Multivariate Statistical Methods for Fault Detection and Classification

William Richard Southern, BEng(Hons), MSc.,

A thesis submitted to the University of Dublin in partial fulfilment of the requirements for the degree of

Doctor in Philosophy

Department of Mechanical & Manufacturing Engineering,
Trinity College Dublin,
Ireland.

In memory of my maternal grandparents
and paternal grandfather.

Declaration

I declare that I am the sole author of this thesis and that all the work presented in it, unless otherwise referenced, is my own. I also declare that this work has not been submitted, in whole or in part, to any other university or college for any degree or other qualification.

I authorise the library of Trinity College Dublin to lend this thesis.

William Richard Southern

October, 2005

Acknowledgements

I wish to offer my thanks to Peter Twigg who helped me at the start of my work. After his move to Manchester, I am indebted to Craig Meskell for his help, advice and general banter throughout and help during the write up.

Throughout my period in Trinity College, I have met a tremendous amount of people. There was a strong communal postgraduate idealogy, and I am glad to have been part of it. Although I would like to mention all acquaintances space precludes this so I wish to extend a thanks to all for the times had in TCD.

The oldest crew of Adriele, Alex, John B, and the 'Murphs' are still around and kicking. Seosamh is flying planes and setting a trend of bankrupting small airlines!. Thanks to all in *my* old office, Mary, Kevin, John G, Laoise, Danny, John V, Paul and John T. Mustard gets a special mention here also. Thanks to Orla in Fluids for being in such good form throughout. George, Laura and Oran too. Thanks to Damien for trying (and miserably failing!) to keep in rounds with me, for his input into mp3 encoding and for his interest into bike related things. A kind thank you is also extended to all staff members of the department during my time there.

The work would have been near impossible without the support and help of my family. My Father Peter, Mother Lorna, Sisters Sonya, Amanda, Kate and my Grandmother Nancy always offered encouragement and support when it was needed and I thank them all.

I reserve the most important thank you for the most important person in my life, Avril O'Connell. I have spent the majority of 4 years in her company and can't begin to imagine what I would be doing otherwise. Avril is the most selfless, supporting and kind person I know. Avril was and is always there for me and I wish to return the favour.

Presentations and publications resulting from this study

W.R. Southern, **'Preliminary Application of Principal Components Analysis to a Microchip Test Process for Mixed Signal/Radio Frequency Components'**, Advances in Process Analytics and Control Technology (APACT) 2003 Conference, York.

W.R. Southern, **'Multivariate Statistical Process Control for Fault Detection using Principal Component Analysis'**, Advances in Process Analytics and Control Technology (APACT) 2004 Conference, Bath.

W.R. Southern and C. Meskell, **'Multivariate Statistical Process Control for Fault Detection and Classification'**, to be published.

Contents

List of Figures

List of Tables

Acronyms

1R	**O**ne **R**ule
DIB	**D**evice **I**nterface **B**oard
DUT	**D**evice **U**nder **T**est
EDA	**E**xploratory **D**ata **A**nalysis
EWMA	**E**xponentially **W**eighted **M**oving **A**verage
FDI	**F**ault **D**etection and **I**solation
FPY	**F**irst **P**ass **Y**ield
JIT	**J**ust **I**n **T**ime
KDD	**K**nowledge **D**iscovery in **D**atabases
MSPC	**M**ultivariate **S**tatistical **P**rocess **C**ontrol
MSQC	**M**ultivariate **S**tatistical **Q**uality **C**ontrol
MSRF	**M**ixed **S**ignal and **R**adio **F**requency
NOC	**N**ormal **O**perating **C**ondition
PC	**P**rincipal **C**omponent
PCA	**P**rincipal **C**omponent **A**nalysis
PDF	**P**robability **D**ensity **F**unction
PQKB	**P**rocess **Q**uality **K**nowledge **B**ase
QFD	**Q**uality **F**unction **D**eployment
QI	**Q**uality **I**mprovement
SPC	**S**tatistical **P**rocess **C**ontrol
STDF	**S**tandard **T**est **D**ata **F**ormat
TQM	**T**otal **Q**uality **M**anagement
WCM	**W**orld **C**lass **M**anufacturing

Nomenclature

x	Scalar
$\mathbf{x}$	Vector
$\mathbf{X}$	Matrix
$\mathbf{x^T}$	Transpose
$\mathbf{X^{-1}}$	Inverse
α	Probability of type I error
β	Probability of type II error
Z	Standard normal variate
$\overline{x}$	Mean value of x
μ	Mean
σ^2	Variance
$\sum$	Covariance matrix
m	Estimated mean
s^2	Estimated variance
$\mathbf{S}$	Estimated covariance matrix
$N(\mu, \sigma^2)$	Univariate normal distribution with mean μ and variance σ^2
λ	Eigenvalue
e_t	Random noise, error
$trace(\mathbf{S})$	The trace of matrix $\mathbf{S}$
$\mathbf{T^2}$	Hotelling's $\mathbf{T^2}$ metric
$\mathbf{Q}$	Squared prediction error, Q-statistic

$\mathbb{R}^d$	d- dimensional Euclidean space
$\mathbf{Z}$	PCA transform variable
$\mathbf{X}$	PCA decomposition
$\mathbf{T}$	PCA score matrix
$\mathbf{P}$	PCA loading matrix
$\mathbf{I}$	Identity matrix
$\mathbf{Q}_\alpha$	Squared prediction error critical value
$H(S)$	Entropy of S
$Gain(S, A)$	Information gain of A relative to S
`TP`	True positive (*correctly classed as pass*)
`TN`	True negative (*correctly classed as fail*)
`FP`	False positive (*incorrectly classed as pass*)
`FN`	False negative (*incorrectly classed as fail*)

Abstract

With increased competition in the market place, it is essential that product quality and process performance are consistently competitive. Statistical Process Control (SPC) strives to differentiate between stochastic and assignable causes of variation.

