nonuniqueness property of eigenvectors may potentially cause variations on the computation across different hardware and software platforms depending on the use of data type, criteria and algorithm used, which the slight difference in the projections can be magnified through later computations and eventually yields contradicting results. This issue can be partially resolved by adapting unit eigenvectors during the analysis, however, such implementation is rarely found in most of the ready implementations as we are aware of.

There are many methods that can be applied to improve the precision of the calculation. The most common and straightforward solution is to use high precision data types such as the "double" or "decimal" in C#. However, it is still possible to have round-up errors and underflows when calculating the product of independent probabilities from high dimensional UGMs. In addition to infrastructure solutions, numerical solutions such as the loglikelihood are also commonly applied in solving potential underflows, especially when multiplying very small values. It indeed improves the range of computability, however, the improvement is built by trading off the precision of the calculations. We found it has been especially difficult in applying the loglikelihood method when the values were partially being very close to 1. Apart from the loglikelihood method, we have also tried to apply numerical methods of other kinds such as the whitening method introduced in Section 4.3.2 and also calculating the fraction arithmetics separately from their significant digits. As also mentioned in Section 4.3.2, we have additionally applied the NIPALS method in further improving the precision of the projections.

Despite all the efforts we have tried to improve computation accuracy, inconsistency still inevitably exists in the testing result. We have observed marginal differences in the experiment results when testing the same proposed method with the built-in libraries in MATLAB and the Accord.Net libraries in C#, where both of them were built on well-established functions and are commonly used by the communities. However, we did observe a decrease in the level of inconsistency along with the pruning of dimensionalities, especially when setting thresholds on the minimum eigenvalues. The improvements are possibly caused by the reduction in the number of variables involved during the computations, which can be considered as an additional advantage for applying the proposed eigenvalue thresholding method.

4.7. Summary

In this chapter, we have continued our discussion on measuring the classification confidence from the last chapter and brought the problem into a high dimensional space. In the discussion, we have raised two important issues, referred to as *singularity* and *sensitivity*, which can potentially lead to computational and predictive failures when measuring the confidence in high dimensional space. In studying these problems, we have utilised the PCA method using EVD for decorrelating the original feature space into a simplified space. As a result, the projection produced by PCA does not only solve the singularity issue but also highlighted two major influencing factors of the sensitivity issue, identified as the dimensionality and the eigenvalue of the feature vector. Based on the understanding of the characteristic of our proposed confidence measure, we have defined another measure in specifically quantifying the sensitivity of the train decision score model, referred to as the *decision sensitivity* measure.

We have validated our findings with complex real-world data of high dimensionality. From experiments, we indeed found that the sensitivity of the decision score has a positive correlation to the feature dimensionality and a negative correlation to the eigenvalue of the feature used. In more specific, we found that the decision sensitivity was majorly affected by the least eigenvalue of the feature used, which inspired us to apply thresholds on the projected feature in adjusting the decision sensitivity. In general, we found that the decision accuracy and sensitivity can reach an optimal by adjusting the projected feature with thresholds. Although the optimal threshold varying from application to application, it can still be set by observing the trends of the change on decision sensitivity, where the ideal threshold of minimum eigenvalues should be positioned at the beginning of the second increase of the decision sensitivity measured. Despite the solution mentioned, we have not yet finalised the best way to adjust the projected eigenfeature, which can be further improved once the distribution of eigenvalue is better understood in the future.

Overall, we did find a potential solution for improving the decision accuracy and sensitivity of an individual decision score model. However, it is important to notice that CDSS commonly receive features of multiple kinds in real practice, which eventually results in multiple decision scores. As each one of them may result in different accuracy and decision sensitivity, it can be very challenging to integrate these decision scores predicted into a final decision outcome, which is going to be further discussed in the next chapter.

Chapter 5. Confidence in Classification Ensemble

In Chapter 3, we have introduced the concept of decision confidence and methods in measuring the confidence of a single classifier based on the Gaussian Bayes principle. In Chapter 4, we further studied decision sensitivity as an important measure for the fitness of a trained classifier. Despite we have suggested several ways in adjusting the performance of a single classifier, our experimental results seem to indicate that individual classifiers may have different performances and their limitations in reaching a maximal level of performance. Such limitations mean that using individual classifiers separately may not be sufficient for sophisticated decision making in complex feature spaces. Indeed, decision making for acute diseases can become more complicated in a real-world clinical setting. Under such a setting, the features of various kinds, such as CT, ultrasonography, blood tests and so on, may be extracted simultaneously from multiple image modalities and sources. Challenging cases of medical diagnosis are normally resolved by joint efforts from multi-disciplinary teams (MDT) where doctors from different departments provide their personal suggestions based on patient information they have at hand and then a final consensus across the board is reached. This manner of decision making allows the team to have a better understanding of the patient’s conditions from different and yet complementary angles, minimising the risk of misdiagnosis.

A simple and direct approach for integrating the different points of view is to combine or concatenate features extracted from different modalities and/or sources into a single feature vector followed by training a single classifier. However, as we extensively discussed in the previous chapter, the CDSS in this case takes the risk of becoming over sensitive due to the high dimensionality of the concatenated feature vector. To tune down such risks, an effective alternative is to build base classifiers on individual features separately and then ensemble the decisions made by the base classifiers into a final outcome. This approach seems quite straightforward, but at the same time raises an interesting question regarding how to combine the multiple decision scores made by the base classifiers into the eventual final decision score. It can be a challenging task to ensure that the combined final decision score draws a comprehensive conclusion together with a properly defined level of confidence regarding such a decision.

In this chapter, we will focus on studying different ways of combining decision scores across multiple classification evidence into a single and final decision score. In the first section, we will overview the principles and rationale behind the information fusion; explaining why and under what conditions that fusing decisions from different sources can improve the overall performance and the robustness of decision making. The second section will survey several established fusion schemes that can be potentially adapted for fusing the decision score measure proposed in this thesis. The third section will introduce a newly proposed correlation-based scheme for fusing the decision scores. The performance of the different fusion schemes will then be tested, evaluated and compared using multiple datasets. The final section will further analyse the performance of the proposed fusion scheme followed by discussions regarding further issues with multiple evidence fusion.

5.1. Principles of Classification Fusion

5.1.1. Classification Errors

An important indicator of a CDSS's matureness is its classifiers' robustness against potential generalization errors. At the core of decision making in CDSS, the performance of trained classifiers can be very much influenced by the unpredictable factors encountered in the deployment. Studying these causes for classification errors is therefore critical for the success of the CDSS.

According to (Tan, et al., 2019), classification errors are mainly caused by three factors: noise, bias and variance. Figure 5.1 uses an analogy of an artillery piece firing at a target to illustrate these three causes. *Noise* refers to the randomness embedded in the data samples that leads their values away from the truth. Such randomness might occur with the descriptive attribute values as well as the class labels. As illustrated in Figure 5.1, noise in data leads to uncertainties around the target class as if the target cannot be observed accurately from a far distance, causing the artillery piece only to be able to aim at a range around the target rather than precisely on the target. In this analogy, classification confidence can be seen as the closeness to the real target; the closer to the target, the higher the confidence level is.

In data science, noise is often related to the unpredictable environmental interruptions made on the feature data that eventually pollute the examples. For instance, an ultrasound image may contain inherent speckle noise. Any features extracted from such an image may be more or less influenced by the noise. Noise may also occur with class labels. In a typical tumour diagnosis situation, rather than labelling the tumour clearly for being benign or malignant, doctors normally give a predicted grade of either 3, 4, or 5 to indicate the level of likelihood towards being benign or malignant. Grades less or equal to 3 indicate a high chance of being benign whereas grades 5 or higher indicate a high chance of being malignant. Grade 4 is considered as borderline, where finer subgrades are often known as 4a, 4b and 4c are used to indicate the likelihood of being benign or malignant. These borderline situations tend to be where different doctors may have different opinions towards the same tumour, creating a degree of vagueness towards the real status of the tumour, a source of inter-observer variability. A noisy dataset is considerably harder to be classified correctly since the noise may lead to unexpected failures due to the uncertainty involved.

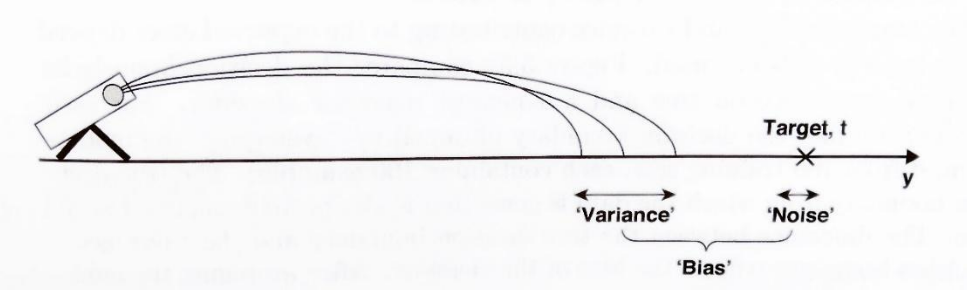


Figure 5.1. Classification Errors (Tan, et al., 2019)

Bias refers to the average distance of the classification result from the real truth, which is also known as systematic errors. In Figure 5.1, bias refers to the distance between the trajectory falling points and the target on average. The occurrence of bias is usually due to the underfitting of the model towards the training data, which cause the decision made to always shift towards a particular region from the actual target and lead to misclassification of some training examples. Such an odd normally indicates that the training is only focused on parts of the knowledge than all the information available. It may also be the case that the provided input feature data are insufficient to separate the training examples of different classes.

Biases can profoundly influence the classification accuracy in the deployment phase and should be ideally minimised. However, intentional bias may be added into the classifiers in some CDSSs to reduce the negative impacts of certain types of misclassifications. For instance, for certain types of tumours such as ovarian masses, doctors prefer classifying them as malignant than benign even the classification is a false positive. This is to prevent potential delays and missed opportunities for treatment because ovarian cancers are normally known as silent killers, meaning that the treatment can be too late if not diagnosed early in good time. However, as for thyroid lesions, doctors prefer classifying them as benign than malignant even at the risk of false negatives because the malignancy of thyroid lesions develops quite slowly due to the envelope of the lesions. Follow-up scans can provide extra opportunities to monitor the development. Therefore, an intentional bias towards either the malignant or benign class may be introduced in real-life clinics at a preferred false positive or false negative error rate. This requirement is quite specific in this particular domain of application compared to the classification of ordinary daily objects.

Variance refers to the sensitivity of the classification results to the test inputs. It is commonly measured by the variance among the output. As the example shown in Figure 5.1, the variance is reflected as the difference between the trajectory falling points at a fixed angle. Consequently, a high variance eventually leads to a less accurate classifier since it produces a broad range of expectations. Variance is usually caused by overfitting of the training data, which implies that the classifiers were over-focused on the random noise in the training data with highly flexible models and being irrespective of the actual nature of the data.

In general, the error rate ε of a classifier x can be seen as the sum of these 3 factors, which can be expressed as:

$$\varepsilon(x) = x_{bias} + x_{variance} + x_{noise} \tag{5.1}$$

Therefore, reducing these three types of error is the primary objective of classifier optimisation. In principle, bias and variance are two independent characteristics of classification errors. However, in many cases, they share a degree of negative correlation during the training of classifiers (Geman, et al., 1992), which leads to many attempts to balance the two and find the best optimal with different techniques (Domingos, 2000; Geurts, 2002). As mentioned previously, noise can also profoundly influence classification accuracy, but it can be very hard to identify due to its random nature. Often, feature selection or dimension reduction techniques can be used to reduce the impact of noise.

5.1.2. Reducing Error with Classifier Ensembles

In machine learning, classification accuracy is heavily constrained by the representativeness of the training examples. However, it is practically very hard to have a training set that reflects all the characteristics of the data at large. Therefore, bias and variance inevitably exist in trained classifiers together with random environment noise. Various methods have been proposed in the past to reduce the bias and variance contained in the trained classifiers through data sampling and classifier training in a more natural manner.

Bagging (or **Bootstrap Aggregating**) (Lee, et al., 2018) is an optimisation technique that uses repeated sampling to capture the real distribution of the data at large. In this method, the training data set has been resampled into several random bootstrap sample subsets that have the same size as the original training set but contain duplicates by using sampling with replacement. Ideally, each bootstrap sample should roughly contain 63% of the original training data, since the probability of the sampling from a sufficiently large training set Ω can be seen as:

$$\lim_{|\Omega| \to \infty} 1 - (1 - |\Omega|^{-1})^{|\Omega|} = 1 - e^{-1} \approx 0.632 \tag{5.2}$$

By doing so, each bootstrap sample set focuses on different parts of the training data and therefore reduce bias and variance contained. Learning algorithms are applied to these bootstrap samples and result in different base classifiers. The ultimate classified label is then selected depending on the majority voting by these trained classifiers. However, the success of the bagging approach very much depends on the sensitiveness of the classifiers used (Grandvalet, 2004); bagging may not improve the classification accuracy significantly with relatively stable base classifiers (Büchlmann & Yu , 2002).

Similarly, **Boosting** has been proposed as an iterative optimisation technique that also use multiple bootstrap samples with replacement to minimise the bias and variance during the training of base classifiers. In contrast to bagging, boosting assigns weights to the training samples. The bootstrap samples are generated iteratively according to a set of weights instead of random sampling, in which the weights are tuned to make the misclassified data being more likely to be selected in the next iteration of bootstrap sampling. One of the sophisticated boosting approaches are known as **Ada Boost** (**Adaptive Boosting**) (Huang, et al., 2019), where it tunes the weight W_i^{j+1} of each training example in the new boosting round dynamically by referring to the expectation of the classifier generated from the previous set of bootstrap training samples $X_i^j \equiv \{(\vec{x}_1^j, \hat{z}_1), (\vec{x}_2^j, \hat{z}_2), \ldots, (\vec{x}_n^j, \hat{z}_n)\}$, which is calculated as

$$W_i^{j+1} = \frac{W_i^j}{N_j} \times \begin{cases} e^{-\alpha_j} & \text{if } z_n^j = \hat{z}_n \\ e^{\alpha_j} & \text{otherwise} \end{cases} \tag{5.3}$$

where N_j is a normalization factor to ensure $\sum_{i=1}^{n} W_i^{j+1} = 1$, z_n^j is the classified label of the input feature $\vec{x}_n^j$ that been used to compare with the true class label $\hat{z}_n$ and α_j indicates the importance of the classifier and is measured as

$$\alpha_j = \frac{1}{2}\ln(\frac{1-\varepsilon_j}{\varepsilon_j}) \tag{5.4}$$

where ε_j denote the error rate of the classifier as

$$\varepsilon_j = \frac{1}{n}\sum_{i=1}^{n}(W_i \Theta_i)$$

$$\Theta = \begin{cases} 1 & \text{if } z_j \neq \hat{z}_j \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

Notice that the weight of each training example is set to 1 as default in the first round of boosting and α_j has a large positive value if the error rate is close to 0 and a large negative value if the error rate is close to 1. Although training classifiers can be computationally costly in this approach, it allows the classifiers to focus more on the examples that are hard to be classified and reduce the classification error significantly. Nevertheless, Boosting can be very sensitive to overfitting issues (Freund & Schapire , 1999), and the number of iterations must be carefully controlled for the robustness of testing performance.

The use of these approaches is under the assumption that we believe combining classifiers with different features complements each other and contribute to a more accurate classification result. As the outcome of each observation in the bootstrap follows a Bernoulli nature, the average error rate $\varepsilon_{ensemble}$ among all the trained classifiers based on different bootstrap samples can then be estimated by a Binominal distribution as:

$$\varepsilon_{\text{ensemble}} = \sum_{i=\lceil 2^{-1}N \rceil}^{N} \left[\binom{N}{i} \varepsilon^{i} (1-\varepsilon)^{N-i} \right] < \varepsilon \qquad (5.6)$$

Clearly, this inequation only holds when ε is less than 0.5. However, this assumption may not always hold since real-world data may contain multiple classes, causing the expectation of classification accuracy to easily fall below 0.5. Moreover, it is also acknowledged that the ensemble methods work better when the classifiers involved are sensitive, as they capture minor perturbations from the training set, which can potentially contribute to finer classification results (Tan, et al., 2019). Moreover, the ensemble methods are also known to be more suitable when the base classifiers produce variant predictions, as it utilises the potential of the joint decision-making process. However, these conditions cannot be always met in practice when applying the ensemble methods. Therefore, different strategies for combining classifiers need to be investigated in the next section.

5.2. Existing Fusion Schemes for Combining Trained Base Classifiers

5.2.1. Fusing Decisions with Standardized Probability Measures

In the previous section, we have introduced some basic concepts about combining multiple weak classifiers of different kinds in reducing online classification errors and boosting classification accuracy. However, discussions were limited to the classifiers that were trained on the same type of feature (by resampling with replacement), which may not be sufficient since various kinds of features can be derived from the same object and be used in real practice. Different types of features can cause the trained classifiers to be statistically non-identical, which naturally creates a bias towards certain classifiers. Therefore, it is essential to unify the classifiers of different nature under the same framework for a more precise classification result, where conditional probabilities and Bayesian theorem can be used as a solution.

In a typical classification scheme, any known class label ω_i predicted by a trained classifier C for a given testing observation $\vec{x}$ can be associated with a posterior probability P, where the final classified label z is nothing but the one that has the highest posterior probability among n classes, i.e.

$$\text{assign } \omega_i \longrightarrow z \text{ if} \underset{i=1\ldots n}{\operatorname{argmax}} \mathrm{P}(\omega_i|\vec{x}, C) \tag{5.7}$$

Following the same concept, each class label in a fused classification scheme of multiple classifiers can also be associated with a posterior probability, where the ultimate classified label will eventually be the one with maximised probability. Therefore, the primary objective of a probability-based fusion scheme is to approximate the posterior probability $P(\omega_i|\vec{x}, C_1 \ldots C_n)$ in different ways. Some fusion rules have been already proposed in the past (Kittler, et al., 1998) and are described next.