This work outlines multivariate methods used for fault detection and classification in a high volume semiconductor device batch testing process. The routine capture of large quantities of test information and storage of historical results to databases occurs in most all processes. Subsequent data analysis and modelling is required for process monitoring, fault detection and classification.

Traditional exploratory methods of clustering and parallel coordinate analysis (parallel-coord) plots are used to demonstrate process contributions and diagnose out of control variables. They also serve as a low level monitoring scheme for process operatives and technicians. Principal Component Analysis (PCA) is applied to the semiconductor device batch test data as a dimension reduction method in order to represent the process through a reduced set of uncorrelated variables. The application of PCA to *mixed-mode* data, (*i.e.* analogue and digital variables), and the construction of a Normal Operating Condition (NOC) model is shown to offer fault detection and classification capabilities.

Supervised learning through decision tree induction is implemented with the batch test data for the purpose of fault classification. Use of the C4.5 tree induction algorithm is evaluated. Results for this nontraditional exploratory method are presented through confusion matrices.

In conclusion, the methods have been described in the context of semiconductor device batch testing but are widely applicable to other data rich environments.

Chapter 1

Introduction

Contents

"The time you enjoy wasting is not wasted time."
(Bertrand Russell, 1872-1970)

1.1 Quality Improvement

Increasing market competition and performance indices demand both lean manufacturing methodologies and a thorough knowledge of a process. It is paramount in this era of consumer driven quality for a company to be recognised as a leading quality provider. The competitiveness in the manufacturing sector, for one, has brought about the need for consistent production at lower costs. Quality Improvement, *QI*, is ubiquitous in process and product consistency and has provided the tools which enable such transformations.

It is difficult to encapsulate the differences that effectively discriminate processes and products of bad quality with processes and products of good quality. Statistical Quality Control (SQC), is infact a subset of a higher level philosophy known as Total Quality Management (TQM). This philosophy is expanded upon in table 2.1.1, but suffice it to say that TQM is essentially a framework of collective ideas, a synergistic set of ideals that spreads across an entire organisation. Caulcutt (1995) suggests that although the ideas that define TQM are well established, proper application of the principles result in a more coherent business.

There are a number of key buzzwords and acronyms used frequently within quality circles. Where one company is interested in implementing SQC, another may want Statistical Process Control (SPC). There is tremendous talk of TQM, Quality Function Deployment (QFD), Just-in-Time methodologies (JIT) and World Class Manufacturing (WCM). Although there is a link between quality and productivity, and hence profitability, any improvement requires an organisation or business to fully comprehend their processes and products needs. Two main components that dictate the success of these efforts are a systematic *reduction in variability* and a focus on *statistical methods* for process (and product) improvement. Clearly from this, it is difficult to succinctly define a TQM framework but in general it is concerned with people, processes and performance.

1.2 Industrial Application

In the semiconductor industry, success is a function of a product being unveiled swiftly and in accordance to market demands. In principle, this success is influenced by:

- Wafer defects
- Contamination
- IC manufacturing defects
- Design errors
- Handling problems
- Packaging problems

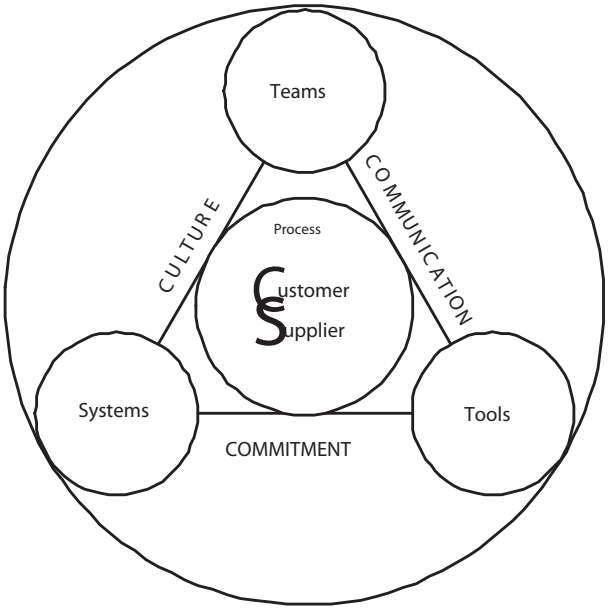


Figure 1.1. Total Quality Management model (Oakland (1999))

- External influences
- Process variations
- Product variations

A systematic long term goal for this work was an improvement in the yield of the devices under test (DUT), specifically for First-Pass Yield, (FPY). Improvement in this area could result in positive knock on effects such as reducing re-test batches, batches on hold, scrappage and being able to increase tester capacity. The old adage *'time is money'* is truly relevant when it comes to the testing of electronic components.

A simplified view of yield is a *pass* statistic on a batch. This can change significantly depending on process type, DUT type and tester programme. The main hypothesis of this thesis is to capture process variation and determine optimal operational characteristics. Prior analysis was preformed at a *high level* where the yield indices and batch statistics were generated both in real time and on completion of each batch. This approach is useful in providing information on high (or low) yielding testers, high (or low) yielding products, product test

Product Attributes	Process Attributes
	Site-to-Site
Wafer-to-Wafer	Socket-to-Socket
Lot-to-Lot	Picker-to-Picker
Batch-to-Batch	DIB-to-DIB
DUT-to-DUT	Handler-to-Handler
	Tester-to-Tester

Table 1.1. Product and process variation. *Some common attributes that constitute variation (both product and process). This table gives an insight to some of the variation sources in the testing process.*

distributions, pareto of failures, cross platform correlations and expected lot return. This method does not provide information on the variables responsible for the process, and hence any process disruptions. *Low level* monitoring of the process is achieved through multivariate methods that capture any process trend from the raw data itself. In the context of this work, no explicit feedback link is available to the tester to allow for setpoint adjustment or process tuning. The test programs were static and derived off line in accordance to a set of testing protocols, which are either derived *in-house* or vendor supplied.