Product Rule

In an ideal circumstance, each one of the classifiers for fusion should be non-identical. If we assume that the classifiers are statistically independent to each other, then the fused posterior probability $\mathrm{P}(\omega_i|\vec{x}, C_1 \ldots C_n)$ will simply be the product of the posterior probability $\mathrm{P}(\omega_i|\vec{x}, C_k)$ among *R* number of classifiers. Therefore, the fused classification scheme can be seen as

$$\text{assign } \omega_i \longrightarrow z \text{ if} \underset{i=1\ldots n}{\operatorname{argmax}} \prod_{k=1}^{R} \mathrm{P}(\omega_i|\vec{x}, C_k) \tag{5.8}$$

Summation Rule

In reality, we can potentially assume that all the posterior probability $P(\omega_i|\vec{x}, C)$ are modifications based on the prior probability $\mathrm{P}(\omega_i)$, expressed as

$$\mathrm{P}(\omega_i|\vec{x}, C) = \mathrm{P}(\omega_i)(1 + \Delta_i) \tag{5.9}$$

where $1 + \Delta_i$ is a multiplier that depends on ω_i and Δ_i has a range of [0, 1]. In addition to this assumption, when $|\mathit{\Delta}_i| \ll 1$, we can then simplify the product rule as

$$\prod_{k=1}^{R} \mathrm{P}(\omega_i|\vec{x}, C_k) \sim \mathrm{P}(\omega_i)(1 + \sum_{i=i}^{R} \Delta_{ik}) \tag{5.10}$$

Following this principle, we can then derive our sum rule as

$$\text{assign } \omega_i \longrightarrow z \text{ if} \underset{i=1\ldots n}{\operatorname{argmax}} [(1-n)\mathrm{P}(\omega_i) + \sum_{i=i}^{R} \mathrm{P}(\omega_i|\vec{x}, C_k)] \tag{5.11}$$

Notice that this formula is just a linear approximation of the real likelihood, which is only valid when both Δ_i and the number of classifiers *R* is small.

Minimum Rule

In the rule (5.8), we noticed that the fused likelihood can have the minimum posterior probability of one base classifier as its upper ceiling, i.e.

$$\prod_{k=1}^{n} \mathrm{P}(\omega_i|\vec{x}, C_k) \leq \min_{i=1\ldots R} \mathrm{P}(\omega_i|\vec{x}, C_k) \tag{5.12}$$

Therefore, we can simplify the product rule into the minimum rule by taking the minimum posterior probability of a base classifier as “the best estimate”, i.e.

$$\text{assign } \omega_i \longrightarrow z \text{ if} \operatorname*{argmax}_{k=1\ldots n} \min_{i=1\ldots R} \mathrm{P}(\omega_i|\vec{x}, C_k) \tag{5.13}$$

Maximum Rule

In rule (5.11), we can further approximate the $\sum_{i=i}^{R} \mathrm{P}(\omega_i|\vec{x}, C_k)$ into R $\max_{k=1\ldots n} \mathrm{P}(\omega_i|\vec{x}, C_k)$ by only focusing on the maximum of the posterior probabilities, so we focus on the most plausible outcome only, where a max rule can then be derived under the assumption of the equal priors as

$$\text{assign } \omega_i \longrightarrow z \text{ if} \operatorname*{argmax}_{k=1\ldots n} \max_{i=1\ldots R} \mathrm{P}(\omega_i|\vec{x}, C_k) \tag{5.14}$$

Mean Rule

Using the mean as a representative indication of a dataset is well used in statistics. Similarly, we can also fuse different classifiers by taking the average of their posterior probabilities, i.e.

$$\text{assign } \omega_i \longrightarrow z \text{ if} \operatorname*{argmax}_{k=1\ldots n} R^{-1} \sum_{i=i}^{R} \mathrm{P}(\omega_i|\vec{x}, C_k) \tag{5.15}$$

Median Rule

In statistics, it is well known that the mean can be heavily influenced by outliers. In contrast, using the median as a simplified representation can be seen as a more robust method against noise, which the rule (5.15) can then be optimised as

$$\text{assign } \omega_i \longrightarrow z \text{ if} \operatorname*{argmax}_{k=1\ldots n} \operatorname*{med}_{i=1\ldots R} \mathrm{P}(\omega_i|\vec{x}, C_k) \tag{5.16}$$

Majority Rules

Majority voting is a well-adapted method for summarising outcomes from different sources. A fusion scheme with a simple majority voting rule can be presented as

$$\text{assign } \omega_i \longrightarrow z \text{ if} \operatorname*{argmax}_{i=1\ldots n} \sum_{k=1}^{R} \Theta_{ik}$$

$$\Theta = \begin{cases} 1 \quad \text{if } \mathrm{P}(\omega_i|\vec{x}, C_k) = \max\limits_{i=1\ldots n} \mathrm{P}(\omega_i|\vec{x}, C_k) \\ 0 \quad \text{otherwise} \end{cases} \tag{5.17}$$

These proposed rules have been already studied with experimental data for performance analysis (Kittler, et al., 1998). Surprisingly, the sum rule overperformed the others in general, despite being based on very strong assumptions. The authors of the paper explained this phenomenon by introducing the concept of testing errors. In online testing, the noise of various kinds will be inevitably contained in the test sample, where then the posterior probability of the classification result $\hat{P}$ can be seen as the sum of the actual posterior probability and noise as:

$$\hat{\mathrm{P}}\left(\omega_i\middle|\vec{x}, C_k\right) = \mathrm{P}\left(\omega_i\middle|\vec{x}, C_k\right) + \varepsilon_{\text{noise}} \tag{5.18}$$

Consequently, by substituting $\hat{P}$ into the proposed fusion rules, the noise may influence the classification accuracy in different magnitude. Taking the product rule that intentionally aims at precise modelling as an example, the multiplication scheme was acting as an amplifier that increases the uncertainty of the classifier excessively and results in an error rate of $1 + \sum_{i=i}^{R} \frac{\varepsilon_{\text{noise}ki}}{\mathrm{P}(\omega_i|\vec{x}, C_k)}$. On the contrary, the sum rule was relatively robust to the noise since it has a nature of strong assumptions, which provides tolerance to a certain extent, and finally contribute to an error rate of $1 + \frac{\sum_{i=i}^{R} \varepsilon_{\text{noise}ki}}{\sum_{i=i}^{R} \mathrm{P}(\omega_i|\vec{x}, C_k)}$. Therefore, we found that the preciseness of the modelling on the training data set is inversely proportional to the classification robustness in online testing. A precise modelling scheme must sacrifice online accuracy to a certain extent, which brought a real question to the research on how a classifier balances between accurate modelling and noise tolerance.

5.2.2. Adopting the Probability-based Rules

To adopt the fusion rules introduced in the last section for the proposed decision score measure, we can simply utilise the probability nature of the measure in deriving similar fusion rules in fusing decision scores. We can drive the confidence measure from the decision score and use it as a probability measure, then apply the rules. Such adoption can be applied to the product, mean and median rules. The rest of the rules are best to be applied on the decision scores as they require class labels as a factor of consideration, where maximum and minimum rules need to be applied by considering the absolute value of the decision scores. The mean and median rules can also be applied to the decision scores directly, which produce the same result as if we have applied the rules over the confidence score measure.

It is also important to note that the confidence measure does not only produce one probability measure but instead two for both the positive and negative classes. Therefore, we have proposed a new fusion rule in utilising such nature, referred to as the “Maxima” rule. The maxima rule has a principle similar to the maximum rule. The difference is that it does not only select the one with the strongest confidence score, but instead select two confidence scores from each of the strongest positive and negative confidence scores measured among all the models. The decision score is then calculated by using these 2 selected values as the positive and negative confidence measured. If we define A as a set of all defined classes, the maxima rule can then be expressed as

$$\text{assign } \omega_i \longrightarrow z \text{ if} \operatorname*{argmax}_{k=1\ldots n}\left\{\max_{i\in A}[\mathrm{P}(\omega_i|\vec{x}, C_k) - \max_{j\in A\cap i^c}\mathrm{P}(\omega_j|\vec{x}, C_k)]\right\} \tag{5.19}$$

5.2.3. Fusing Decisions with Different Priorities

In addition to the fusion methods introduced in the previous sections, fusion by assigning weights to different classifiers is also commonly used in many research works (Guan, et al., 2017; Tong, et al., 2017; Prasad & Bruce, 2008). This type of method uses weights of different kinds to assign different priorities to fuse the outcomes of the base classifiers. The weights, therefore, reflect the amount of contribution of each base classifier to the final fused decision outcome. In the last decade, many different approaches, such as utilising the accuracy measurement obtained from the validation phase, have been proposed for computing such weights (Valdovinos, et al., 2005). If we define α_k as the overall accuracy of the kth classifier (decision model), then a naïve accuracy weighted fusion can be written as

$$S_D{}'(\omega_i|\vec{x}, C_{1\ldots k}) = R^{-1}\sum_{k=1}^{R} \alpha_k\, S_D(\omega_i|\vec{x}, C_k) \tag{5.20}$$

In the expression above, α_k is acting as a prior bias to the classifier C_k, which thresholds the decision score to the nature expectation based on past experience. Note that, if the bias of each decision model is identical to each other, i.e., $\alpha_1 = \alpha_2 = \cdots = \alpha_R$, then Formula 5.20 can be rewritten as

$$S_D{}'(\omega_i|\vec{x}, C_{1\ldots k}) = \alpha R^{-1}\sum_{k=1}^{R} S_D(\omega_i|\vec{x}, C_k) \tag{5.21}$$

which yields the same decision outcome as the MEAN rule introduced in Formula 5.15 but only with different degrees of magnitude. In other words, the MEAN rule can be seen as a weighted fusion under the assumption that the accuracies of every decision model are equal to 1. In addition to the generic accuracy fusion introduced, the weight can be further refined by replacing the accuracy measures with a positive predictive value α^{+} and a negative predictive value α^{-} depending on the classification outcome as

$$\begin{cases} S_D{}'(\omega_i|\vec{x}, C_{1\ldots k}) = R^{-1} \sum_{k=1}^{R} \Theta_k S_D(\omega_i|\vec{x}, C_k) \\ \Theta_k = \begin{cases} \alpha^{+}{}_k \; if \; S_D(\omega_i|\vec{x}, C_k) > 0 \\ \alpha^{-}{}_k \; else \end{cases} \end{cases} \tag{5.22}$$

where the weight can be considered as the true positive rate, i.e., the accuracy when predicting positive and the decision model classifies the data sample as positive, and the weight can be considered as the true negative rate, i.e., the accuracy when predicting negative and the decision model classifies the data sample as negative.

5.3. Correlation (Diversity) based Fusion

In the previous two sections, we have introduced several well-established methods for classification fusion. However, these methods were originally designed in providing classification labels instead of decision scores, which have made them less suitable for fusing continuous decision scores. Despite our attempts to customize them for decision score fusions as explained in the previous section, these methods are still not utilising the numeric potential of decision scores. Most of the methods introduced previously were simply adding or selecting the scores from a set of base classifiers, and rarely has any of them considered the correlations between the decision scores to be fused. This fact motivates us to propose a new method that considers the Pearson correlations between each pair of the decision scores to be fused. By using this correlation-based fusion method, we are not only utilizing the numeric nature of the decision score, but also combining global and local information of them.

In Section 5.2.1, we have introduced the product rule based on the assumption of independent predictions by base classifiers. However, correlations may well exist between the predictions due to implicit knowledge in common. Indeed, it is well acknowledged that the correlations between the predictions from different classifiers may also play a very important role in the fusion performance (Srinivas, et al., 2009). In Formula 4.7, we have stated and proved that any multivariate Gaussian distribution can be seen as a linear transformation of a collection of *n* independent univariate Gaussian distribution. If we consider a simple case with a multivariate Gaussian distribution that contains only 2 dimensions, an equation can then be presented as

$$\lambda^2 - \left({\Sigma_{1,1}}^2 + {\Sigma_{2,2}}^2\right)\lambda - (r^2 - 1){\sigma_1}^2{\sigma_2}^2 = 0 \tag{5.23}$$

where λ is a diagonal matrix of eigenvalues of the original covariance matrix Σ and r representing the Pearson's correction coefficient between the 2 dimensions, which is equivalent to $\frac{\Sigma_{1,2}}{\Sigma_{1,1}\Sigma_{2,2}}$. In this form, we found that each pair of correlated dimension variables are bounded with a linear coefficient $r^2 - 1$ and this relationship can be inherited in higher dimensions as well, which inspired us to adopt a similar kind of coefficient for a correlation-based fusion on decision score.

As an initial design of correlation-based decision score fusion, given a set of decision score methods $\mathcal{C} = \{C_1, C_2, \dots, C_R\}$ of R classifiers, a fusion function $S_D{}'$can be introduced based on a set of correlation measurements $\mathscr{r} = \{r_{1,2}, r_{1,3}, \dots, r_{2,3}, r_{2,4}, \dots, r_{k-1,k}\}$ regarding each pair of the decision models. Two hypotheses regarding $S_D{}'$ are proposed as the guidance of correlation-based fusion.

<u>Hypothesis 1</u>:

$$S_D{}'(\omega_i|\vec{x}, \mathcal{C}, \mathscr{r} \subseteq \{1\}) = \; S_D(\omega_i|\vec{x}, C_{k=1\dots R}) \tag{5.24}$$

Hypothesis 1 aims at a special case where all the decision scores generated from each data source are fully correlated with each other. In this scenario, each one of the decision scores can be simply seen as a linear duplicate of the other, which does not further contribute to the final decision making. Therefore, the fused decision score can be sufficiently presented by adopting any one of the decision scores.

<u>Hypothesis 2</u>:

$$S_D{}'(\omega_i|\vec{x}, \mathcal{C}, \mathscr{r} \subseteq \{0\}) = \sum_{k=1}^{R} S_D(\omega_i|\vec{x}, C_k) \tag{5.25}$$

Hypothesis 2 aims at a special case where all the decision scores generated from each data source are fully independent to each other. In this scenario, each one of the decision scores can be seen as completely irrelevant to the others, which fully complements the final decision making. Therefore, the fused decision score can be seen as a summation of all the decision scores.

Following these two generic hypotheses, a general correlation-based fusion function $S_D{}'$ can be derived by combining them with the coefficient $r^2 - 1$ introduced previously as

$$S_D{}'(\omega_i|\vec{x}, \mathcal{C}, \mathscr{r} \subseteq [0,1]) =$$
$$n^{-1}\left\{\sum_{k=1}^{R} S_D(\omega_i|\vec{x}, C_k) - \sum_{j=1}^{R}\sum_{k=j+1}^{R} r_{j,k}^2 [S_D(\omega_i|\vec{x}, C_k) + S_D(\omega_i|\vec{x}, C_k)]\right\} \tag{5.26}$$

where $\sum_{k=1}^{R} S_D(\omega_i|\vec{x}, C_k)$ is the decision score under the independent condition and then being penalized by $n^{-1}\sum_{j=1}^{R}\sum_{k=j+1}^{R} r_{j,k}^2 [S_D(\omega_i|\vec{x}, C_k) + S_D(\omega_i|\vec{x}, C_k)]$ amount based on their correlations in each pair of dimensions. In this form, the penalization factor becomes

$\frac{R-1}{R}\sum_{k=1}^{R} S_D(\omega_i|\vec{x}, C_k)$ if all the base classifiers fully correlate to each other and become zero if all the base classifiers are being independent to each other, which therefore satisfy both hypotheses 1 and 2.

5.4. Experiment Results

In this section, we intend to test and evaluate the behaviours of different fusion schemes using different data sets. For this purpose, we selected two datasets. The first dataset is the Miscarriage dataset of low dimensionality and obvious correlations. The second dataset is the CBIS-DDSM breast lesion dataset, which is of much higher dimensionality and less obvious relationships among the extracted features. Both datasets have been already used for evaluation in Chapters 3 and 4 respectively.

5.4.1. Baseline Performance

In order to observe the effects of various fusion schemes, we have initially extracted multiple descriptive features, built individual classifiers based on each descriptive feature, tested individual base classifiers without the use of any fusion schemes, and then use the performance of individual classifiers as a baseline benchmark. In addition, we have also recorded the performance of a classifier built based on the simple concatenation of these features as a naïve fusion scheme benchmark. These results will be compared in the later Sections 5.4.2 - 5.4.4 in discussing whether fusion methods of different kinds can truly improve the system performance or not.

Table 5.1 Baseline Performance on Miscarriage Dataset

Features	Accuracy (%)		Decision Sensitivity	
	Mean	Std	Mean	Std
Gma	91.58	6.66	0.0287	0.0109
Gmi	93.68	4.15	0.0407	0.0157
Tma	91.05	6.10	0.0360	0.0065
F_All	97.89	2.72	0.0190	0.0057

In Chapter 2, we have introduced the miscarriage data set with its three-dimensional feature vector containing Gestational Major Diameter (Gma), Gastetional Minor Diameter (Gmi) and Transpose Major Diameter (Tma) labelled by sonographer on ultrasound images as the length, height and width of a gestational sec. For the fusion experiments, rather than combining them into a single MSD measure as we did in Chapter 3, we use these individual measurements as separate features extracted from the same input image. The detailed accuracy and sensitivity performance of using each of the individual features are listed in Table 5.1 and compared against the concatenated feature vector (F_all) with 10-fold cross-validation. As shown in Table 5.1, all three features had similar levels of accuracy that vary from 91.05% to 93.68%. Given the quite small number of observations in the dataset, the differences may not be statistically significant. Nevertheless, the results in Table 5.1 are indicative at least. Among the three base classifiers built on the individual features respectively, the classifier for the Gmi feature had the highest average accuracy and the lowest standard deviation, indicating the robustness of the base classifier. At the same time, the classifier for the Gmi feature also had the highest level of decision sensitivity, demonstrating the feature's better ability in discriminating miscarriage cases compared to the other features. It is not surprising that Gmi is the best feature among the three because Gmi primarily defines the shortest diameter of the gestational sac, which associates with the volume of the sac most.

Besides, combining the three features through concatenation has boosted the classifier's accuracy by another 4% while reducing the standard deviation by 1.5%. The increased feature dimensionality has enhanced the discrimination power of the classifier, by making it easier to find a decision hyperplane for separating two different classes. On the other hand, with the features of very low sensitivity scores, fusion by simple feature concatenation has failed to improve the modelling sensitivity aspect but made it worse, which further reduced the modelling sensitivity by more than 50%. Such a result does not seem to match our expectations. As we have discussed in Chapter 4, the increase in dimensionality should lead to an increase in the modelling sensitivity, and so the experiment findings contradict that claim. However, it is important to notice that the dimensionality change in the experiment conducted was relatively marginal, i.e., the dimensionality was only raised from one to three. The sensitivity reading might not be benefited significantly from such a small increase in dimensionality. In addition to that, the increase in the feature dimensionality has resulted in a narrower coverage of the confusion zone, which reduced the likelihood of observing the testing data with fine values, which increased the difficulty in measuring sensitivity accurately.

Not surprisingly, the sensitivities of all individual descriptive features are very low due to the very low dimensionality. This matches our expectation as explained in Section 4.2, where the degree of sensitivity measure is positively correlated to the level of dimensionality. Despite the reasons that we have already discussed in previous chapters, there are multiple explanations for the cause of these readings with very low sensitivities. Firstly, it is important to know that Gma, Gmi and Tma are simply the same type of measure on the same object but from different perspectives, which made them being similar natures with strong associations. Therefore, the feature vector we used before and after the concatenation are alike by nature. In addition to that, our proposed decision sensitivity measure heavily relies on the test data, where the hit of the readings depends not only on the distribution of the test data, but also the breadth of the confusion zone of the trained model. For the miscarriage dataset, the model trained was very much like a step function, where it has a very narrow confusion zone with sharp changes in values. Therefore, the tested data points rarely hit the confusion zone and resulted in very insensitive readings.

To further study the fusion effects, we conducted another experiment using the larger CBIS-DDSM dataset with much higher dimensional features. We extracted the GLCM and LBP features from the dataset as we did in the experiments in Chapter 4. We maintained the same parameter setting for the GLCM feature extraction, where we derived 13 Haralick's features from the GLCM in three different distances (1, 2 and 3) with four different angles (0°, 45°, 90°, and 180°), which eventually results in a feature vector of 3 × 4 ×13 = 156 dimensions. As for the LBP feature, we also maintained a similar setting for the feature extraction, i.e., using a 1-pixel radius without segmentation. However, instead of the 256-bin basic LBP histogram feature, we only used the ULBP codes (see Section 2.2.4). This is because the focus of this chapter is about the fusion of models built on features of different dimensionalities rather than studying decision sensitivity in extremely high dimensional feature space. It is commonly known (Satpathy, et al., 2014) that ULBP codes occur more frequently than the other LBP codes in an image. We eventually used all 58 ULBP codes plus one bin for all non-ULBP codes, resulting in a feature vector of 59 dimensions. As explained in Chapter 4, GLCM is effective in capturing image global texture patterns whereas LBP is good at capturing local texture patterns with a local region of the image. Having both as features will represent the texture properties of an image more thoroughly than using just one of them.

Apart from the two texture features mentioned, we have also extracted the Histogram of Gradient (HOG) feature from the dataset as a feature that focuses on regional information, such as local edges and contrast, where it extracts magnitude measurement of the distribution of intensity changes in different orientations. In this experiment, we measured HOG in 9 equally spread orientations, a customary method that is commonly used for HOG (Dalal & Triggs, 2005). Each image was segmented into a 5×5 cell matrix, where the HOG was extracted on a sliding window of 2×2 cells with 1 cell overlap. These extracted features are eventually concatenated into a super feature vector of 9×4×4 = 144 dimensions. In the end, we have also extracted the global histogram (HIST) feature from the entire image, which looked at 8 different statistical moments from the image histogram, which are mean, variance, skewness, kurtosis, energy, entropy, max value and max frequency. These 4 extracted features have provided us with good coverage from global to local information across different dimensionalities, making the dataset capable of testing different fusion methods. After the decision models are trained, the minimum and the maximum Eigen thresholds are also optimised on the training set with greedy search. In more detail, we first find the best minimum Eigen threshold that produces the highest accuracy without pruning on the maximum side, then fix the best minimum Eigen threshold found and find the maximum Eigen threshold following the same strategy.

Table 5.2 Baseline Performance on Breast Cancer Dataset

Features	Accuracy (%)		Decision Sensitivity	
	Mean	**Std**	**Mean**	**Std**
LBP(59)	72.51%	3.40%	1.000	0.000
HOG(144)	67.22%	2.68%	0.999	0.000
GLCM(156)	73.69%	3.82%	1.000	0.000
HIST(8)	67.43%	2.44%	0.964	0.012
F_All	69.79%	3.41%	1.000	0.000

As shown in Table 5.2, all features had relatively similar performance in terms of accuracy, which vary from 67.22% to 73.69%. Despite these accuracy results were not high, they are still considered acceptable as all of them were higher than the expectation of 64.05%. Among all the features, GLCM has the best accuracy. This is not a surprise since it has the highest dimensionality and higher dimensionality in general offers more discriminating power to separate the classes. On the other hand, however, higher dimensionality also brings a higher risk of model overfitting, which is unsurprisingly shown by the highest standard deviation of the feature compared to the other features. On the contrary, HIST, which has the lowest dimensionality, had the lowest standard deviation among all the features extracted. With the second-lowest classification accuracy, the experiment results have shown a very good validation of our proposed principle, where the classification accuracy is proportional to the feature dimensionality and the classification robustness is inversely proportional to the feature dimensionality.

Unlike the previous experiment with the miscarriage dataset, feature concatenation did not boost classification accuracy significantly with the CBIS-DDSM dataset. We believe this is partially affected by the relatively poor accuracy of individual classifiers, which potentially includes a large amount of interference when concatenating them. More importantly, the pruning of PCA thresholding on the concatenated features limited the discrimination power of the super-dimensional feature concatenated. This is especially harmful when the features concatenated are more naturally variant, since it is more difficult in finding representative Eigen projection in low dimensions when the input vectors are diverse. Furthermore, it is worth noting that the miscarriage dataset had far fewer training examples, which is more likely to overfit after concatenation. All these factors joined together and eventually results in an accuracy of 69.79%, which was close to the average performance among the individual features concatenated. Nevertheless, the concatenated feature still had a very high standard deviation on the test accuracy, which was 3.41% and was only slightly better than the GLCM feature. This has again shown the limitation of fusion by concatenation, which does not always improve classification accuracy and has risk in over fittings.

Regarding decision sensitivity, all features extracted were extremely sensitive. HIST was the least sensitive feature with a reading of 0.964, which was still very high given that it is only an eight-dimensional feature. A plausible explanation can be the suitability of the feature extracted. A very important factor of breast calcification analysis is the shape and spatial distribution of the calcifications. However, HIST feature does not provide any information from such domain, which potentially leads to sensitive (random) predictions. To be noticed, the reading of 0 standard deviation of decision sensitivity was simply because the sensitivities measured are extremely similar to each other. The actual standard deviation value is expected to be a very small value but none zero. After all, feature concatenation still did not improve the decision sensitivity, which again showed limitations of such a method.

The detailed confusion matrices of each feature tested can be found in Appendix A.

5.4.2. Rule-Based Fusions

As the first step of evaluating decision score fusion, we first tested the rule-based fusion schemes as listed in Section 5.2.1 on the miscarriage dataset with 10-fold cross-validation. The results are shown in Table 5.3.

Table 5.3 Rule-based Fusion Performance on Miscarriage Dataset

Features	Accuracy (%)		Decision Sensitivity	
	Mean	Std	Mean	Std
product	89.47	5.55	0.0079	0.0061
sum	96.84	3.68	0.0173	0.0099
min	86.32	6.66	0.0250	0.0058
max	97.37	3.72	0.0038	0.0033
avg	96.84	3.68	0.0092	0.0023
median	96.32	4.33	0.0183	0.0085
maxima	97.37	3.72	0.0111	0.0025

As shown in Table 5.3, most of the rule-based fusion schemes have an average accuracy that is generally superior to that of the individual classifiers, where they also had a marginally worse accuracy than the concatenation-based fusion result (see Table 5.1). Among the schemes, the minimum rule had the worst accuracy and the highest standard deviation, which is unsurprising since it considers the decision from the classifier with the weakest confidence value. The maximum rule considers the classifier with the strongest confidence value and therefore yields the best accuracy. However, it is also important to note that the minimum rule has produced the best sensitivity reading among all the tested fusion schemes including feature concatenation. Furthermore, as mentioned in Formula 5.12, the minimum rule can be seen as a simplification to the product rule since the result of the product rule is very much bound with the minimum reading among all the prediction results, which inevitably caused the product rule to produce an accuracy reading that being very similar to the minimum rule. However, unlike the minimum rule, the product rule scheme has much poorer performance on decision sensitivity, which is again understandable since we have already noticed that it can be easily affected by environmental noises. To bear evidence for this statement, the summation rule, which was marked as the most robust method to environmental noises, had much better decision sensitivity and better average accuracy. Despite the advantages and disadvantages discussed in Section 5.2.1, the median rule scheme still seems to have the best overall performance with fairly good accuracy and enhanced decision sensitivity. It is still the best scheme among all the tested schemes even the decision sensitivity of the scheme is lower than the individual classifiers and feature concatenation.

To further study the behaviour of rule-based fusion schemes, we conducted the second experiment using the CBIS-DDSM dataset. The test results are shown in Table 5.4.

Table 5.4 Rule-based Fusion Performance on Breast Cancer Dataset

Features	Accuracy (%)		Decision Sensitivity	
	Mean	Std	Mean	Std
product	65.40	2.02	0.287	0.190
sum	73.16	3.62	0.608	0.187
min	66.90	1.86	0.797	0.081
max	71.66	3.70	0.136	0.113
avg	73.16	3.62	0.389	0.076
median	73.37	3.30	0.471	0.067
maxima	71.50	3.60	0.522	0.138

Many variations in accuracy across different rule-based fusion schemes can be observed. The sum, average and median rules have a similar performance to that of individual classifiers and a marginally better accuracy to the concatenation-based fusion. The minimum rule and product rules were again having relatively poor accuracy but at the same time lower standard deviations among the 7 rules. It is important to note that all the schemes have an improved decision sensitivity reading compared to those of individual base classifiers and that of concatenation-based fusion. Nevertheless, the product rule and maximum rule seem to have a sensitivity that was overly enhanced towards the minimum side, which can be too insensitive. The sum, average and median rules had the most outstanding performance among all the rules we have tested, which have improved the overall classification accuracy by almost 4% compared to fusion by feature concatenation. The common factors between these three rules are that they all looked at the average performance between each classifier rather than only considering the strongest or the weakest among the classifiers fused. So, the fusions make the final decisions being robust to environmental noises and potential overfitting. As a positive reflection, all sum, average and median rules had a much sensible decision sensitivity reading comparing to the other fusion methods we have implemented so far. The decision sensitivity readings of these three rules were around 0.5 and considerably close to the ideal reading, where each of the projected dimensions is expected to follow standard normal distribution on average. The median rule had the best overall performance compared to all the fusion rules we have tested, which did not only have the best accuracy but also was most close to 0.5 on decision sensitivity. However, it is worth mentioning that the median rule still did not overperform the best individual classifier, which was the classifier built on GLCM features with an accuracy of 73.69%. Nevertheless, the median rule is considerably being better since the difference of only 0.3% may not be significant enough in marking the strength or weakness but the decision sensitivity was certainly improved greatly.

5.4.3. Weight-Based Fusions

We have also tested the weight-based fusion schemes as described in Section 5.2.3. Similarly, these schemes were firstly tested on the miscarriage dataset with 10-fold cross-validation and the results are shown in Table 5.5.

Table 5.5 Weight Based Fusion Performance on Miscarriage Dataset

Features	Accuracy (%)		Decision Sensitivity	
	Mean	Std	Mean	Std
W_Acc	96.84	3.68	0.0083	0.0019
W_PVs	97.37	3.72	0.0093	0.0018

The experiment results in Table 5.5 show that fusion has improved classification accuracy overall and also produced more robust classification results. More specifically, both of the weight-based fusion methods produce similar results, which increase overall accuracy by at least 3% compared to individual base classifiers and further reduce the standard deviation of the accuracy to 3.7. The weighted fusion by using PVs has produced slightly superior results because the method is more customised at an individual class level instead of overall accuracy, and therefore produces a finer result. However, the difference was not significant enough to firmly conclude that weight-based fusion by using PVs is better than the benchmark performance. Besides, both of the tested weight-based fusion methods also have similar decision sensitivity readings. But the decision sensitivity performances were much inferior compared to that of individual classifiers only or fusion by feature concatenations.

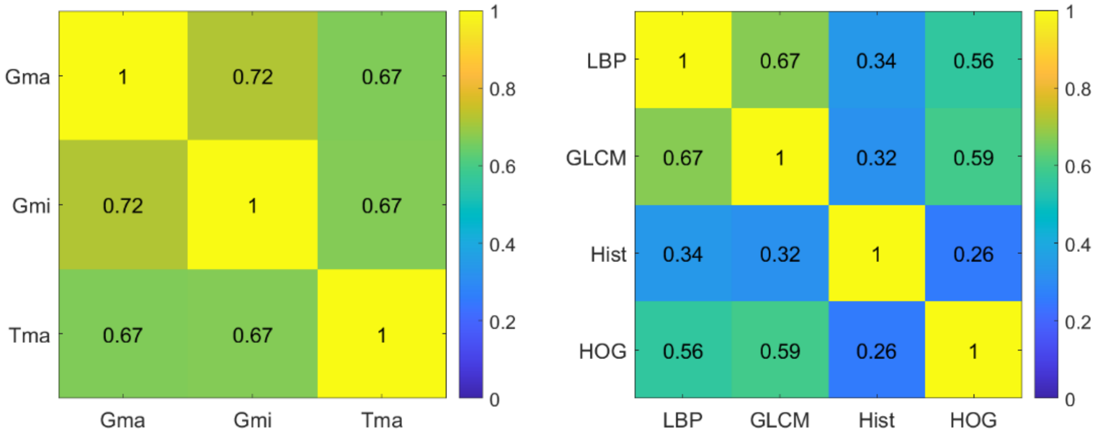
As before, we have also tested the weight-based fusions on the CBIS-DDSM dataset, and the results are presented in Table 5.6. Both fusion methods by using overall accuracy and PPV/NPV as weights have again showed similar accuracy results, which was very close to the accuracy performance of individual classifiers and being superior to fusion by feature concatenations. In contrary to the previous experiment, the weighted fusion has a slightly superior accuracy this time compared to the PPV/NPV weighted fusion. However, the difference was still not significant enough to conclude which one is better. The very similar outcome between these two methods is due to the almost identical PPV and NPV of the trained model, which was 0.8 and 0.77 on average for benign and malignant in respective. The almost identical bias causes the adjustment to be marginal and very close to the weights by using overall accuracy only. We believe the balanced bias was due to the balanced benign/malignant ratio and the automatic feature pruning with PCA thresholding.

Table 5.6 Weight Based Fusion Performance on Breast Cancer Dataset

Features	Accuracy (%)		Decision Sensitivity	
	Mean	Std	Mean	Std
W_Acc	73.16	3.39	0.316	0.066
W_PVs	73.05	3.40	0.391	0.078

5.4.4. Proposed Correlation-Based Fusions

In our next experiment, we evaluate the proposed correlation-based fusion scheme on the two selected data sets respectively. In deriving the correlation matrix as required by the proposed scheme, we have tested the computed decision score model on the training set and recorded the correlations of the predictions between the base classifiers. Although it is recommended to use a separate validation set in deriving the correlation matrix, we decided to use the testing set as the validation due to the insufficient number of data examples. The correlation matrix is derived without referring to any class labels. The extracted correlation matrices are shown in Figure 5.2.



(a) Decision Correlations on Miscarriage Dataset *(b) Decision Correlation on CBIS-DDSM dataset*

Figure 5.2. Correlations Across Decision Scores of Base Classifiers based on Different Extracted Features

As explained before, the three descriptive feature variables from the miscarriage dataset are various diameter measurements of the same object, i.e., the gestational sac, where the sac has an oval shape by nature. Consequently, it is expected that the decision scores predicted are correlated since the three dimensions of the sac grow proportionally to the size of the sac. Indeed, as Figure 5.2 shown, there have been fairly strong correlations exist between the decision scores predicted, which was about 0.7. The Gma and Gmi had greater correlations compared to the correlations to Tma, which is again understandable since both Gma and Gmi were measured from the same plane and Tma was measured from the transpose plane (on another image). Based on the correlation matrix measured, the miscarriage data set can be very good in testing the performance of the correlation-based fusion method where the decision scores predicted were highly correlated.

Unlike the miscarriage dataset, extracted features in the CBIS-DDSM dataset have shown various degrees of decision score correlations when they are used for classification. Among the four types of features we have used, the decision scores for the LBP feature and the GLCM feature are most correlated with a correlation coefficient of 0.67, which can be considered relatively strong. We believe the strong correlation measured was due to both of the features are focusing on similar local textures, which caused the classifiers to learn information of similar kinds. Comparatively, the HOG feature, which looks at regional features from a more global perspective, has much lower decision score correlations than those for LBP and GLCM.