As the process is a complex entity, splitting it into sub-processes, which are easier to study enables a breakdown of the variation sources. This *low level* sub-process analysis is shown in Table 1.1.

1.2.1 Process Description and Data Collection

The data were extracted from the mainframe UNIX/AIX servers and initial analysis suggested it contained both *continuous (analogue)* and *discrete (digital)* test variables. Initial tests were carried out under *'hand plug mode'*, where a DUT is placed into the device interface board (DIB) by hand and testing is performed off line. This has the advantage of bypassing the DUT handler and loading trays, thus minimising mechanical variations. The process of testing a randomly selected DUT was repeated a total of five times. This method is closer to a simulation and so is not totally representative of a generic batch but it is a good and reliable

method to extract useful data from the testers for presentation, analysis and a insight into potential sources of variability. The data format from the tester is a commonly used standard in the semiconductor industry, Standard Test Data Format (STDF). This proprietary format is specific to each tester and *a priori* preparation is essential. The **MST1C** semiconductor device was chosen due to its high volume nature and analogue and digital test vectors.

1.3 Problem Solving Methodology

Prior to any analysis, it is important to have a problem solving protocol in place. This specific protocol was similar to the one set out in Montgomery & Runger (2003),

Step 1 Develop a clear and precise description of the problem.

Step 2 Identify (tentatively), the important factors affecting the problem.

Step 3 Propose a model for the problem (stating limitation and assumptions).

Step 4 Conduct appropriate experimentation and/or collect data to validate the tentative model (as per Steps 2 and 3).

Step 5 Refine the model based on observed data.

Step 6 Manipulate the model to assist developing a solution.

Step 7 Test the proposed solution.

Step 8 Infer and conclude based on the solution.

This proposed framework summarises the basics of statistical thinking when it comes to analysing a system or process. Deming (1993) uses the words *'process'* and *'system'* interchangeably. They can be differentiated with respect to their boundaries, but essentially they amount to the same thing, *i.e.* a network of interdependent components which must work together to try and achieve common aims. The 'interdependent' reference hints at the complexity of components that make up systems, such as handlers, testers, lot feeders and telemetry infrastructures.

1.4 Statistical Methods in Industry

There are many algorithms capable of performing SPC, fault detection and classification and pattern recognition. An important subset of these techniques is revealing features that were previously unexpected and being able to classify or predict instances on a model based approach. The routine capture of high volumes of operational data has become common place in many organisations with the decrease in cost of data storage devices and the increase in levels of computing power. Historical data are becoming more complex and this has led to the development of more efficient and robust techniques for data analysis. The process of using historical data to discover regularities and improve future decisions is a machine learning area commonly called *Knowledge Discovery in Databases (KDD).* This phrase was coined by Piatetsky-Shapiro (1991) whilst describing analytical data analysis and knowledge extraction on large volume databases. The term *Data Mining* is often used in place of KDD and has very similar connotations. The subtle difference is however, data mining is an application of specific algorithms for pattern extraction from data and is therefore a step in the KDD process. KDD is more focussed on an entire framework *i.e.* where data are stored, accessed, analysed, modelled and presented. Fayyad et al. (1996) introduce KDD as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. Machine learning paradigms can be separated into two main areas

Unsupervised Learning *'Learning without a teacher'.* This offers the possibility of exploring the data without guidance in the form of class information and the aim is to establish the existence of classes or clusters in the data.

$$D\left(\mathbf{x}\right) \rightarrow f(x)$$

Where $\mathbf{x} = \{x_1, x_2, ..., x_p\}$ are variables in $\mathbf{x}$-space and D is a distance metric. Clustering, dimensionality reduction, feature selection and anomaly detection are examples of unsupervised machine learning techniques.

Supervised Learning *'Learning with a teacher'.* The training data are accompanied by labels indicating the class. The new data is classified based on the training set.

$$(\mathbf{x},\ f(\mathbf{x})) \rightarrow \widehat{f}(\mathbf{x})$$

Where $\mathbf{x} = \{x_1, x_2, ..., x_p\}$ are inputs and $f(\mathbf{x})$ is an output (or class label). The objective is to determine a good approximation to $\widehat{f}$. Classification, regression trees and neural networks are supervised machine learning techniques as they require class patterns with known class assignments. Classification is used in the prediction of categorial class labels (either discrete or nominal) and regression is used for continuous-valued functions.

1.5 Thesis Aim

The aim of this thesis is to present a detailed study on multivariate statistical methods used for fault detection and classification in the semiconductor device testing industry.

1.6 Thesis Overview

- Chapter 1: Introduction. This chapter presents an overview of the industrial background of this thesis. It also introduces concepts of Quality Improvement, statistical methods used in industry and data analysis paradigms.

- Chapter 2: Literature review. This Chapter presents a philosophy of quality for both processes and products and gives a descriptive overview of variation. Statistical methods in process control are described and the distinction between unsupervised and supervised methods is discussed.

- Chapter 3: Unsupervised Learning Methods. This Chapter outlines the use of unsupervised machine learning methods such as Clustering, Parallel coordinates analysis and Principal Component Analysis. A mathematical introduction to PCA and the concept of multivariate fault detection is presented also.

- Chapter 4: Supervised Learning Methods. This Chapter outlines the use of supervised machine learning methods such as One-Rule (1-R) classification and Decision Trees induction. Classifier performance is presented through a confusion matrix.

- Chapter 5: RESULTS. This Chapter details the application of semiconductor batch test data to both unsupervised and supervised exploratory methods. It provides a description of parallel-coord plots for process visualisation. The development of a Normal Operating Condition (NOC) model is detailed as is fault detection and classification score plots and multivariate indices.

- Chapter 6: DISCUSSION This Chapter illustrates how the multivariate tools and analysis techniques are used in the context of this thesis. Two process states are described and discussed through parallel-coord Figures, PCA score plots and contribution plots.

- Chapter 7: CONCLUSION This Chapter concludes and summarises points made throughout this thesis. It also identified areas of future work.