However, the HOG feature has a slightly higher decision correlation to the GLCM feature than the LBP feature. This may well be because the GLCM feature is using statistic moments from the co-occurrence matrix, which made the feature vector focuses less on the local textures and hence closer to the HOG feature characteristics. Consequently, the histogram feature, which looks at statistical features from an entirely global perspective had the least correlation to the other three types of features. In general, the features we extracted can be ranked from local to global as LBP, GLCM, HOG and Hist. The measured decision score correlations reflect such a corresponding ranking.

To further develop some understanding of the effect of correlation-based fusion, we plot the predicted decision scores against the input feature values on the miscarriage testing set in Figure 5.3. Plots (a) and (b) present the trends of the original decision scores, which are based on an abstract feature MSD and a multivariate feature respectively. Plots (c) and (d) present the trends of the fused decision scores from each feature, which is based on the mean fusion rule and the correlation-based fusion scheme respectively. Compared to the original feature, the mean fusion scheme has enlarged the confusion zone, which can be considered as a compromise to the worst-performing classifier. Interestingly, the correction-based fusion has shown a clearer linear trend compared to the others, which reflects that the penalty introduced was indeed acting as an orthogonalization/decorrelation method. Furthermore, the correlation-based fusion has shown remarkably high confidence in classifying miscarriage cases. Although a small part of the PUV cases was misclassified with relatively large variations, this is tolerable because it may indicate the highlighted PUV cases may eventually evolve into real miscarriage cases.

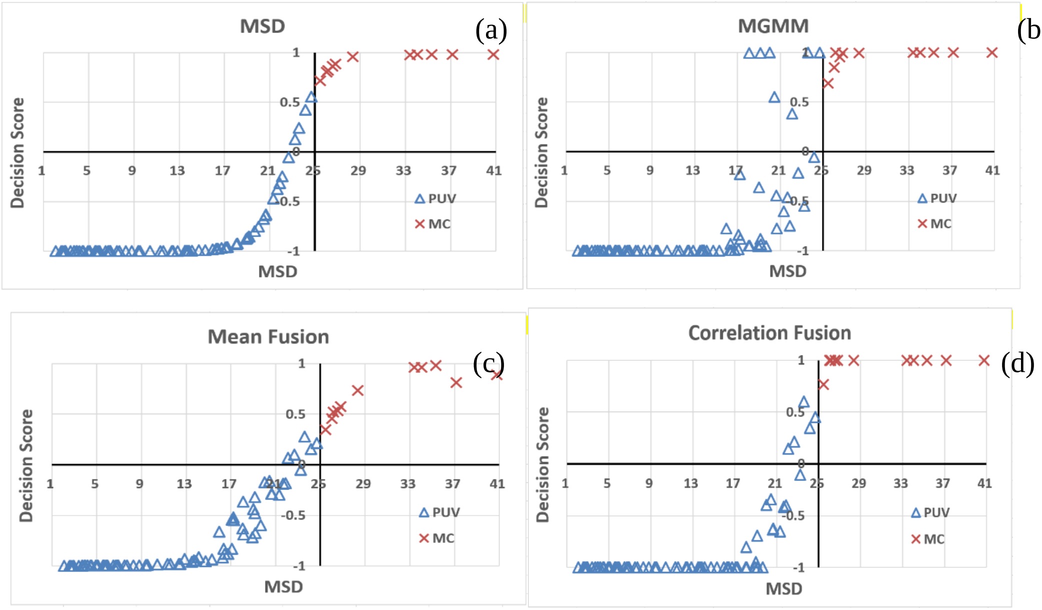


Figure 5.3 Predicted Decision Score at Different Feature Value

We have also evaluated the performance of the correlation-based fusion scheme with 10-fold cross-validation like we did for the other two kinds of fusion (i.e., simple rules and weighted fusions). The test results on the miscarriage data set are shown in Table 5.7. The figures show that correlation-based fusion has indeed improved classification accuracy. Although the accuracy was slightly worse than the fusion result by feature concatenations, the difference in prediction accuracy is not significant enough to conclude such a statement, and the decision sensitivity was indeed better than the concatenation-based fusion. In addition, it is important to note that the difference in decision score sensitivity can be greater than it looks since the sensitivity function follows sigmoid characteristic, which the slope is approaching zero when the sensitivity reading is approaching 0 or 1 and therefore the range [0.019, 0.02] certainly covers a larger domain than what [0.499,0.5] covers.

Table 5.7 Correlation Based Fusion Performance on Miscarriage Dataset

Features	Accuracy (%)		Decision Sensitivity	
	Mean	Std	Mean	Std
r^2	96.84	0.02	3.68	0.01

Finally, the figures in Table 5.8 show that the correlation-based fusion scheme also achieves relatively good classification accuracy for the breast cancer dataset. The accuracy was significantly higher than the fusion result by feature concatenations but was slightly worse than the best base classifier on individual features, i.e., the base classifier for GLCM. However, the difference is again quite marginal. More importantly, it is worth noting that the correlation-based fusion provided a much better decision sensitivity than using any individual classifier or fusion by feature concatenation, which greatly refined the highly sensitive decision inputs while maintaining good accuracies.

Table 5.8 Correlation Based Fusion Performance on Breast Cancer Dataset

Features	Accuracy (%)		Decision Sensitivity	
	Mean	Std	Mean	Std
r^2	73.26	3.32	0.799	0.073

5.5. Fusion in Practice: Discussion & Future Refinement

In Section 5.4, we have evaluated the accuracy and sensitivity of different fusion schemes. We have found that fusion, in general, helps in improving the overall accuracy and robustness of the decisions made. However, it is also useful to understand the impact of the fusion on individual cases in practice. These kinds of considerations are expected to shed some light on how to improve our proposed correlation-based fusion scheme.

5.5.1. Effect of Fusion in Decision Making: Case Studies

Although the performances presented in Section 5.4 are informative, it remains interesting to understand how and why the predicted decisions change when fusions are applied for individual practical cases. Unfortunately, the early pregnancy dataset has a relatively simple characteristic and does not provide us with enough distinctive features. Although we did extract features of various kinds from the CBIS-DDSM dataset, these image-based texture features are difficult to understand from a clinical knowledge point of view. Therefore, we will use another dataset with more explainable features specifically for supporting this discussion. The dataset was obtained from International Ovarian Tumour Analysis (IOTA) Group (Dirk, et al., 2010). Named IOTA Ovarian Dataset in this study, it contains a total of 242 grayscale 2D ultrasound images. Each image is accompanied by the pathology label. In total, there are 138 images of benign and 104 images of malignant masses. Out of the 242 cases, 239 cases are also accompanied by patient ages, ranging from 14 to 88 years old. Using this age descriptor enables us to examine decision fusion on a more meaningful and explainable footing.

We have first built a univariate Gaussian Bayes model on the training set by using the patient age as the feature. Figure 5.4 shows the decision score (Y) measured at different ages (X) when the trained model is applied to the test examples. A positive measurement indicates a prediction of a malignant mass, whereas a negative measurement suggests a prediction of a benign mass.

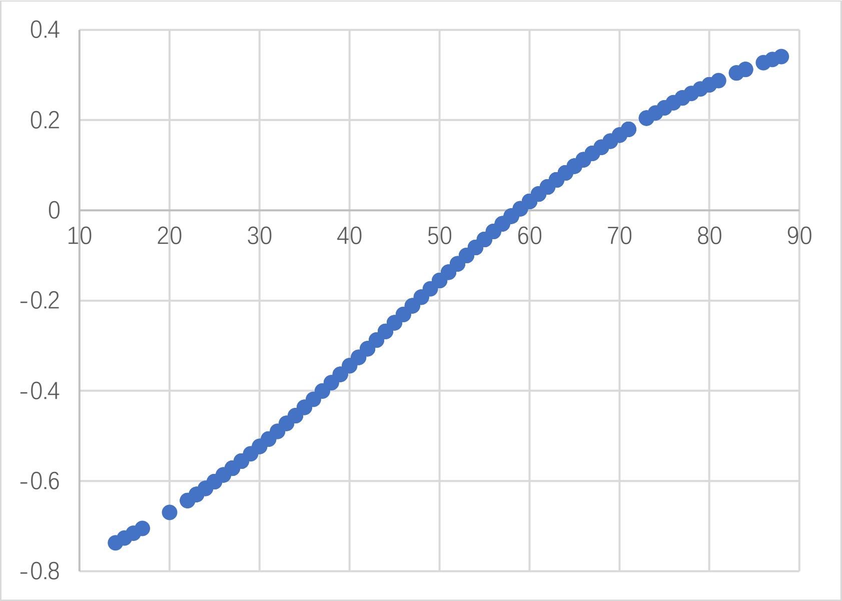


Figure 5.4 Change in Decision Score at different age

The figure shows that the level of malignancy of ovarian masses increases proportionally along with the increase of patient age. The prediction of the tumour status flips from benign to malignant when the patient age reaches about 58 years old, i.e., towards the end or shortly after the period of menopause for most women. Menopause is a natural part of ageing that usually occurs between 45 and 55 years of age, with the average age for menopause at 51 in the UK (NHS Trust, 2018). Studies have already shown that postmenopausal women are more likely to have more invasive tumours than premenopausal women (Moorman, et al., 2008), because the function of the ovary decays with the ageing and eventually reaches the minimum after menopause (other factors such as the age of pregnancy may also have an impacted). The findings from the figure coincide with the known facts very well. Besides, we found that decision confidence obtained was much stronger for the young patients (peaked at 0.75 at age around 14) than that for the old patients (peaked at 0.35 at the age around 88). As another fact, this was mainly caused by the latency of cancer, where cancer normally needs 10–20 years to develop, from the initiating event until the disease appeared clinically. These facts have again demonstrated the validity of our proposed decision score measure. However, despite the strong correlation between the age and decision score, the level of accuracy using the age model is not very high (see Table 5.9 later).

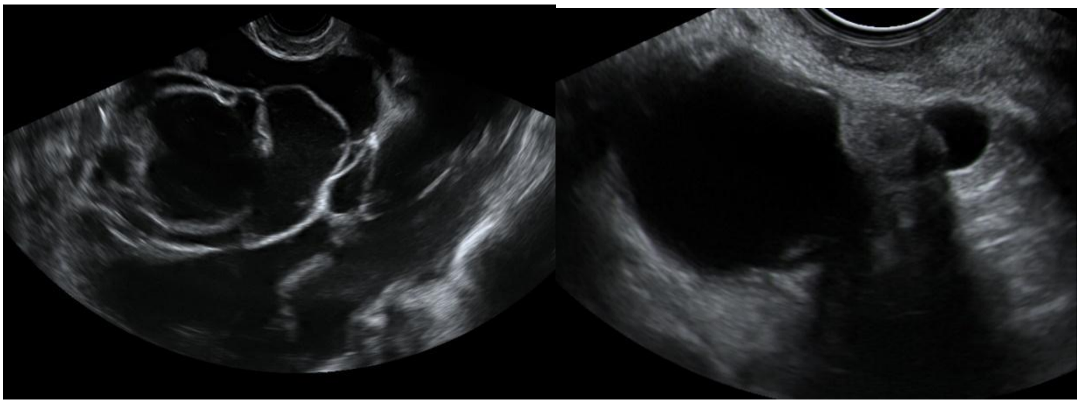
To investigate the fusion of classifiers, we adopted an effective image-based feature specifically intended for ovarian messes, known as Fast-Fourier based Geometric Feature (FFGF), extracted from the frequency domain of the ultrasound images (Al-karawi, 2019). The extraction of the feature is briefly explained as follows. First, the original ultrasound image was pre-processed by using an adaptive block-based Wiener filter. The pre-processed image was then converted into the frequency domain by using the Fast-Fourier Transformation (FFT) with adjustment on intensity levels. The adjusted FFT spectrum image was then binarized using the minimum cross-entropy followed by morphological operations to refine the outcome. These processes eventually produce a Boolean mask of elliptical shape at the centre of the FFT spectrum image, representing energy change patterns in the original ultrasound image. It has been observed that the malignant tumours normally have larger and “fatter” elliptical shapes, indicating the spread of energy variations in ultrasound images caused by complex compositions of tissue structures within malignant tumours. The benign tumours tend to have smaller and “slimmer” elliptical shapes, indicating the uniformity in energy spreads. Consequently, the major and minor diameters of the ellipse in the FFT spectrum image together with its area form a three-dimensional FFGF feature vector. Based on the two kinds of features extracted (age and FFT), decision score models were trained and tested with 10-fold cross-validation. The test classification accuracies by the base classifiers built on each feature as well as the fusion accuracy of the two classifiers using the mean fusion scheme are shown in Table 5.9.

Table 5.9 Test Accuracies on the IOTA Ovarian Dataset

	AGE	**FFGF**	**Mean Fusion**
Malignant	48.1%	82.7%	77.9%
Benign	68.9%	63.0%	71.1%
All	59.8%	71.5%	74.1%

The test results showed that age alone, with an overall accuracy of 59.8%, was not an optimal feature for classifying the malignancy of ovarian tumours. It is worth mentioning that the accuracy in classifying malignant (only 48.1%) cases is much lower than that in classifying benign cases (68.9%), where the level of accuracy is only marginally higher than a purely random guess (***E***[*malignant*] = 43%). This is very understandable since it is rather irresponsible in classifying the tumour malignancy by only referring to the age of a patient. However, age can still be quite effective at classifying benign cases (nearly 12% higher than a random guess (***E***[*benign*] = 57%) because of the latency as mentioned before. The results have also again demonstrated the soundness of our proposed measure of the classification confidence, where the confidence is measured very low for the malignant cases but relatively high for the benign ones (see Figure 5.4). Compared to the age, the FFGF feature has a much better performance with an overall accuracy of 71.5%. The feature is especially good in classifying malignant cases with a true positive rate of 82.7%. However, the performance in classifying benign cases was still not very satisfactory, poorer than that for the age. It seems that the FFGF features capture more image information on energy variations for the malignant cases, which made it much easier in classifying malignant cases than benign cases. By combining the decisions made using the mean-rule fusion, we have successfully improved the overall accuracy to 74.1%. More importantly, we also have a significant improvement in the classification of benign cases up to 71.1%, higher than both classifiers built on age and FFGF features separately. Despite a marginal decrease in the accuracy in classifying malignant cases by about 5%, the fused decision can be considered as less biased and more robust in classifying both classes. These experimental observations have again demonstrated the soundness of the fusion approach.

Finally, as the main interest of this discussion, we have particularly investigated decision/classification changes before and after the decision fusion. As reported, the overall accuracy of the decision model trained using age and FFGF are 59.8% and 71.5% respectively. Among all the decisions made, the two models have 47.7% (114/239) disagreement between them. However, this high percentage of disagreement does not mean that all related confidences are strong enough to alter the decisions made. In fact, only 15.8% (18/114) of the decisions made have been changed after fusing the age model's predictions with the FFGF model's predictions. Among these altered cases, 66.7% (12/18) show the corrected classification outcomes.



(a) (b)

Figure 5.5 Ultrasound Ovarian Scans of Patients of Different Ages (a) 20 years old (b) 69 years old

We then purposely select two examples of the ultrasound images of ovarian masses involving decision changes before and after the fusion and show them in Figure 5.5. The pathology outcome of images (a) and (b) were both benign. However, image (a) has been firstly classified as malignant based on the FFGF feature with a decision score of 0.623 but changed into -0.046 after applying the mean fusion, correcting the case to benign but with very low confidence. This has matched the decision made by the doctor as a borderline case (note that the IOTA Ovarian dataset also records the doctor's predictions besides the pathology result of the tumours). On the contrary, image (b) has been firstly classified as benign using the FFGF-based model with a decision score of -0.082 but changed into 0.036 after the mean fusion, resulting in a misclassification with very low confidences. It is worth mentioning that we presented these two cases to a medical consultant with many years of experience and expertise in ovarian mass diagnosis (the meeting was online and took a place on 21st/Jun/2018). The consultant commented that he would very much agree with the fusion result for these cases by considering the patient age besides the inspection of the ultrasound images in clinical practice.

5.5.2. Refining the Proposed Correlation Fusion Scheme

In Section 5.3, we have introduced a fusion method based on decision correlations between pairs of classifiers. We consider four different scenarios under which information on decision correlation can become useful in guiding the right strategy for decision fusion:

- Scenario 1: fully correlated classification results

  If all classifiers are predicting the same class labels at all times, as the agreed classification outcomes among different classifiers are considered as ordinary events, fusion should offer no added value in boosting the prediction probability. This is the same as what we have introduced in the Formula 5.24. In this scenario, only one of the base classifiers should be sufficient for the decision making and saving time for classification.

- Scenario 2: correlation happens on correct classification outcomes only

  If the base classifiers share a good commonality in the correct classification outcomes, but each classifier makes different mistakes, then, it results in a positive correlation in the correct classifications and independent misclassifications. In this scenario, we should obtain higher confidence if multiple classifiers have the same predictions since matched results further enhance the correct predictions.

- Scenario 3: correlation appears on misclassifications only

  On contrary to Scenario 2, if base classifiers make highly correlated predictions in misclassifications, but each classifier produces correct predictions independently, which then results in a positive correlation in the miss classifications but being independent in the correct classifications. In this scenario, we should reduce the confidence obtained if the classifiers yield similar results since correlated predictions tend to become common mistakes.

- Scenario 4: fully independent

> If all features are nearly complementary, i.e., their overlapping predictions are rare, but each of the classifiers can be highly trustable. Then, we should consider each of them as an equivalent information source and fuse them in an unbiased way (as introduced in Formula 5.25).