- BIBLIOGRAPHY

- Appendix A PRINCIPAL COMPONENT LOADINGS

- Appendix B CUMULATIVE MODEL VARIANCE

- Appendix C PC1-PC2-PC3 NOC ELLIPSOID WITH SUBPLOTS

Chapter 2

Literature Review

Contents

"Quality is not an act, it is a habit." (Aristotle, 384BC-322BC)

2.1 The Philosophy of Quality

The study of quality is not as recent a phenomenon as one might imagine. Quality concepts and ideas have significantly influenced and helped develop both mankind and the environment. The fundamental underpinnings of quality, it seems, have been present (albeit latent), in many early product development and processing techniques. Table 2.1 briefly outlines the important chronological developments. For a further, more detailed synopsis see Montgomery (2001).

Milestone	Year
European guilds maintain standards	1700-1800
Industrial revolution brings change	1800-1900
AT&T begin product testing & inspection	1907-1908
Shewhart pioneers control charting	1924
Dodge *et al.* refine acceptance sampling	1928
Wald develops sequential sampling	1942
Deming in Japan delivering SQC seminars	1946
Taguchi pioneers experimental design	1948
Page develops CUSUM chart	1954
Roberts develops EWMA chart	1959
Zero-defects model popularised in America	1960's
Introduction of Total Quality Management (TQM)	1970's
Taguchi methods introduced to companies	1980
Widespread use of experimental design methods in Japan	1980's
Emergence of Statistical Process Control (SPC)	1980's
Motorola initiates Six-Sigma (6σ) thinking	1989
Quality standards formalised	1990's
Extension to Multivariate SPC	1995-today
Chemometric & Process Analytical techniques devised	1996-today
6σ Methodology for Industry Standard	2000-today

Table 2.1. A history of quality.

Quality was largely determined by the efforts of an individual craftsman, Eli Whitney. In 1794 he influenced the American mass-production concept with his system of standardised, interchangeable parts whilst manufacturing muskets for the government. Frederick W. Taylor introduced some principles of scientific management as mass production industries began to develop prior to 1900. His main area of interest was in the improvement of productivity, and he pioneered dividing work into tasks so that the product could be manufactured and assembled with ease.

During the 1920's, Walter A. Shewhart of Bell Telephone Laboratories (Bell Labs) pioneered the use of statistical techniques for monitoring and controlling quality. Bell laboratories wanted to economically monitor and control the variation in quality of components and finished products. Shewhart recognised that inspecting and rejecting (or reworking) a product was not the most economical way to produce a high quality product. He demonstrated that monitoring and controlling variation throughout production was a far superior way of controlling quality. Shewhart invented a visual tool for monitoring process variation which he called control charts. These subsequently became Shewhart control charts in honour of their inventor. In the 1930's, the United States telecommunications industry operated by Bell Labs was recognised as the international standard for quality, with a great deal of credit due to Shewhart for his control charting techniques.

At this time it was realised that SPC methodologies were of importance, and hence, they began to spread to other industries, including the Bureau of the Census. With the onset of World War II, the need for high volume and consistently high quality armament production arose. To assist these efforts, Bell Labs offered training in SPC techniques at Universities such as Stanford and Columbia. It was around this time that the first quality control (QC) journal, Industrial Quality Control, was published (1944), and a QC professional society was formed, namely the American Society for Quality Control (ASQC) in 1946. The ASQC is now known as the American Society for Quality (ASQ). After the war, there followed a period where Statistical Process Control (SPC) methodologies were 'laid off' and the rather rigorous quality control tools were eased. With a stable economy, and a high demand for consumer goods, many manufacturers felt little need to invest into SPC techniques. In the United States, it remained like this for several

years postwar, and indeed the growth of SPC outside of the defence industry was sluggish at best. In Japan, the postwar situation was different. Since the economy had been devastated during the war, the industries required rebuilding from the ground upwards. In the late 1940's, Joseph M. Juran and W. Edwards Deming, both understudies of Shewhart, traveled to Japan. Juran's mission was to educate the Japanese on quality management and Deming's to help with the census. They had found little interest amongst U.S. companies with their quality-management philosophies, but the Japanese welcomed them Juran (1988). The 1950's and 1960's saw the emergence of reliability engineering, the introduction of several important textbooks on statistical quality control and the viewpoint that quality is a way of managing an organisation. The impact of quality management on the Japanese industry was phenomenal, and thus the same philosophies, previously overlooked, were adopted as a matter of urgency to compete with the quality conscious Japanese.

2.1.1 The TQM Philosophy

Statistical Process Control (SPC), is a high level abstraction of the Total Quality Management (TQM) philosophy. Deming (1993), with his *'system of profound knowledge'*, alluded to how complex organisations operate, and that an understanding of this complexity can bring long-term improvements in quality and efficiency. The so called four pillars of profound knowledge are:

Appreciation for a system Organisations are interactive systems and must be managed as systems. Management has a role in plant wide optimisation.

Knowledge about variation Variation is always present. The most important thing is not to just measure and quantify it, but also to understand it.

Theory of knowledge Increase ones knowledge of the way a system (or process) operates. This can be achieved through the Plan-Do-Check-Act cycle.

Psychology A basic knowledge of human interaction and behaviour in different situations and the ability to extract the most from a person is essential.

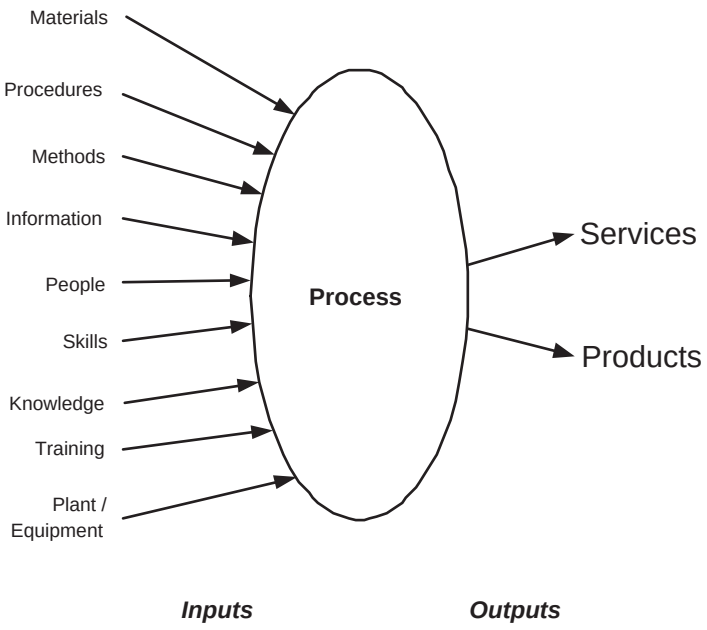


Figure 2.1. Definition of a process (Caulcutt (1995))

2.2 Process Quality and Product Quality

In order to define the nature of the quality methodology, a distinction has to be made with regard to process quality and product quality. Caulcutt (1995) describes a process as a collection of inputs and outputs. This structure is shown in Figure 2.1, where there are multiple inputs, a process and outputs. The basic purpose of a quality system is quality planning, quality control and quality improvement. Process quality can be conceptualised through 11 process dimensions (Table 2.2). These reflect the quality of the technological process, or the means by which inputs are transformed into desirable outputs. A transformation can be defined as the process in which an input is remodelled into a desirable output. Misterek et al. (1990) suggests that process quality is a state of superior performance on a number of process dimensions. In a more recent paper, Dooley et al. (2000), the authors cite the importance of a process quality knowledge base (PQKB). The PQKD is subsequently defined as an information system that acquires process knowledge and represents it for the purpose of quality performance, quality control and quality improvement.

Table 2.2. *The Eleven Dimensions of Process Quality, Misterek et al. (1990)*

Capability	*The ratio of the width of specification limits on a characteristic relative to the natural variation of that characteristic*
Stability	*The predictability of a process*
Requisite Variety	*A process under differing environments*
Robustness	*Sensitivity of a process to different environment conditions*
Flexibility	*The variety of outputs available*
Reliability	*The frequency with which a process fails*
Serviceability	*The effort required to repair a process*
Efficiency	*The addition of value to the output*
Technical	*The level of technical sophistication required*
Compatibility	*Comparison between processes within the same operational environment*
Safety	*The risk of harm to process operators*

Product quality is a parametric entity and many quality practitioners have sought to identify the key characteristics. The byproduct of this has created a significant amount of information in literature. While the notion of quality is multi-dimensional, normally it is perceived that quality represents *"fitness for use"* Juran (1988), is *"inversely proportional to variability"* Montgomery (2001), represents *"value"* Feigenbaum (1991), *"conformance to requirements"* Crosby (1979) and *"meets expectations"* Buzzell & Gale (1987). More recently Stone-Romero et al. (1997) defined product quality as a perception by customers based upon

Table 2.3. *The Eight Dimensions of Product Quality, Garvin (1987)*

Performance	*The primary operating characteristics of a product*
Reliability	*The frequency of failure of a product*
Durability	*The length of time before a replacement product*
Serviceability	*The ease of product failure repair*
Aesthetics	*The appearance of a product*
Features	*The secondary characteristics of a product*
Perceived Quality	*Indirect association: brand name, image, advertising*
Conformance	*The degree in which the characteristics of a product fall within specified limits*

their appraisal of four criteria: flawlessness, durability, appearance and distinctiveness.

Garvin (1987) in his review of available literature, succinctly identified five approaches in defining product quality, and subsequently extracted eight dimensions, which he considered to be the basic elements of product quality. These eight dimensions are shown in Table 2.3.

2.3 Variation

The inherent or natural variability in any process is the cumulative effect of many small, unavoidable and difficult to identify stochastic or common causes. A process is in control or more formally in a *'State of Statistical control'*, when only stochastic causes are present, Montgomery (2001). A process is out of control when the constant system of common causes changes, or when any additional source of variation is temporarily present. The sources of variation that are not part of the stochastic variation pattern are known as assignable causes of variation and mainly arise from operator errors, defective products, raw materials or improperly functioning processes. A process that is in control is a stable process but this does not necessarily mean that the output is within the specification

limits for the process. Shewhart (1986) introduced a theory of variability to describe the differences between these two causes.

The stochastic cause/assignable cause paradigm is often difficult to distinguish between in practice as assignable causes will appear, seemingly at random. A fundamental objective of statistical process control is to readily identify the occurrence of any assignable causes of variation (*e.g.* a process shift) so that a corrective action can be applied in order to alleviate the variation. A statistical hypothesis is used, formulating a statement about the probability distribution of a random variable or indeed, about the parameters of one or more populations. This creates two decision based errors, namely type I and type II, associated with the corrective action:

Type I Decision Error (False Alarm) This occurs when the process is judged to be out of control when in fact it is in control.

Type II Decision Error (Failed Alarm or *miss*) This occurs when the process is judged to be in control when in fact it is out of control.

The probability of making a type I error is denoted by α,

$$\alpha = P\{\text{type I error}\} = P\{\text{reject } H_o | H_o \text{ is true}\}$$

The probability of making a type II error is denoted by β,

$$\beta = P\{\text{type II error}\} = P\{\text{fail to reject } H_o | H_o \text{ is false}\}$$

Where H_o is the null hypothesis. Both types of error are typically associated with process disruption and economic loss. Sometimes it is convenient to report the *power* of a test, where the power can be interpreted as the probability of correctly rejecting a false null hypothesis. This is calculated from 1 - β,

$$Power = 1 - \beta = P\{\text{reject } H_o | H_o \text{ is false}\}$$

A summary of hypothesis testing type errors is shown in Table 2.4. The confusion matrix, 4.6, outlines this concept in a machine learning framework. `Type I errors` are `False Positive` entries and `Type II errors` are `False Negative` entries in a confusion matrix.