Following the analyses of the different scenarios, our fusion scheme based on the correlation between decision scores can be modified or tuned in the manners as summarized in Table 5.10.

Table 5.10 List of Fusion Strategies based on Different Correlation Scenarios

Fusion Scenario	**Correlation Scenario**		**Fusion method**	**Description**
	Correct classification	**Miss classification**		
Match	Independent	Independent	None	No bias
	Independent	Positive	Penalty	More likely to be wrong
	Positive	Independent	Reward	More likely to be correct
	Positive	Positive	Max/Min	Choose one
Mismatch	Independent	Independent	None	No bias
	Independent	Positive	Reward	More likely to be correct
	Positive	Independent	Penalty	More likely to be wrong
	Positive	Positive	Max/Min	Choose one

As Table 5.10 shows, the fusion outcome of independent classifiers can always be fused with mean or product rules, which is the expectation under the independence assumption. On the other hand, fusions on correlated classifiers should depend on their relationships. In the case where both correct and misclassifications are correlated, the prediction from each classifier is considered as potential duplications. In such cases, it is understandable and reasonable to take the decision score with the maximum confidence since the highest confidence provides the most guarantee on correct classifications. However, it may also be good to take the decision score with the minimum confidence, since it is much safer to present the worst case. At last, for the classifiers that have different correlations on the correct and misclassifications, different penalty and reward schemes as listed in Table 5.10 can be applied for tuning the final fusion decision score to an appropriate value. In general, the fusion strategy, in this case, should depend on the purpose of applications, which may need more discussion in the future.

Following similar principles, some other researchers have introduced an indicator as a weight for tuning the fusion outcomes by using the ratio between recall r and false positive rate q of each classifier, where $r > q$ (Pochampally, et al., 2014). However, their method was based on a conventional classifier ensemble, which cannot be directly applied to our decision score measurements. However, developing a similar method based on decision scores can be very interesting and should be further researched in the future.

5.6. Summary

In this chapter, we have first elaborated the reasons behind fusion schemes and how fusion may benefit the CDSS by potentially improving the accuracy and robustness of the decision made. In more detail, we have summarized two schools of thought for fusing the decision scores derived from multiple sources based on probabilistic normalisation or adding additional weights. However, these commonly used methods do not suit the nature of the decision score measure perfectly. Therefore, we have proposed a correlation-based fusion method in maximising the potential of the decision score measure. In the experiment, we found that both the median rule and our proposed correlation-based fusion had very good performance under different scenarios. We are not certain yet to conclude which one performs the best, but these two methods have definitely outperformed the other methods tested. Future research may aim at refining the correlation-based fusion according to more specifically defined fusion scenarios in bringing up the fitness of the fusion method under different conditions.

In general, we found that fusing decision scores may or may not always improve classification accuracy, but it can certainly improve the decision sensitivity of the CDSS and produce more robust predictions. This makes the fusion scheme especially useful after the CDSS has been deployed, as it greatly reduces the risk of false diagnosis caused by over/under fittings. However, the clinical environment can be very unpredictable and face changes constantly, where new variants of diseases can be found from time to time with frequent updates on the diagnostic standards and protocols. Despite the fusion can offer a certain level of robustness, unfortunately, it still very much works within environments that are similar to the training data only, which may not be useful anymore after the testing environment changed. Therefore, it is essential to further study on potential methods that refine the trained models after the deployment of the CDSS, which will be further discussed in the next chapter.

Chapter 6. Maintaining Confidence in Deployment

In previous chapters, we have investigated plausible ways for measuring decision confidence under different scenarios. In reality, the clinical environment is constantly facing changes as a result of new acute unseen cases encountered, the evolving knowledge and understanding gained often result in more effective medical treatments (Park, et al., 2012). As a good example, the Coronavirus disease 2019 (COVID-19) was first discovered at the end of 2019 and soon became a global pandemic. After 2 years of spreading, COVID-19 was evolved into many variants from the initial Alpha variant (B.1.1.7) to the latest Omicron variant (B.1.1.529), where each one of them had different characteristics and required different treatments (Tregoning, et al., 2021). Therefore, being adaptive to unforeseen circumstances plays an essential role in a CDDS, which requires the CDSS to be elastic on the diagnostic strategy used and be able to make adjustments to the diagnostic strategy in a timely and efficient manner. However, one of the shortages of many existing works in classification including the previously proposed measures is that the modelling heavily relies on the training set, which means the trained model has been built on historical known cases and may well not work for any new cases that have never been encountered before. With the increase of such new cases, the trained model may eventually become obsolete after a certain period of time. Consequently, a self-motivated online learning scheme needs to be introduced to update the model constantly after the model deployment.

- There can be mainly two approaches for accommodating such an adaptation capability in CDDS. The most straightforward solution is to perform complete retraining to the decision score model using the old training data combined with the new observations. However, this retraining process can be very costly and interrupt the routine functioning of the model. The interruption is caused by the fact that most of the model retraining (particularly deep learning models) is a computationally intensive undertaking. An alternative solution is to periodically adjust the functional parameters to achieve robust performance with better efficacy. As tuning these parameters is comparably a lighter computational burden than retraining the entire model, it prevents the system from constantly retraining itself and greatly minimised the potential overheads and interruptions involved. However, this alternative cannot replace the model retraining completely as simple parameter tuning may not be sufficient towards adapting all variants, and hence comprehensive retraining of a robust and optimal updated model is still necessary and unavoidable. Therefore, as the final part of this thesis, we will attempt to further investigate the issues underpinning the following two main system requirements:

- How to monitor and assess the system performance regularly to evaluate the reliability and usability of the diagnosis strategy and how to apply adjustment schemes spontaneously according to the assessment results.
- How to and when to modify the features and classifiers used to boost up system performance according to the nature of the new observations in a cost-effective manner.

In this chapter, we are going to discuss these two requirements in detail and look at several potential solutions for monitoring and altering the decision score measures after the deployment. More specifically, the chapter is structured into the following sections. Section 6.1 introduces several commonly used metrics for monitoring system performance. Section 6.2 reviews several classical schemes that spontaneously update classification models in maintaining the model's robustness. Based on the understanding of these classical schemes, Section 6.3 proposes several plausible schemes for refining the decision score measures spontaneously with relevant discussions. Section 6.4 evaluates these proposed schemes through an experiment over a specially created dataset that simulates new arriving observations over a period of time. Section 6.5 summarises the findings and further analyze the findings through discussions.

6.1. Performance Evaluation

6.1.1. Functional Efficiency

Functional efficiency refers to the direct cost in performing and completing an operation in the CDSS, which can be mainly measured by computational cost and human effort spent. The computational cost relies heavily on the efficiency of the decision algorithm and the data structure representing and storing the designed model. As a commonly used metric, such time complexity can be presented by using the big O notation, which is a mathematical measurement that describes the limiting behaviour of a subject when it approaches a particular value. If a non-negative function *f* is defined as the complexity of the decision algorithm of any given size of raw data inputs *x*, then another nonnegative function *g* can be defined as the efficiency measure of the *f* by using big O notation as

$$O[g(x)] = f(x)\ as\ x \to n \tag{6.1}$$

where g asymptotically dominates *f* at a particular data size n. That is, the computational cost when the size of the data is n. Consequently, a less efficient decision algorithm shares a high $g(x)$, which indicates a high cost in computational time and system resources at the given the data of size n. The human effort is the time and labour cost in developing and maintaining the CDSS, which is normally defined as PM (Person-Months). The human effort cost in system development and maintenance are normally sharing a negative relationship, where a mature system ideally expects more effort paid in development but consequently requires less effort paid in maintenance.

These two types of costs mentioned are in fact sensitive and strongly related to the dynamic changes in the clinical environment. The expansion of the database due to the continued accumulation of the new knowledge obtained can increase the scale of the problem, i.e., increase in the value of *n*, which diminish the functional efficiency of the system and consume more resource to refine. Therefore, the functional algorithm *f* needs to be efficiently designed to achieve a tolerable g along with the continued growth of n. Alternatively, filtering processes can be periodically applied to the data in controlling the *n* within a reasonable size, which ensures the functional efficiency g remains a tolerable constant. Furthermore, as mentioned, the growth in the scale of the problem can also put a heavy workload on maintaining the system, which makes the spontaneous tuning of the modelling parameters more appreciated.

6.1.2. Diagnosis Accuracy

In previous chapters, we have already introduced and used accuracy measures for evaluating the performance of our proposed measures. Besides, the accuracy measures can also be used as a good indication of the potential cost that the CDSS may suffer. Such cost mostly raises serious concerns, as inaccurate diagnoses can result in mistreating the patients and potentially lead to death in the worst scenario, which naturally weights this factor with the highest priorities when evaluating system performance. In most cases, the cost of other criteria (such as the functional efficiency we discussed in the last section) can be sacrificed to a certain extent to ensure the potential impact from inaccurate diagnosis is minimised.

Different methods can be used to evaluate the overall decision accuracy, where the most common method is to calculate the average classification accuracy regarding the test samples, which is calculated by using the total number of the right classified labels divided by the total number of test samples. However, certain labels can be more desirable than the others in clinical diagnosis to accomplish damage control (as the example given in Section 4.1: bias). This kind of bias is not well reflected with simple overall accuracy. Therefore, the true positive rate or true negative rate base on the preference of the class label is also commonly used to measure the diagnosis accuracy instead of using the overall accuracy, which is calculated by focusing on the amount of the right classified desirable labels divided by the total number of the desirable labels. These methods can be used for monitoring the performance of the CDSS when encountering new observations in the deployment phase.

The accuracy measures mentioned above can provide good information on whether a trained model is obsolete or not, however, it cannot provide further evidence on whether the model can be reusable again in the current environment after minor adjustments. Therefore, it is also important to assess the general fitness of the design model to the new observations in determining a suitable strategy for refinement. As a solution, the ROC (Receiver Operating Characteristic) curve (Matjaž & Zoran, 2011) is commonly used for examing the general fitness of a model designed, which is a plot of the true positive rate against the false positive rate under different settings. The reliability of the test model can then be determined by the AUC (Area Under the ROC Curve) as

$$\int_{-\infty}^{\infty}[\int_{x}^{\infty} f_0\,(x) \int_{x}^{\infty} f_1\,(x)]\,dx \tag{6.2}$$

where f_0 and f_1 are two functions regarding the true positive rate and false positive rate at a chosen parameter x respectively. In common understanding, as illustrated in Figure 6.1, the test model is considerably least useful when its AUC equals 0.5 and most useful when AUC is close to 1. Indeed, when AUC equals 0.5, the model does not show any prediction power to the changes in the parameter settings as it always predicts randomly. On the contrary, the optimal parameter setting can be selected with ease when AUC equals 1, as a slight change can reduce the false positives significantly while maintaining very good accuracy in true positive predictions. Therefore, by simply referring to the AUC measure, we can determine the appropriate strategy for refining the models, where simple tuning on the parameters can be applied when AUC was close to 1 and model retraining or redesigning need to be considered when AUC was close to 0.5.

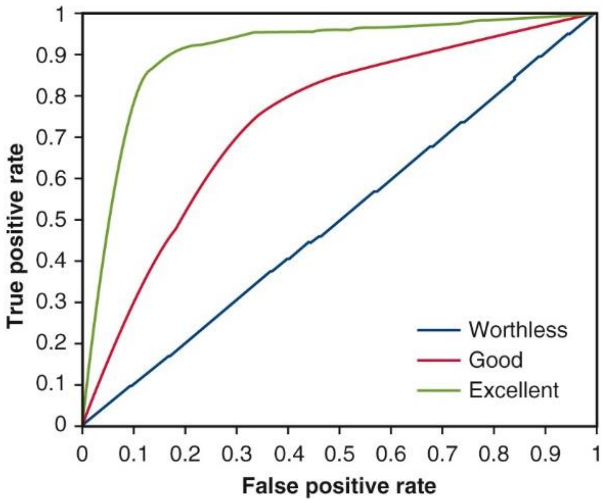


Figure 6.1 Comparing models with different ROC curve

6.2. Continual Learning Schemes

In corresponding to the dynamic changes in the clinical environment, the CDSS needs to adjust its decision-making strategy accordingly to ensure the robustness of diagnostic accuracy. However, the rapid modification of the prediction model can be extremely costive under many circumstances, which shows an essential need for specially designed decision algorithms that can cope with the dynamic environment cost-effectively. Researchers have already noticed the importance of such a requirement. Various kinds of adaptation have been proposed by modifying the conventional classification algorithms in the past. Some of them are reviewed in the following subsections.

6.2.1. The PEBLS Algorithm

The PEBLS (Parallel Exemplar-Based Learning System) is an adaptive classification scheme based on a conventional kNN classifier (Steinbach & Tan, 2009), which assigns an additional weight to each training sample. These weights $\{W_1, W_2, \dots, W_n\}$ regarding *n* training samples can tune themselves dynamically according to the result in each validation round and therefore accommodate to the complex online environment, which each of the weight W_i in corresponding to the i^{th} training sample is defined as

$$W_i = \frac{T_i}{\tau_i} \tag{6.3}$$

where τ_i is the total number of the correct classification when using the i^{th} training sample as a reference and T_i is the total number of times that the i^{th} training sample is referenced as a nearest neighbour. Notice that both of the τ and T should be initialized to 1 to avoid division by zero error, and the result is always a positive real number.

Following this, the weighted distance d' between two feature point $\vec{x}_m$ and $\vec{x}_n$ can then be defined as

$$d'(\vec{x}_m, \vec{x}_n) = W_m W_n d(\vec{x}_m, \vec{x}_n) \tag{6.4}$$

which cause the training samples that makes correct predictions being more favoured in the future testing rounds and the training samples that often produce errors being punished. Such idea can be extended after the model has been deployed and the weights associated with each training sample can be continuously adjusted.

6.2.2. The ID4 Algorithm

The ID4 algorithm is a decision tree induction algorithm that shares the same concept as the basic ID3 tree-construction algorithm, in which the tree is grown by maximising the information gain at each node induced. The deficiency of the conventional ID3 algorithm is that the whole structure of the tree has to be reconstructed from the start as a result of any modification made to the training set, due to the fact that the information gains overall attributes have to be recalculated. As a solution, ID4 algorithm was proposed, which allows the induced decision tree to partially replace its branch based on new instances observed (Schlimmer & Fisher, 1986).

The updating cost of the conventional ID3 tree is equivalent to reconstructing a new tree, which the cost regarding *n* training samples(instances) can be defined as

$$\sum_{a=|A|}^{|A|-d} a \times n = O(|A|^2 \times n) \tag{6.5}$$

where $|A|$ is the number of attributes and *d* is the depth of the tree (which cannot exceed $|A|$). In addition, in the worst scenario, a tree may need to be rebuilt after every new observation encountered, then the above cost has to repeat itself for n many times, which finally accumulates to a cost as

$$\sum_{i=1}^{n} i \times |A|^2 = O(|A|^2 \times n^2) \tag{6.6}$$

On the contrary, the ID4 tree only updates the necessary parts of the tree, which the cost is only proportional to the square of the number of attributes, which is calculated as

$$\sum_{i=1}^{n} |A|^2 = O(|A|^2 \times n) \tag{6.7}$$

Therefore, the ID4 algorithm is more cost-effective than the conventional ID3 algorithm due to its adaptiveness to small changes.

6.3. Potential Methods for Decision Score Refinement

6.3.1. One-pass Mean/Variance

As described in Section 3.3.2, the posteriors in the proposed MGM model are determined by a set of mean vectors $\vec{\mu}_{\omega_i}$ and variance vector ${\vec{\sigma}_{\omega_i}}^2$. Retraining on these parameters can be very costly since the computation normally requires iterating through the entire training set repeatedly. Alternatively, these parameters can be updated based on the previous result efficiently on the fly. The new parameters $\vec{\mu}_{\omega_i}{}'$ and $\vec{\sigma}_{\omega_i}{}^{2'}$ can be calculated based on the previous parameters $\vec{\mu}_{\omega_i}$ and $\vec{\sigma}_{\omega_i}{}^{2}$ at a given new reading $\vec{x}_{\omega_i}$ as

$$\begin{cases} \vec{\mu}_{\omega_i}{}' = \vec{\mu}_{\omega_i} + \dfrac{(\vec{x}_{\omega_i} - \vec{\mu}_{\omega_i})}{n_{total}} \\ \vec{\sigma}_{\omega_i}{}^{2'} = \dfrac{\Sigma(n_{total}, \omega_i)'}{n_{total} - 1} \\ \Sigma(n, \omega_i)' = \Sigma(n, \omega_i) + (\vec{x}_{\omega_i} - \vec{\mu}_{\omega_i})(\vec{x}_{\omega_i} - \vec{\mu}_{\omega_i}{}') \\ \Sigma(n, \omega_i) = \sum_{k=1}^{n} (\vec{x}_{k\omega_i} - \vec{\mu}_{\omega_i})^2 \end{cases} \tag{6.8}$$

where the algorithm additionally requires to buffer the number of total elements n_{total} and the sum of the squared deviation $\Sigma(n, \omega_i)$. One of the disadvantages of these solutions is that the calculation results are not numeric reliable. The influence of $\frac{(\vec{x}_{\omega_i} - \vec{\mu}_{\omega_i})}{n_{total}}$ can decrease along with the increase of n_{total}, where the additional change in each update eventually becomes very small as it can be ignored, therefore causing the decision model to be unchanged. This "vanishing gradient" phenomenon also often occurs with neural network training, which bears some analogy here. In addition to this, the accumulating sum of the squared deviation can be a potential cause of the computation overflow. Therefore, this solution in fact does not fundamentally solve the refinement issue. All parameters are still required to be retrained after a period of refinement.