Decision	H_o **Is True**	H_o **Is False**
Fail to reject H_o	`No error`	`Type II error`
Reject H_o	`Type I error`	`No error`

Table 2.4. Decisions in hypothesis testing. *In this decision matrix, 2 instances show no error, i.e. a correct decision and the remaining 2 show different types errors.*

2.3.1 Probability Distributions

A probability distribution is a mathematical model that relates to the probability of occurrence of a particular variable in a population. Two types of probability distributions are described:

- **Continuous distributions** Where the variable measured can be expressed on a continuous scale.
- **Discrete distributions** Where the variable measured can only take on certain values, such as integers *0,1,2..,k*.

A probability density function (PDF) commonly used to describe a continuous process is the normal distribution or Gaussian. If x is a normal random variable, the normal distribution is described as:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \qquad -\infty < x < \infty$$

where μ is the mean and σ is the standard deviation. The normal distribution is used very frequently and has a special notation, $x \sim N(\mu, \sigma^2)$, which suggests that x is normally distributed with a mean of μ and a variance of σ. In practice, it is often convenient to convert a normal distribution into a standardised normal distribution with zero mean and unit variance, $x \sim N(0, 1)$, through the standardised normal variate, $z = \frac{x-\mu}{\sigma}$. The exponential term thus changes to $e^{\frac{-z^2}{2}}$.

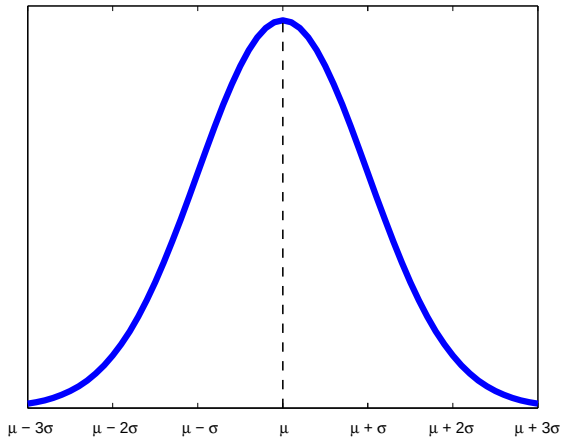


Figure 2.2. A normal distribution. *This shows the area under the normal 'bell-shaped' curve representing the probability of a value being between $\pm\sigma$, $\pm 2\sigma$ and $\pm 3\sigma$. A précis is shown in Table 2.5.*

Table 2.5. *The normal distribution with σ levels*

σ Level	Probability	Percentage
$\pm\ \sigma$	0.6827	68.27 %
$\pm\ 2\sigma$	0.9545	95.45 %
$\pm\ 3\sigma$	0.9973	99.73 %

In contrast, when an outcome is classed as either a *'success'* or a *'failure'*, a binomial distribution is used. The binomial PDF is described as:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x} \qquad x = 1, 2, ..., n$$

which is interpreted as the probability of obtaining x outcomes from n independent events. The mean is np and the variance is $np(1-p)$.
$\binom{n}{x}$ is combinatorial notation for $\frac{n!}{x!(n-x)!}$.

2.4 Statistical Process Control

Statistical methodologies used in process control form both process performance monitoring and fault detection schemes. A high level objective of SPC is to

monitor the performance of a process over a period of time and verify that the process remains in a 'state of statistical control' as defined by a certain limit or statistical hypothesis. SPC is used to detect an assignable cause deviation so that a corrective action may be taken before quality is adversely affected. In general, the philosophy of SPC is to accept variation as inevitable, whilst recognising that quality improvement is more heavily skewed towards defect prevention than defect detection. SPC uses statistics in the collection, processing and analysis of data in order to achieve and maintain control of variation. Thus, an understanding of variation is highly significant in the comprehension of SPC.

SPC is a collection of problem solving tools used in quality improvement. Montgomery (2001) and Oakland (1999) outline seven major tools used for this process, commonly quoted as the *'Magnificent Seven'*:

1. Histogram
2. Check sheet
3. Pareto diagram
4. Cause and Effect diagram
5. Defect concentration diagram
6. Scatter diagram
7. Control chart

Traditional SPC is achieved through univariate monitoring of process parameters and the most commonly used tool is the control chart. Specific charts such as Shewhart charts, cumulative sum (CUSUM) charts Woodward & Goldsmith (1964), exponentially weighted moving average (EWMA) charts Roberts (1959) and Hunter (1986) and range/dispersion (R) charts are used for slightly different purposes. These charts are conventionally applied to process data under normal process (steady state) conditions where the process mean, μ, and standard deviation, σ, are monitored. Changes or deviations in either parameter may indicate an out of control condition. There is an assumption of normality and independence in the underlying theory of control charting, *i.e.* the data come from a

normal distribution and are independent and identically distributed (*i.i.d.*) random variables. Autocorrelation in a data structure can frequent the presence of false alarms, a Type I error, especially when there tends to be a large number of variables that affect the overall quality of the process. The central limit theorem (CLT) states that if a large number of independent random variables are added together, the sum is normally distributed under general conditions contrary to the distributions of the original variables. Sachs et al. (1995) define SPC as a binary view of the condition of a process, *i.e.* either the process is within specification limits or it is not. An out of control condition may be caused by a single variable or multiple variables.

2.4.1 Control Charts

Shewhart (1986) introduced a theory of variability to describe his control charting technique. This theory refers to the normal variability, or noise, that occur within a system. Through a series of equations, the Shewhart chart has the ability to translate all the dependencies of a system into one compact analytical tool. Shewhart (1986) outlined the following relationship:

$$x_t = \mu_0 + e_t \tag{2.1}$$

where x_t is the process mean at time t, μ_0 is the in control process mean of the sample and e_t is the random noise of the sample at time t. The relationship in Equation 2.1 enabled Shewhart to develop charts capable of plotting variability within a system. This type of control chart is applied to the monitoring of variables or quality characteristics that follow a normal distribution. When variables have numerical measurements on a continuous scale, $\overline{x}$, σ and R charts can be used. Variables not readily represented on a continuous scale are classed as either *'Conforming'* or *'Non-Conforming'* and quality characteristics of this type are known as attributes Montgomery (2001). A common example of such an event is the proportion of failed semiconductor units upon completion of a test process run. There are four types of charts used in the monitoring of attributes, the p chart (fraction of defects), np chart (number of defects), c chart (number of defects) and the u chart (number of defects per unit). Attribute control charts are applied to data that follow a discrete distribution. Fraction non-conforming and

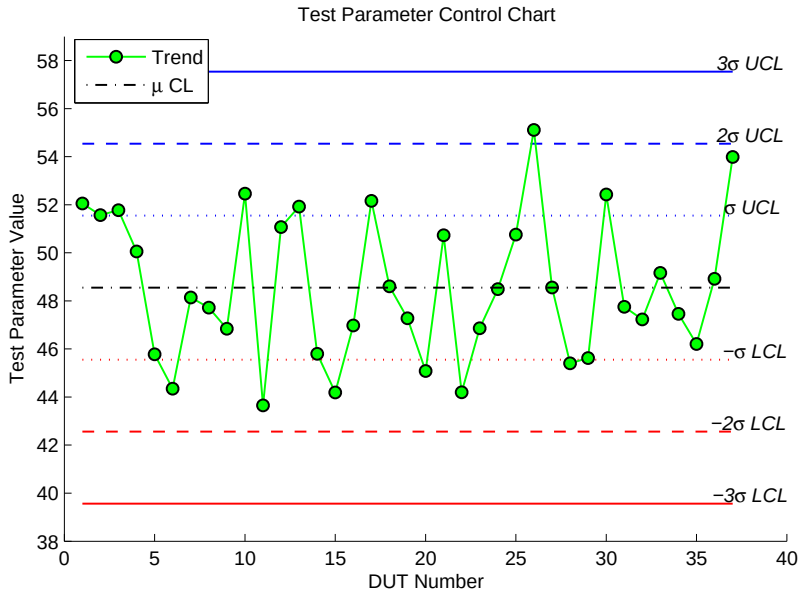


Figure 2.3. Variable Control Chart. *Test parameter trend showing μ-centreline and σ-Upper Control Limit (UCL) and σ-Lower Control Limit (LCL) levels. Only 1 DUT breaches the 2-σ UCL in this case.*

fraction conforming charts and indices are used frequently in the calculation of First Pass Yield (FPY). FPY is simply the ratio of failures to total batch starts and is often quoted as percentage yield:

$$[1 - \frac{Failures}{Entire \quad batch}] * 100$$

Monitoring FPY is a high level solution to a low level problem and is often used as an economic index to describe a process.

Space precludes the inclusion of the entire complement of SPC charts, but Figure 2.3 outlines a control chart calculated from μ test data. This is a univariate control chart, only showing one process parameter point outside of a $\pm 2\sigma$ control limit. A univariate chart, such as Figure 2.3, is useful to determine variation in a process parameter and track changes in the process. Its usefulness, however, decreases as the amount of process parameters per test operation increases.

2.5 Multivariate Statistical Process Control

Multivariate Statistical Process Control (MSPC) is simply a multivariate extension of SPC. MSPC is increasingly recognised as a useful tool for providing an early warning of process changes, assignable plant faults, process disturbances and for giving the analyst a deeper understanding of the process. Computing technology, with improved data logging and analysis features, has increased the importance of multivariate analysis whereby the main objective is to identify the root causes of variation and take corrective action. Martin et al. (1999) use the term Multivariate Statistical Process Monitoring (MSPM) in place of MSPC, which conceptually gives a more appropriate description of the monitoring aspect of process control strategies. Kourti et al. (1996) state that univariate SPC procedures are inadequate for most modern industrial processes, as they are traditionally based on charting only a small number of final product quality variables. In Kourti (2002) the problem with monitoring a complex process through a univariate control strategy is illustrated. The author states that many industries made the transgression to MSPC a decade ago as monitoring a small number of variables (usually the final product quality variables) is totally inadequate and does not explain the underlying process fault conditions and emergence of abnormal situations. Most industrial processes generate massive amounts of data on hundreds of process variables through various sensors arrays. The univariate SPC methodology of examining each variable singularly and independently makes the interpretation and diagnosis of a fault condition very difficult and convoluted MacGregor & Kourti (1995). This method only considers the magnitude of deviation inherent in a single process variable independently of all other process variables. Simultaneous monitoring of individual variables separately will fail to recognise possible cross-correlations that may exist and will increase the insensitivity of the control charts for detection of out-of-control conditions, Samanta (2001). This can be quite misleading as not all the variables are independent and only a few underlying events are driving the process at any one time. The final product quality is defined by the simultaneous correct values of each variable, and therefore is a multivariate property. In summary, the product quality is a logical *'AND'* of the test metrics.

With this in mind, it becomes apparent that to monitor a reduced set of

variables for a particular process, fault conditions and abnormal situation can be detected more accurately.

Process monitoring is usually conducted at two levels, Marlin (2000). The first level is the immediate operation of the process by test operatives and technicians and the second is long term performance analysis by test engineers. Long term performance degradation is more difficult to diagnose than sudden process failure and is often dictated by the quality and quantity of historical data and any methods applied to extract relevant features.

When the data being analysed are large and correlated, the signal-to-noise ratio (SNR) is low in each variable. One of the main objectives in data compression is to enhance the SNR, *i.e.* to minimise the noise terms which are assumed to be totally stochastic components.

2.5.1 Process Operation Data Characteristics

Data characteristics are vitally important in both process description and modelling. Wang (1999) lists the basics characteristics of process data where data volume, data dimension, uncertainty, noise, dynamic trends, sampling frequencies, data redundancy and complex interactions are important.