6.3.2. Adjustment on Class Priors

In contrast to the tuning of posterior probabilities, the priors used in the relevant class model can also be modified in adapting environmental changes after deployment. Compared to tuning posteriors in compromising new observations, changing priors mainly alter the bias between different classes in general but not regarding specific classifications. The update of priors can be simply presented in a linear form as

$$\mathrm{P}(\omega_i)' = \begin{cases} m_{>} \cdot \mathrm{P}(\omega_i) + c_{+} \text{ if } \omega_i \rightarrow z \\ m_{<} \cdot \mathrm{P}(\omega_i) + c_{-} \text{ else} \end{cases} \tag{6.9}$$

where m is a constant of reward to the original decision score computed that affects the decision score based on its magnitude, where decision scores with higher confidence are influenced more heavily compared to the lower ones. Symbol m$_>$ denotes a constant that is greater than 1, which amplifies the prior of the targeted class when the classifier made a correct prediction. On the contrary, $m_<$ denotes a constant that is lower than 1, which attenuated the prior of the targeted class when the classifier made an incorrect prediction. c is a constant of bias defining the environmental preference of difference. c_+ denotes a positive constant that awards the prior of the targeted class when it made a correct prediction and c_- denotes a negative constant that penalises the prior of the targeted class when it made an incorrect prediction.

As defined in Formula 6.9, *m* and *c* have a dominating effect on the functional output. The system can easily be over tuned if the awarding/penalising factors were set too high. Meanwhile, the tuning can also be too slow in adapting to the new changes if the awarding/penalising factors were set too low. These two parameters must be set up carefully in achieving equilibrium of the tuning, which can be a very challenging task.

6.3.3. Transformation Function on Decision Score

In real-life practice, the relation between the decision score and the final decision strength can be more varied due to the external bias involved during the decision making. As an example that has been raised many times in this thesis, a doctor may be in favour to diagnose a case as malignant more than benign when the confusion occurred in breast lesion identification, since such bias provides a safer option to the patient and minimised the potential risks of misdiagnosis. Therefore, it is desirable to introduce an intermediate transformation function T between the decision score S_D and the final decision strength S'_D as a hidden factor, which maps $S_D : [-1,1]$ into $S'_D : [-1,1]$ as

$$S_D{}'(\omega_z|\vec{x}) = \mathcal{T}[S_D(\omega_z|\vec{x})] : [-1,1] \mapsto [-1,1] \tag{6.10}$$

The function $\mathcal{T}$ can be estimated by applying regressions of any kind once we have collected enough data regarding the real facts after the model deployment. Taking linear regression model as an example, the transformation function $\mathcal{T}$ can be presented as:

$$\mathcal{T}[S_D(\omega_z|\vec{x})] = m \cdot S_D(\omega_z|\vec{x}) + c \tag{6.11}$$

where *m* and c, in this case, are the two constants that represent the transformation rate and the external bias respectively. Therefore, given a validation set $\{\vec{v}\}$ with their expected decision score measurement $\{\widehat{\mathcal{V}}\}$, we can model our unknown parameters in the transformation function $\mathcal{T}$ by minimising their loss as:

$$\left\{m, c \middle| \{\vec{v}\} \longmapsto \{\widehat{\mathcal{V}}\}\right\} = \text{argmin}\left(\sum_{i=1}^{|\{\vec{v}\}|} |\widehat{\mathcal{V}}_i - \mathcal{T} \circ S_D(\vec{v}_i)| \,\middle|\, \{\vec{v}\} \longmapsto \{\widehat{\mathcal{V}}\}\right) \tag{6.12}$$

6.4. Experimental Analysis

6.4.1. Creating Testing Set Simulating Observations after Deployment

In this experiment, we have again used the CBIS-DDSM breast tumour data set. Unlike the cross-validation based approach that we implemented before, this time we have randomly selected 50% of the calcium data as the training set. The remaining 50% of the calcium data have been further divided into 15 patches to emulate observations after deployment in a time series for testing purposes. In addition, we have purposely chosen another group of mass data as unseen abnormalities, as mass tumours have a very different appearance compared to calcium tumours. These abnormal data have been mixed with the test patches created in simulating unforeseen changes after deployment. The initial proportion of abnormalities was set to 2% and then increased accumulatively by 5% in each test patch. In other words, the initial test patch contains 2% of unseen examples and increase gradually until the last testing patch contains 72%.

6.4.2. Experiment Result and Analysis

In the experiment, we have again used the GLCM feature as it has shown promising results in the previous experiments. However, we have decided to use a version of GLCM with lower dimensionality in making sure that the model can be well trained with the downscaled dataset. More specifically, we have derived the 13 Haralick's features from the GLCM that only in 1-pixel distance with 45° angle, which results in a relatively low dimensional feature of 13 dimensions.

Before testing the model with heavily polluted data as designed in Section 6.4.1, we have first tested our linear refining method introduced in Section 6.3.3 with noises that are similar to the training sample. In this initial test, both training samples and noise samples are calcium-type tumours but only with different subtypes (the training examples are round calcium and the noise examples are line calcium). Unlike what we did in other experiments where cross-validation methods were used, it can be slightly difficult to measure the expected performance value in this experiment as the testing follows a train/test split protocol. Therefore, we have repeated the testing 10 times with different seeds of randomization in making sure that the obtained observations were enough in conducting a thorough test. The averages of the testing accuracies were recorded and plotted in Figure 6.2.

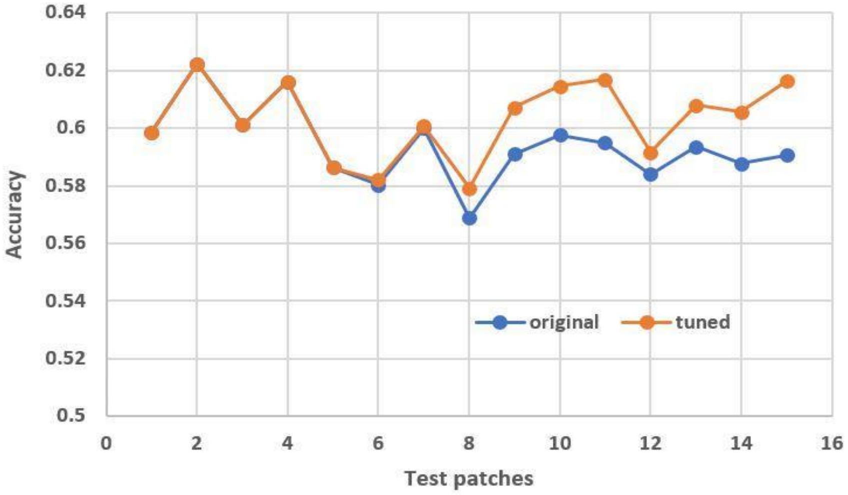


Figure 6.2 Classification accuracy on similar testing samples

According to Figure 6.2, it appears that the performance of the two trained models start to show distinctive differences after the eighth round. As designed, the eighth patch contain 37% of noise, which provided us with an initial clue about the proportional threshold that the noise may influence the testing results. Overall, the test results showed that models with refinement had a slightly better performance compared to the unrefined model on average. However, the difference was not significant enough to conclude that the method proposed was effective in improving the online classification result, but at least showed that the linear fine-tuning method did not degrade but maintained the system performance. On the contrary, the performance of the unrefined model showed a clear decaying pattern along with the pass in time.

In the second test, we have maintained the setting on the feature and testing protocol used, but changed the testing dataset to a heavily polluted one as described in Section 6.4.1. Again, the averages of the testing accuracies were recorded and plotted in Figure 6.3.

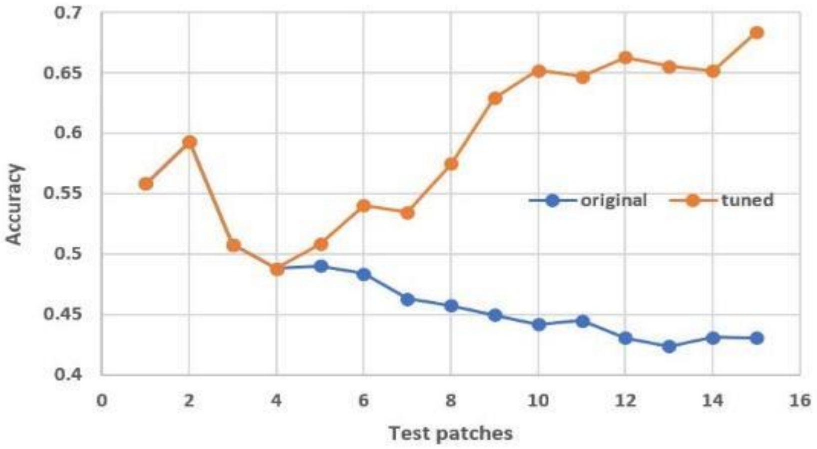


Figure 6.3 Classification accuracy on large variated testing samples

As shown in Figure 6.3, the two trained models showed much quicker and significant differences in their performance compared to the first experiment. The difference appeared after the fifth round, which was designed to contain 22% of noise. This is understandable since the noise in this experiment was much distinctive and abnormal, which accelerated the degradation of the system performance. The refined model overperformed the unmodified model clearly in long run. The performance of the unmodified model decreases continuously along with the increase in noises, while the refined model remains robust with increasing performance.

In the final experiment, we have upgraded the GLCM features used into three different distances (1, 2 and 3) with four different angles (0°, 45°, 90°, and 180°), which results in a high dimensional feature vector of 3 × 4 ×13 = 156 dimensions, which is the same as what we used in the experiments in the previous chapters. This experiment aims to test the performance of the linear refinement method in high dimensional spaces. We have again tested it with the heavily polluted dataset under the same testing protocol. The averages of the testing accuracies were plotted in Figure 6.4.

This time, similar to the first experiment, the experiment result shows that the refined performance was not significantly different from the unrefined one. A possible explanation lay at the nature regression requirement on high dimensional data. As we know, high dimensional data essentially requires more data points in fitting the regression line accurately. Therefore, it inevitably requires more patches and iterations to refine the model before superior performance may appear. It is very likely that Figure 6.4 was only showing the very initial stages of the refinement. A piece of good evidence in supporting such an argument is that the performance difference appears from the ninth round of the testing, which was far later than what the previous experiments found.

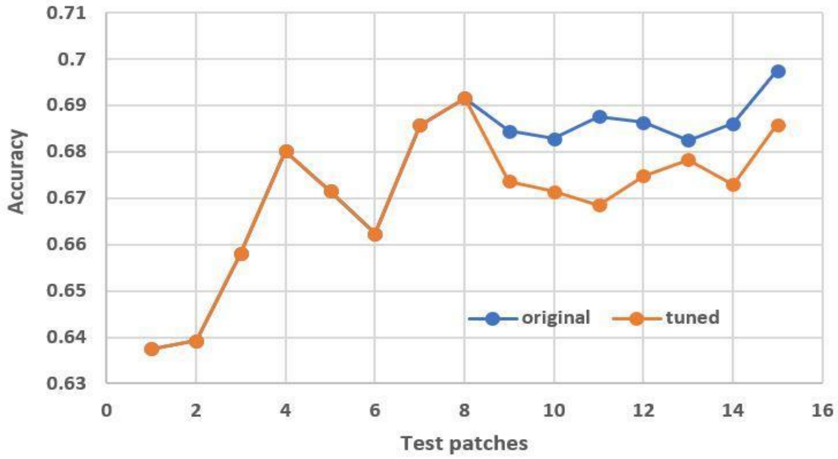


Figure 6.4 Classification accuracy on high dimensional data

As another interesting observation, the experiment with high dimensional features has shown a higher accuracy compared to the experiments built on low dimensional features. This again showed that the high dimensionality has provided better discrimination power to the classifier compared to low dimensionality, which has been observed repeatedly in many experiments of this thesis.

6.5. Discussions

6.5.1. Using Decision Sensitivity as a Monitoring Metric

In Section 6.1, we have introduced several metrics for monitoring the performance of CDSS after deployment. However, we have not mentioned the measure of decision sensitivity as introduced in Section 4.4.2. Similar to classification accuracy, decision sensitivity is also a measure of modelling performance derived from a batch of data, where it is a real value ranging between 0 and 1 and being most desirable when a mid-range value is maintained. In principle, it can monitor how the decision score model reacts to new observations after deployment by measuring whether it is still being reasonably sensitive to unseen data. However, the proposed sensitivity measure may still be immature yet to be put for performance monitoring, and hence why we have not conducted any experimental analysis.

In Section 4.4.2, we provided a clear definition for the decision sensitivity measure, where a measurement of 1 denotes extremely sensitive, indicating potential model overfits whereas a measurement of 0 denotes extremely insensitive, indicating potential model underfits. Following the definition, observing a sensitivity that was close to 0 or 1 can be a strong indication that the model is getting obsolete. However, it is difficult to draw a clear conclusion when the measured decision sensitivity was in the middle of the range. Unlike the accuracy measure where a higher value is always more desirable than the lower ones, it is hard to argue whether a decision sensitivity of 0.6 is more desirable than 0.4 or not. The vague and incomparable nature of the decision sensitivity measure caused some major difficulty when applying it as a criterion for performance monitoring and model refinement.

As a practical solution, a margin of tolerance with a lower and an upper bound threshold within the decision sensitivity measurement range can be considered. In other words, a model can be identified as being obsolete when the decision sensitivity measured was less than 0.05 or greater than 0.95. Any value measured beyond such margin can be considered as an alert for fault, and subsequently triggers further treatments to the model. However, such a solution can be quite risky in meeting the punctual and timely requirement for retraining a new classification model for CDSS, as the decision score model used can be already very unreliable by the time when the fault alert is triggered and can potentially cause huge damage (such as cost of money, dangerous to life, treatment time spent etc.) with the number of false decisions already made.

Besides these concerns, in Section 4.2, we have listed two main factors that influence the measurement of decision sensitivity, i.e., the dimensionality of the feature vector used and the eigenvalues of it. Unfortunately, both of these two factors are costly to refine and require the retraining of the built model. Although in Section 6.3, we introduced three possible methods for efficiently refining the decision score model, but the impact of these methods on the decision sensitivity remains an open question. Nevertheless, none of these proposed methods is expected to impact the decision sensitivity greatly as they do not adjust the feature used in the decision score model, which does not impact dimensionally and eigenvalues of the model trained and therefore not expect to affect the decision sensitivity very much. Nonetheless, it is still interesting to observe how the change in accuracies affects the measurement of decision sensitivity.

Overall, the decision sensitivity can be potentially useful for monitoring the performance of models, but only when a better understanding is reached. Based on that, future research may lead to proposing a refinement method that does not only consider accuracy but also the decision sensitivity during deployment.

6.5.2. Online Decision Score Refinement by Modifying Priors

In Section 6.4, we have tested the refinement method proposed in Section 6.3.3 under different settings, where preliminary experiments showed a promising result. But unfortunately, we were not able to present any result regarding the adjustment-based method proposed in Section 6.3.2, since experiments failed quickly without showing any readable performance for evaluation. More specifically, the adjustment-based method introduced in Section 6.3.2 requires two sets of predefined parameters $m_{>}/m_{<}$ and c_{+}/c_{-} to constantly refine the decision score measure according to the performance observed after deployment which expects to achieve an optimal equilibrium in long run. However, we have not been able to find such equilibrium in our preliminary test. For simplicity of variable control, we have initially tried to set c_{+}/c_{-} to 0 with $m_{>}/m_{<}$ valued between 1.1-1.7/0.9-0.3. Unfortunately, we found that the amount of penalizing/awarding was too large even with 1.1/0.9 and the decision score measure has been quickly over tuned to constantly bias one of the classes after the first couple of iterations, which makes the experiment result not meaningful. As we have found in the preliminary experiment, the selection of parameters can be very critical for adjustment-based methods.

Regarding the selection of variables, parameter optimization has always been a popular study especially when involving data in serials/patches. As a good reference, the training of Convolutional Neural Network (CNN) uses a huge amount of data arranged in a list of small batches in refining the parameters used in the model. The training uses the Stochastic Gradient Descent (SGD) method to update the weight at each node based on the validation outcomes of each batch, where the amount of change is based on a predefined hyper-parameter η, commonly known as the "learning rate". Through the past decades, many alternatives to the SGD method have been proposed but they all follow a similar concept, where the simple iterative solution can be expressed as

$$w_{i+1} = w_i - \eta \nabla Q(w_i) \tag{6.13}$$

where w_i denotes the weight at the ith round and $\nabla Q(w_i)$ denotes the amount of changes expected to make in meeting the optimal weight for correctly predicting the example at the ith round. As we can see, these concepts are very similar to what we have introduced in Section 6.3.2 and the selection of η is again critical for correct modelling. η can be determined naively through exhaustive trials, but such a solution is of course very costly and undesirable. As a common practice, people do sometimes increase the amount of learning rate from batches to baches in the training process and then creates a plot of learning rate versus loss. The learning rate with minimum loss is then considered as the most optimum value. However, this selected value is customised to the training set and may not be suitable after the system has been deployed. As the clinical environment changes constantly with a high cost to any potential risk (due to the life-critical nature of clinical decisions), most of the optimum learning rates found can be too small to adapt to environmental changes swiftly. The method has to be able to adjust the decision score measure timely with precision, where ideally the method should be able to adjust the amount of updates dynamically in meeting the requirement with the least amount of time used.

There have been many adaptive scheduling techniques being proposed for altering the learning rate on the fly, where most of them involve introducing a decaying factor in adjusting the learning rate based on time or number of observations (Ge, et al., 2019) (Cao, et al., 2019). Ideally, the decaying factor is normally a fraction so the models are gaining less and less to the most appeared examples as they should already be very well learned. For CDSS, we can alternatively reverse such logic by additionally introducing a decaying factor greater than one when the system makes the same misprediction repetitively, so that the model is gaining more and more in correcting such a misprediction. This additional solution should be able to correct any unforeseen misprediction timely while maintaining precisions on the decision score measure. However, our experiment data was not sufficient for expanding on such a study and the investigation on optimal parameter refinement methods can be potentially beyond the scope of this thesis.

6.6. Summary

In this Chapter, we were mainly interested in exploring potential methods for maintaining the level of robustness of the decision score measure after the CDSS has been deployed. We have first looked at several metrics that can be used for monitoring CDSS performance after deployment. Following that, we have gone through multiple methods that can efficiently refine themselves in corresponding to newly-arrived observations, which eventually leads to the proposal of 3 alternative methods for automatically refining the decision score models after the CDSS has been deployed. Unfortunately, some of them did not perform as we expected due to the limitation of testing data and research scope.

Nevertheless, we have still been able to gain some positive knowledge on one of the proposed methods, which was based on refining the decision model with linear transformation functions. We have tested this method with several experiments in mimicking different situations that CDSS may encounter after the deployment. The proposed method has shown robust performance in all the experiments conducted. More importantly, the proposed method tends not to be sensitive to similar features, which makes the model very unlikely to be over tuned in the long run. At the same time, it remains sensitive to data that vastly variated from the training examples so it can still correctly react to any faulty prediction in time. We have additionally found that the time (iterations) required for the refinement method making a positive change can be potentially proportional to the feature dimensionality, where decision score measured built on higher dimensional feature essentially requires more time (iterations) to refine the model.

Chapter 7. Conclusions and Future Work

7.1. Summary of the Undertaken Research Investigations

The research work as reported in this thesis was designated to investigate various aspects of integrating automated Clinical Decision Support Systems (CDSSs). The research aims are meant to exploit the tangible and growing benefits of using machine learning techniques in modern healthcare systems primarily in support of medical diagnosis. The objective is to determine and understand the constraints of modern CDSS models and propose potential solutions to these constraints. Having initially conducted a brief review of CDSS background and its essential functional elements (Chapter 1), it became clear that optimising the decision-making module of CDSS cannot rely entirely on prior information obtained during the training stage. Instead, there was a need to introduce a different scheme of confidence-centric decision making in CDSS. Such a scheme does not only provide an eventual decision outcome, but couples the outcome with a level of decision reliability. This way of decision making is sensible to doctors, consistent with customary practice in clinics, and hence easier to integrate the machine models into the final diagnostic decision-making process within CDSS. Confidence of decisions also contributes towards decision explainability, an essential requirement in clinical practice (Section 2.1), because the confidence level can be seen as a reflection of decision strength.

After conducting a broad review of existing various measures for decision strengths, the thesis proposed a unified confidence measure based on the Gaussian Bayes principle (Chapter 3). In addition, the proposed measure of confidence is combined into a single decision score measure, which does not only reflect the strength of decision making in percentage levels but also highlights the decision made with positive or negative signs measure (Section 3.3). Based on the decision measure, multiple variants have been tested and compared using a dataset of low dimensions and explainable features. The experimental results demonstrated that the proposed decision measure worked well in low dimensional features space for classifying miscarriage cases (Section 3.4). These results raised the question as to whether this success can be extended and generalised into more complex and diverse diagnostic scenarios.

In particular, our next objective was to investigate the validity of the proposed decision score measure in a more sophisticated setting (Chapter 4), where the features involve more complex relationships and are of high dimensionality. We have tested the proposed measure on a new dataset about breast lesions in mammography that uses complex high dimensional features for classification (Section 4.5), but encountered issues due to the well-known challenge of the “curse of dimensionality”. As a solution, we found it can be very useful to revisit the problem under orthogonal projections of the original dimension with different levels of dimension reduction. This was implemented using the well-studied Principal Component Analysis (PCA) method (Section 4.3). Within the projected PCA feature spaces, we have thoroughly studied the behaviour of the confidence measure and found that the confidence measure is sensitive to the variation of dimension reduction together with the eigenvalues in the projected dimension. Visualisation of the behaviour across a sequence of dimension reduction levels, has helped develop a new measure, referred to as the *decision sensitivity measure*, that can be used to quantify the behaviour of the confidence measure and can be used as an evaluation criterion (Section 4.4).

After the investigation about measuring the strength of decisions over a single set of features, this thesis acknowledged the limitations of a single classifier. Hence, we investigated how to measure the combined strength of decision making by multiple classifiers in a joint decision-making framework (Chapter 5). We reviewed the principles and rationales behind various well-established information fusion schemes, tested their strengths in adapting the confidence measure (Section 5.2) within the schemes, and further proposed a correlation-based scheme for fusing the proposed decision scores (Section 5.3). Performances of these fusion schemes were evaluated and compared on the two datasets introduced in the previous chapters (Section 5.4).

The last piece of the research work reported in this thesis aims at optimizing the CDSS decision making module in a cost-effective manner post deployment (Chapter 6). In particular, it is necessary to understand the effects on decision confidences under unpredictable clinical environments, where CDSS face unforeseen changes constantly. As a result, the CDSS is required to modify its decision-making module spontaneously, continually and regularly in adapting to such an unpredictable environment. Therefore, the thesis further investigated possible ways to spontaneously refine the trained models cost-effectively to achieve more robust performance. More specifically, the thesis first broadly reviewed several commonly used metrics and methods for monitoring and refining models (Section 6.2). Sensible adaptations of these methods to adjust our proposed confidence measure are then investigated (Section 6.3). Experiments were conducted on a specifically created dataset that emulates time serial data using the breast tumour data set used previously in chapters 4 and 5 (Section 6.4).

7.2. Major Contributions and Findings

In this thesis, we have proposed a special CDDS framework that uses decision strength as a core element throughout the entire design of the CDSS (see Section 2.1). In more specific, the proposed framework consists of three major components for measuring the strength of decision making, fusing multiple decisions made and updating trained decision-making models dynamically, respectively. Such design provides a mutual and robust solution in bridging the understanding gap between experimental study and practical application, which makes the proposed CDSS fulfilling the obligation to explain.

In this thesis, we have proposed 4 variant types of models for measuring the strength of the decision making (Section 3.3). Overall, these models have shown good performance for classification. Besides, all the variants are found to well represent the known clinical facts (Section 3.4.2), demonstrating the soundness of adopting the Gaussian Bayes principle for modelling classification decision strength. In particular, we found that multivariate Gaussian models have more distinguishable powers when performing classifications than the univariate Gaussian models. In addition, mixture models also have had a better fit to the expected confidence measurement compared to non-mixture models. Both elements combined into a strong final belief that the Multivariate Gaussian Mixture Model (MGMM) is the most ideal model to be used when building our proposed confidence measure in general.

We have also found that measuring the strength of decision making in high dimensional spaces requires more sophisticated techniques compares to their application in a low dimensional space (Section 4.1). We found that the major obstructions of deploying the proposed confidence measure under complex scenarios are from two main factors: (a) the dimensionality of the feature used, and (b) the standard deviation of the feature used in the orthogonal space. Our contribution in dealing with these challenges is the proposed threshold-based PCA method that allows the confidence measure to alter itself towards more reasonable and robust predictions according to the training environment (Section 4.2). More significantly, we found that pruning the feature dimensions with very large or small eigenvalues can both benefit modelling the confidence measure (Section 4.4). More specifically, pruning the feature dimensions with very large eigenvalues helps to reduce the ambiguity of the confidence measured and pruning the feature dimensions with very small eigenvalues makes the confidence measure more robust. Nevertheless, such a finding has to be constrained within a limit to prevent potential damage to the information gained in the confidence measure due to over-pruning. In general, we have found that the decision sensitivity follows a down-up-down trend when pruning the feature dimensions from the smallest eigenvalue one by one. Following such a finding, the ideal threshold of minimum eigenvalues is considered to be positioned at the beginning of the second peak (i.e., after the first up and before the second down). At such a threshold, the confidence measured maintains the best trade-off between robustness and discriminant power. The significance of this multi-faceted contribution can be appreciated, as most of the PCA implementations only consider the efficiency and decision accuracy when applying it but rarely consider the implications on the decision confidence.

The third main contribution of this thesis, is related to fusing confidence measures from classifiers built on multiple features, which aims at improving the robustness of the prediction. More specifically, we developed a categorisation of 10 different fusion methods in terms of three different schools of thought (Section 5.2 – 5.3), covering the traditional rule-based and weight-based fusion, plus our newly developed correlation-based fusion. Most of them have shown comparable classification accuracy to the best of individual classifiers with superior robustness (Section 5.4). In particular, we found that both the median rule scheme and our proposed correlation-based scheme had very good performance in all testing scenarios.

The last but not least of this thesis main contribution is concerned with post-deployment diagnostic schemes. In particular, we have proposed three methods that efficiently adapt classification confidence in response to newly-arrived observations (Section 6.2). The corresponding investigations revealed that the weight-based method essentially requires a careful setup when tuning the prior of the Gaussian Bayes model. Setting weight as a constant is very likely to over exaggerate the award/penalty and therefore cause the model to constantly bias towards one of the classes. Compared to the weight-based method, refining the decision model with linear transformation functions have shown more robust performance in our experiments under different conditions. Refinement from the transformation-based method was not being sensitive to familiar examples but at the same time remaining sensitive to data that vastly variated from the training examples, which makes it correct unseen faulty prediction on time while also remaining robust to seen examples. We have also found that the time (iterations) required for the refinement method making a positive change can be potentially proportional to the feature dimensionality, where decision score measured built on higher dimensional features essentially requires more time (iterations) to refine the model.

7.3. Future Works

Our proposed decision score measure is based on Bayesian principles with Gaussian models. The Gaussian model is considered to be one of the most universally applicable models as it assumes a normal distribution to the observations, which is mostly expected in natural events. However, the modelling is not limited to Gaussian form alone, and there are many other probability distribution models such as binomial or Poisson distributions, etc. that are also applicable (Matthews & Vernon, 2015). But each of these models has its unique characteristics, which makes them more suitable under different presumptions. It would be very interesting to explore the behaviours of these models in future research in order to make our proposed decision measure system more universally adaptable.

For our proposed Gaussian Bayes decision score measure, we have specifically studied its behaviour in high dimensional space and proposed a threshold-based PCA method for optimising the model trained. However, tuning the eigenvalue thresholds on both the maximum side and minimum sides in a consistent manner can be challenging due to the different magnitudes of information contained on each side. A practical solution in determining the appropriate thresholds could rely on the use of Confidence Interval (CI), which determines the maximum and minimum eigenvalue thresholds as the upper and lower limits of the CI at any confidence level specified. In this form, the eigenvalue thresholds can be determined and tuned in the context of confidence levels as an empirical method depending on the environmental requirements. However, the unsymmetrical property of the eigenvalue distribution causes the computation of the CI to be very difficult. If possible, approaches based on CI can be further tested and validated in the future along with the increased understanding of eigenvalue distributions.

Apart from the decision score measure, we have also proposed a decision sensitivity measure that helped in evaluating the fitness of the decision score model derived. Following the definition, observing a sensitivity that is close to 0 or 1 can be a strong indication that the decision score model is less appreciable to the seeing examples. However, it is difficult to draw a clear conclusion when the measured decision sensitivity was in the middle of the range. The vague and incomparable nature of the decision sensitivity measure made it very difficult for using it as a fine metric for performance evaluation. Therefore, the characteristic of our proposed decision sensitivity measure would definitely need further studies for contributing to better-defined evaluation metrics.

Additionally, we have introduced a fusion method based on general correlations between pairs of classifiers in coping with the decisions made from different features. Despite it has shown promising performance in our preliminary tests, we could still further refine it into more sophisticated scenarios, where different disciplines can be applied in achieving better performance. For example, the fusion outcomes of independent classifiers can always be fused with mean or product rules, which is the expectation under the independence assumption. On the other hand, fusions on correlated classifiers can depend on their relationships. For the classifiers that have different correlations on the correctly classified and misclassified, different penalty and reward schemes can be applied for tuning the final fusion decision score to an appropriate value. In this case, the fusion strategy should be made dependent on the purpose of applications, which may need more investigation in the future.

References

Al-karawi, D., 2019. *Texture Analysis based Machine Learning Algorithms for Ultrasound Ovarian Tumour Image Classification within Clinical Practices.* Buckingham: The Univrsity of Buckingham.

Andrecut, M., 2008. Parallel GPU Implementation of Iterative PCA Algorithms. *Journal of computational biology: a journal of computational molecular cell biology,* 16(11), pp. 1593-1599.

Basavanhally, A. N. et al., 2010. Computerized Image-Based Detection and Grading of Lymphocytic Infiltration in HER2+ Breast Cancer Histopathology. *IEEE Transactions on Biomedical Engineering,* 57(3), pp. 642-653.

Belsley, D. A., Kuh, E. & Welsch, R. E., 1980. The Condition Number. In: *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity.* New York: John Wiley & Sons, p. 100–104.

Berner, E. S., 2007. *Clinical Decision Support Systems Theory and Practice.* New York: Springer.

Bourne, T. & Bottomley, C., 2012. When is a pregnancy nonviable and what criteria should be used to define miscarriage?. *Fertility and sterility,* pp. 1091-1096 .

Boutell , M. & Luo , J., 2004. Bayesian Fusion of Camera Metadata Cues in Semantic Scene Classification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2(2).

Büchlmann , P. & Yu , B., 2002. Analyzing Bagging. *The Annals of Statistics,* 30(4), pp. 927-961.

Çamlica, Z., Tizhoosh, H. R. & Khalvati, F., 2015. Medical Image Classification via SVM Using LBP Features from Saliency-Based Folded Data. *International Conference on Machine Learning and Applications,* pp. 128-132.

Cao, G., Song, W. & Zhao, Z., 2019. Gastric Cancer Diagnosis with Mask R-CNN. *11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC),* Volume 1, pp. 60-63.

Cardoso, J. S. & Cruz-Almeida, Y., 2016. Moving Beyond the Eigenvalue Greater Than One Retention Criteria in Pain Phenotyping Research. *Pain,* 157(6), pp. 1363-1364.

Carlin, B. P. & Loui, T. A., 2000. *Bayes and Empirical Bayes Methods for Data Analysis.* 2nd ed. New York: CHAPMAN & HALL.

Dai, J. & Xu, Q., 2013. Attribute Selection Based on Information Gain Ratio in Fuzzy Rough Set Theory with Application to Tumor Classification. *Applied Soft Computing,* 13(1), p. 211–221.

Dalal, N. & Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* Volume I, p. 886–893.

Datta, B. N., 2010. Chapter 7. QR Factorization, SVD, and Projections. In: *Numerical Linear Algebra and Applications, Second Edition.* Philadelphia: Society for Industrial and Applied Mathematics, pp. 227-228.

Désir, C., Bernard, S., Petitjean, C. & Heutte, L., 2012. A Random Forest Based Approach for One Class Classification in Medical Imaging. *Machine Learning in Medical Imaging,* pp. 250-257.

Dinov, I. D., Christou, N. & Gould, R., 2009. Law of Large Numbers: the Theory, Applications and Technology-based Education. *Journal of Statistics Education,* 17(1), pp. 1-19.

Dirk, T. et al., 2010. Ovarian Cancer Prediction in Adnexal Masses Using Ultrasound-based Logistic Regression Models: A Temporal and External Validation Study by The IOTA Group. *Ultrasound Obstet Gynecol,* 36(2), pp. 226-234.

Do, C. B., 2008. *The Multivariate Gaussian Distribution,* Stanford: Stanford University.

Domingos, P., 2000. A Unified Bias-Variance Decomposition and its Applications. *Proceedings of 17th International Conference on Machine Learning,* pp. 231-238.

Doorhof, D., 2018. *Thesis: Using Reinforcement Learning to Improve Clinical Decision Making in Neonatal Care.* Heidelberglaan: Faculty of Science, Information and Computing Sciences, Utrecht University.

Doyle, S., Feldman, M., Tomaszewski, J. & Madabhushi, A., 2012. A Boosted Bayesian Multiresolution Classifier for Prostate Cancer Detection from Digitized Needle Biopsies. *IEEE Transactions on Biomedical Engineering,* 59(5), pp. 1205-1218.

Elshinawy, M., Badawy, A. H., Abdelmageed, W. & Chouikha, M., 2011. Effect of breast density in selecting features for normal mammogram detection. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro,* pp. 141-147.

Filmus, Y., 2010. *Two Proofs of the Central Limit Theorem.* [Online]

Available at: www.cs.toronto.edu/~yuvalf/CLT.pdf

[Accessed 22 April 2017].

Folland, G. B., 1999. *Real Analysis: Modern Techniques and Their Applications.* 2nd ed. New York: Wiley.

Freund, Y. & Schapire , R. E., 1999. A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence,* 14(5), pp. 771-780.

Geman, S., Bienenstock, E. & Doursat, R., 1992. Neural Networks and the Bias/Variance Dilemma. *Neural Computation,* 4(1), pp. 1-58.

Ge, R., Kakade, S. M., Kidambi, R. & Netrapalli, P., 2019. The Step Decay Schedule: A Near Optimal, Geometrically Decaying Learning Rate Procedure For Least Squares. *Neural Information Processing Systems (NeurIPS),* p. 14977–14988.

Geurts, P., 2002. *Contributions to Decision Tree Induction: Bias/Variance Tradeoff and Time Series Classification (Doctoral dissertation).* s.l.:ULiège-University of Liège.

Goldstein, A., 2000. Errors in Ultrasound Digital Image Distance Measurements. *Ultrasound in Medicine & Biology,* 26(7), pp. 1125-1132.

Grandvalet, Y., 2004. Bagging Equalizes Influence. *Machine Learning,* 55(3), pp. 251-270.

Gray, R. M., 2010. In: *Probability, Random Processes, and Ergodic Properties.* s.l.:Springer, p. 163.

Guan, X. et al., 2017. An Object-Based Linear Weight Assignment Fusion Scheme to Improve Classification Accuracy Using Landsat and MODIS Data at the Decision Level. *IEEE Transactions on Geoscience and Remote Sensing,* 55(12), pp. 6989 - 7002.

Gulshan, V. et al., 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Journal of the American Medical Association,* 316(22), pp. 2402-2410.

Han, D., Du, H. & Jassim, S., 2016. *Towards a Confidence-Centric Classification Based on Gaussian Models and Bayesian Principles.* York.

Han, D., Du, H. & Jassim, S., 2018. Controlling Sensitivity of Gaussian Bayes Predictions based on Eigenvalue Thresholding. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems,* 5(16).

Haralick, R. M., 1979. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE,* 67(5), pp. 786-804.

Holsapple, C. W. & Whinston, A. B., 1996. *Decision Support Systems: A Knowledge-based Approach.* Mineapolis: West Publishing Co.

Hong, L., Wan, Y. & Jain, A., 1998. Fingerprint Image Enhancement: Algorithm and Performance Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 20(8), pp. 777 - 789.

Hongbo, D., 2010. *Data Mining Techniques and Applications, An Introduction.* :Cengage Learning.

Huang, Q. et al., 2019. On Combining Biclustering Mining and AdaBoost for Breast Tumor Classification. *IEEE Transactions on Knowledge and Data Engineering,* 32(4), pp. 728 - 738.

Hughes, G. F., 1968. On the Mean Accuracy of Statistical Pattern Recognizers. *IEEE Transactions on Information Theory,* 14(1), pp. 55 - 63.

Hu, J., Brown, M. K. & Turin, W., 1996. HMM based Online Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 18(10), pp. 1039 - 1045.

Ibrahim, D. A., Al-Assam, H. & Hongbo, D., 2016. Automatic Segmentation and Measurements of Gestational Sac Using Static B-mode Ultrasound Images. *Mobile Multimedia/Image Processing, Security, and Applications,* Volume 9869.

Kesavan, S., 2019. Signed measures. In: *Measure and Integration.* Singapore: Springer, pp. 178-195.

Khalili , M. et al., 2020. Epidemiological Characteristics of COVID-19: a Systematic Review and Meta-analysis. *Epidemiology & Infection,* 148(130).

Khazendar, S., Al-Assam, H. & Du, H., 2014. Automated Classification of Static Ultrasound Images of Ovarian Tumours Based on Decision Level Fusion. *6th Computer Science and Electronic Engineering Conference (CEEC),* pp. 148-153.

Kittler, J., Hatef, M., Duin, R. P. & Matas, J., 1998. On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 20(3), pp. 226-239.

Kothari, S., Phan, J. H., Stokes, T. H. & Wang, M. D., 2012. Pathology Imaging Informatics for Quantitative Analysis of Whole-slide Images. *Journal of the American Medical Informatics Association,* 20(6), pp. 1099-1108.

Kragten , J., 1994. Tutorial Review. Calculating Standard Deviations and Confidence Intervals with a Universally Applicable Spreadsheet Technique. *Analyst*, 119(10), pp. 2161-2165.

Kumar, P. N., Satoor, S. & Buck, I., 2009. Fast Parallel Expectation Maximization for Gaussian Mixture Models on GPUs Using CUDA. *11th IEEE International Conference on High Performance Computing and Communications,* pp. 103-109.

Lee, R. S., Gimenez, F., Hoogi , A. & Rubin, D., 2017. A Curated Mammography Data Set for Use in Computer-aided Detection and Diagnosis Research. *Scientific Data*, Volume 4, pp. 2161-2165.

Lee, S.-J., Xu, Z., Li, T. & Yang, Y., 2018. A Novel Bagging C4.5 Algorithm Based on Wrapper Feature Selection for Supporting Wise Clinical Decision Making. *Journal of Biomedical Informatics*, Volume 78, pp. 144-155.

Li, R., Ye , S. & Shi, Z., 2002. SVM-KNN Classifier-A New Method of Improving the Accuracy of SVM Classifier. *Acta Electronica Sinica,* 30(5), pp. 745-748.

Liu, Y., 2000. *Statistical Behavior of the Eigenvalues of Random Matrices,* Princeton: Princeton University.

Majeed, T. F., Jawad, N. A.-. & Sellahewa , H., 2013. Breast Border Extraction and Pectoral Muscle Removal in MLO Mammogram Images. *5th Computer Science and Electronic Engineering Conference (CEEC),* pp. 119-124.

Maringe, C. et al., 2020. The Impact of The COVID-19 Pandemic on Cancer Deaths Due to Delays in Diagnosis in England, UK: A National, Population-based, Modelling Study. *Lancet Oncol,* 21(8), p. 1023–1034.

MathWorks, 2013. *Eigenvalues and Singular Values.* [Online]

Available at: https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/moler/eigs.pdf

[Accessed 25 Jun 2018].

Matjaž, M. & Zoran, B., 2011. ROC Analysis of Classifiers in Machine Learning: A Survey. *Intelligent Data Analysis,* 17(3), pp. 531-558.

Matthews, D. E. & Vernon, F. T., 2015. *Using and Understanding Medical Statistics.* 5th ed. :Karger Medical and Scientific Publishers.

McCracken, S. S. & Edwards, J. S., 2017. Implementing A Knowledge Management System within An NHS Hospital: A Case Study Exploring the Roll-out of An Electronic Patient Record (EPR). *Knowledge Management Research & Practice,* Volume 15, p. 1–11.

Mobasseri, B. G. & Lulu, A., 2021. Radiometric Identification of Signals by Matched Whitening Transform. *Sensors,* 21(24).

Moorman, P. G. et al., 2008. Hormonal Risk Factors for Ovarian Cancer in Premenopausal and Postmenopausal Women. *American Journal of Epidemiology,* 167(9), pp. 1059-1069.

Musen, M. A., Shahar, Y. & Shortliffe, E. H., 2013. Clinical Decision-Support Systems. In: *Biomedical Informatics.* New York: Springer, pp. 643-674.

Nardin, S. et al., 2020. Breast Cancer Survivorship, Quality of Life, and Late Toxicities. *Frontiers in Oncology,* Volume 10, p. 864.

Nasution, M. Z. F., Sitompul, O. S. & Ramli, M., 2018. PCA Based Feature Reduction to Improve The Accuracy of Decision Tree C4.5 Classification. *Journal of Physics: Conference Series,* Volume 978.

Nguyen, T.-T., Nguyen, T.-H. & Ngo, B.-V., 2021. *A GLCM Algorithm for Optimal Features of Mammographic Images for Detection of Breast Cancer.* Tokyo, s.n., pp. 295-299.

NHS Improvement, 2016. *Evidence from NHS Improvement on Clinical Staff Shortages A Workforce Analysis,* London: NHS Improvement.

NHS Trust, 2018. *Menopause.* [Online]

Available at: https://www.nhs.uk/conditions/menopause/

[Accessed 20 June 2020].

Ogunleye, A. & Wang, Q.-G., 2020. XGBoost Model for Chronic Kidney Disease Diagnosis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(6), pp. 2131-2140.

Owen-Williams, R. & Cornish, D., 2020. *Deaths registered in England and Wales: 2019,* Newport: Office of National Statistics.

Papadopoulos, H., Vovk, V. & Gammermam , A., 2007. Conformal Prediction with Neural Networks. *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI),* Volume 2, pp. 388-395.

Park, S. et al., 2012. Evolutionary History of Human Disease Genes Reveals Phenotypic Connections and Comorbidity Among Genetic Diseases. *Scientific reports,* 2(1), pp. 1-7.

Pastur, L. & Shcherbina, M., 2011. *Eigenvalue Distribution of Large Random Matrices.* Providence, Rhode Island: American Mathematical Society.

Patel, V., 2019. *Deaths registered in England and Wales: 2018,* Newport : Office of National Statistics.

Patil, V. V. & Kulkarni , H. V., 2012. Comparison of Confidence Intervals for the Poisson Mean: Some New Aspects. *REVSTAT – Statistical Journal,* 10(2), pp. 212 - 227.

Petrovai, D. M., 2019. Some Examples of Non-Measurable Lebesgue Functions. *Procedia Manufacturing,* Volume 32, pp. 640-642.

Pochampally, R. et al., 2014. Fusing Data with Correlations. *ACM SIGMOD international conference on Management of data,* pp. 433-444.

Prasad, S. & Bruce, L. M., 2008. Decision Fusion With Confidence-Based Weight Assignment for Hyperspectral Target Recognition. *IEEE Transactions on Geoscience and Remote Sensing,* 46(5), pp. 1448 - 1456.

Ren, J., 2012. ANN vs. SVM: Which One Performs Better in Classification of MCCs in Mammogram Imaging. *Knowledge-Based Systems,* Volume 26, pp. 144-153.

Ren, Y., Zhang, L. & Suganthan, P. N., 2016. Ensemble Classification and Regression-Recent Developments, Applications and Future Directions. *IEEE Computational Intelligence Magazine,* 11(1), pp. 41 - 53.

Risvik, H., 2007. *Principal Component Analysis (PCA) & NIPALS algorithm,* Oslo: University of Oslo.

Rubinstein, R. Y. & Kroese, D. P., 2016. *Simulation and the Monte Carlo Method.* :John Wiley & Sons.

Satpathy, A., Jiang, X. & Eng, H.-L., 2014. LBP-Based Edge-Texture Features for Object Recognition. *IEEE Transactions on Image Processing,* 23(5), pp. 1953 - 1964.

Sauter, V. L., 1997. *Decision Support Systems for Business Intelligence.* New Jersey: John Wiley & Sons Inc..

Schlimmer, J. C. & Fisher, D., 1986. A Case Study of Incremental Concept Induction. *Association for the Advancement of Artificial Intelligence (AAAI),* Volume 86, pp. 496-501.

Shafer, G. & Vovk, V., 2008. A Tutorial on Conformal Prediction. *Journal of Machine Learning Research* , pp. 371-421.

Shannon, C. E., 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal,* 27(3), pp. 379 - 423.

Shi, X., Wong, Y. D., Chai, C. & Li, M. Z.-F., 2021. An Automated Machine Learning (AutoML) Method of Risk Prediction for Decision-Making of Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems,* 22(11), pp. 7145-7154.

Shobowale, K. O., 2020. Ontology in Medicine as a Database Management System. In: V. Jain, R. Wason, J. M. Chatterjee & D. Le, eds. *Ontology-Based Information Retrieval for Healthcare Systems.* Beverly: Scrivener Publishing LLC, pp. 69-90.

Sivasubramanian, S., 2012. Amazon dynamoDB: A Seamlessly Scalable Non-relational Database Service. *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data,* pp. 729-730.

Sohl-Dickstein, J., Novak, R., Schoenholz, S. S. & Lee, J., 2020. On the Infinite Width Limit of Neural Networks with a Standard Parameterization. *International Conference on Learning Representations (ICLR).*

Song, L. et al., 2007. Supervised Feature Selection via Dependence Estimation. *24th international conference on Machine learning,* pp. 823-830.

Srinivas, N., Veeramachaneni, K. & Osadciw , L. A., 2009. Fusing Correlated Data from Multiple Classifiers for Improved Biometric Verification. *12th International Conference on Information Fusion,* pp. 1504-1511.

Srivastava, S. et al., 2008. Computer-aided Identification of Ovarian Cancer in Confocal Microendoscope Images. *Journal of Biomedical Optics,* 13(2).

Steinbach, M. & Tan, P.-N., 2009. kNN: k-Nearest Neighbors. In: V. Kumar, ed. *The Top Ten Algorithms in Data Mining.* New York: Chapman & Hall/CRC, pp. 151-161.

Sung, H. et al., 2021. Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians,* 71(3), pp. 191-280.

Tabesh, A. et al., 2007. Multifeature Prostate Cancer Diagnosis and Gleason Grading of Histological Images. *IEEE Transactions on Medical Imaging,* 26(10), pp. 1366-1378.

Tan, P.-N., Steinbach, M. & Kumar, V., 2019. *Introduction to Data Mining.* 2nd ed. Essex: Pearson Higher Education.

Tessler, F. N. et al., 2017. ACR Thyroid Imaging, Reporting and Data System (TI-RADS): White Paper of the ACR TI-RADS Committee. *Journal of the American College of Radiology,* 14(5), pp. 587-595.

Timmerman, D. et al., 2016. Predicting the Risk of Malignancy in Adnexal Masses Based on the Simple Rules from the International Ovarian Tumor Analysis Group. *American Journal of Obstetrics and Gynecology,* 214(4), pp. 424-437.

Tong, T. et al., 2017. Multi-modal Classification of Alzheimer's Disease Using Nonlinear Graph Fusion. *Pattern Recognition,* Volume 63, pp. 171-181.

Tregoning, J. S. et al., 2021. Progress of the COVID-19 Vaccine Effort: Viruses, Vaccines and Variants Versus Efficacy, Effectiveness and Escape. *Nature Reviews Immunology,* Volume 21, p. 626–636.

Turban , E., Aronson , J. E. & Liang , T.-P., 2004. *Decision Support Systems and Intelligent Systems.* Upper Saddle River: Prentice-Hall, Inc. .

UK Visas and Immigration, 2021. *Skilled Worker Visa: Shortage Occupations for Healthcare and Education.* [Online]

Available at: https://www.gov.uk/government/publications/skilled-worker-visa-shortage-occupations-for-health-and-education/skilled-worker-visa-shortage-occupations-for-healthcare-and-education
[Accessed 20 November 2021].

Valdovinos, R. M., Sánchez, J. S. & Barandela, R., 2005. Dynamic and Static Weighting in Classifier Fusion. In: *Pattern Recognition and Image Analysis.* Berlin: Springer, pp. 59-66.

Verma, J., Nath, M., Tripathi, P. & Saini, K. K., 2017. Analysis and Identification of Kidney Stone Using kth Nearest Neighbour (KNN) and Support Vector Machine (SVM) Classification Techniques. *Pattern Recognition and Image Analysis,* Volume 27, p. 574–580.

Wagholikar, K. B. et al., 2013. Formative Evaluation of the Accuracy of A Clinical Decision Support System for Cervical Cancer Screening. *Journal of the American Medical Informatics Association* , pp. 749-757.

Wang, J., Neskovic, P. & L. N., 2006. *A Statistical Confidence-Based Adaptive Nearest Neighbor Algorithm for Pattern Classification.*, p. 548–557.

Wang, S. et al., 2020. An Improved Random Forest-based Rule Extraction Method for Breast Cancer Diagnosis. *Applied Soft Computing,* Volume 86.

Wold, S., 1987. Principle Component Analysis. *Chemometrics and Intelligent Laboratory Systems,* 2(1-3), pp. 37-52.

Wood, A., Shpilrain, V., Najarian, K. & Kahrobaei, D., 2019. Private Naive Bayes Classification of Personal Biomedical Data: Application in Cancer Data Analysis. *Computers in Biology and Medicine,* Volume 105, pp. 144-150.

Xue, L. et al., 2006. An Algorithm Based on KNN and Improved SVM for Licence Plate Recognition. *Journal of Sichuan University,* 43(5), pp. 1031-1036.

Yang , L. et al., 2009. Virtual Microscopy and Grid-Enabled Decision Support for Large-Scale Analysis of Imaged Pathology Specimens. *IEEE Transactions on Information Technology in Biomedicine,* 13(4), pp. 636 - 644.

Zhang, W., Arvanitis, A. & Al-Rasheed, A., 2012. *Singular Value Decomposition and its numerical computations,* Michigan : Michigan Technological University.

Appendix A

Detailed Confusion Matrices Regarding Experiments in Section 5.4.1

Table A1. Confusion Matrix of Concatenated Feature Performance on Miscarriage Dataset

F_All		Truth		Accuracy
		PUV	MISS	
Predict	PUV	158	3	98.14%
	MISS	1	28	96.55%

Table A2. Confusion Matrix of Gestational Major Feature Performance on Miscarriage Dataset

Gma		Truth		Accuracy
		PUV	MISS	
Predict	PUV	154	7	95.65%
	MISS	9	20	68.97%

Table A3. Confusion Matrix of Gestational Minor Feature Performance on Miscarriage Dataset

Gmi		Truth		Accuracy
		PUV	MISS	
Predict	PUV	157	4	97.52%
	MISS	8	21	72.41%

Table A4. Confusion Matrix of Transpose Major Feature Performance on Miscarriage Dataset

Tma		Truth		Accuracy
		PUV	MISS	
Predict	PUV	150	11	93.17%
	MISS	6	23	79.31%

Table A5. Confusion Matrix of Concatenated Feature Performance on Breast Cancer Dataset

F_All		Truth		Accuracy
		B	M	
Predict	B	864	335	72.06%
	M	230	441	65.72%

Table A6. Confusion Matrix of LBP Feature Performance on Breast Cancer Dataset

LBP		Truth		Accuracy
		B	M	
Predict	B	928	271	77.40%
	M	243	428	63.79%

Table A7. Confusion Matrix of HOG Feature Performance on Breast Cancer Dataset

HOG		Truth		Accuracy
		B	M	
Predict	B	830	369	69.22%
	M	244	427	63.64%

Table A8. Confusion Matrix of GLCM Feature Performance on Breast Cancer Dataset

GLCM		Truth		Accuracy
		B	M	
Predict	B	919	280	76.65%
	M	212	459	68.41%

Table A9. Confusion Matrix of Histogram Feature Performance on Breast Cancer Dataset

HIST		Truth		Accuracy
		B	M	
Predict	B	775	424	64.64%
	M	185	486	72.43%