2.5.2 Classification of Batch Processes

Most industrial processes consist of three main components, *input*, *processing* and *output* and control procedures may be established on any or all of these process components. A batch process, by definition, assumes a batches as an input, Fuchs & Kenett (1998). Batches can also occur when a process is changed at a specified time intervals, giving rise to the nature of temporal batch data. This is common in many industries, *i.e.* process, pharmaceutical, microelectronic and chemometric. Temporal batch data can arise when a process is altered or reset. The relationships between process variables are maintained however, until shifted to allow for another process configuration. Mason et al. (2001) define *Category 1* and *Category 2* batch processes. A category 1 (Cat1) batch process has limited between batch variation and observations are assumed to come from the same d-dimensional normal distribution, $N_d(\mu, \sum)$ with a common mean vector, μ, and covariance matrix, $\sum$. A category 2 (Cat2) batch process operates

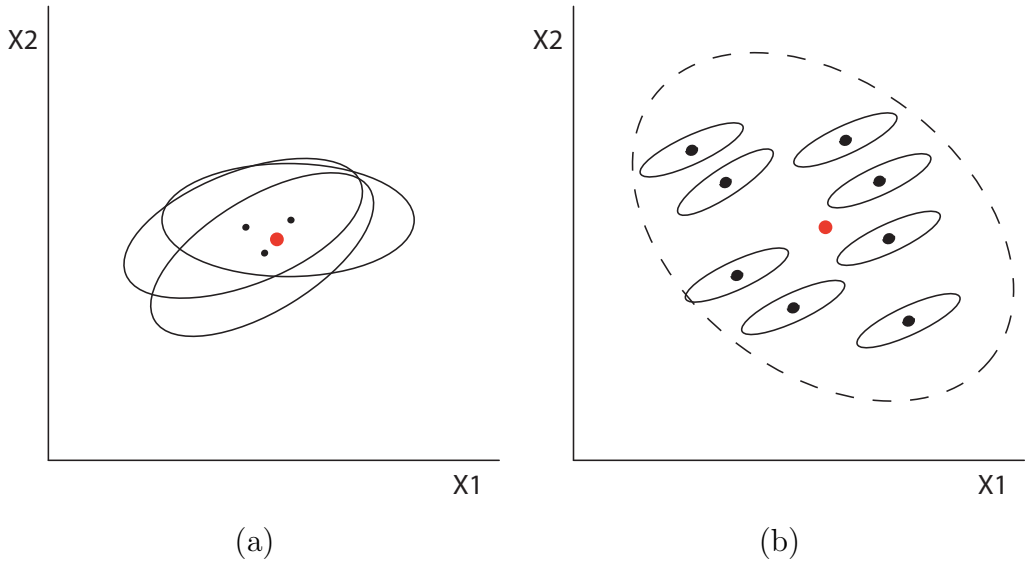


Figure 2.4. Control ellipsoids. *The elliptical regions represent in-control data from separate batches, with the '•' points representing each mean and the '●' points represents the overall mean for* (a) *category 1 batch process* and (b) *category 2 batch process.*

with a significant separation between batch mean vectors. The observations are assumed to come from different multivariate normal distributions, $N_d(\mu_i, \sum)$, where $\mu_i, i = 1, 2, ..., k$, represents the population mean of the i^{th} batch. The difference among the batch means may be due to *known* of *unknown* causes. Figure 2.4 details each categorical condition. Figure 2.4 (a) shows in control production regions for batches taken from a Cat1 process containing two process variables, X_1 and X_2. The ellipsoids represent in control data from three separate batches whose individual mean vectors are shown by '•'. Overall, the mean is shown by '●'. It is clear to observe that the individual batch means are close in proximity to the overall global mean indicating a closeness between the batches. In Figure 2.4 (b), the small ellipsoids represent the in control process regions of individual batches while the larger ellipse represents the overall global mean. This shows several batches, whilst maintaining in-control relationships between the process variables, may be out of control relative to the global mean.

2.6 Unsupervised Learning Methods

The unsupervised learning methods of clustering and Principal Component Analysis (PCA) were used to analyse and reduce the dimensionality of the test data. These methods are discussed in Chapter 3.

2.6.1 Principal Component Analysis

Principal Component Analysis (PCA), is a key technique in the analysis of multivariate data. It encompasses three possible objectives: description, interpretation and modelling of the data, Krzanowski (2000). The methodology of PCA is reviewed in Section 3.2.2 and is comprehensively described in Jackson (1991) and Jolliffe (1986). Lane et al. (2001) develop a multi-group process representation that overcomes the problem of having a single model for every product. In many industrial applications, the Principal Components (PCs) actually have physical interpretation and thus can be used as control variables in their own right. Jackson (1991) used PCA to examine audiometric data from a large number of employees. The experimentation, while neither process or product related, made use of the data reduction feature inherent in PCA. PCA is used extensively within the Chemometrics field, Wold & Sjöström (1998), Ku et al. (1995) and Wise & Gallagher (1996), Semiconductor Etch and Monitoring Wise et al. (1999), Goodlin et al. (2002) and Skinner et al. (2002), Multivariate Statistical Process Control Kourti (2002), Wise et al. (1999), Lane et al. (2003), Simoglou et al. (2000), Exploratory data analysis in the food industry Pravdova et al. (2002) and Penza et al. (2001). Zuendorf (2003) detail the use of PCA in functional brain imaging in order to determine the patterns causing the greatest *variance* in the images. Asgharian & Hansson (2003) detail the use of PCA in determining factors that are important in calculating expected market returns and risk management of financial portfolios.

2.6.2 Monitoring Indices

An extension to the univariate control charting techniques is required to capture possible cross-correlations, simultaneously monitor many variables and provide an accurate fault index, Mason et al. (2001). If an anomaly is detected through

process monitoring then fault diagnosis should be performed to identify the cause (*source*) of the fault for corrective action. Process monitoring evaluates the operating condition of the process and assesses whether it is operating within a predefined operating region, Leung (2002).

2.7 Supervised Learning Methods

In contrast to latent variable modelling, a supervised learning technique known as decision tree learning was employed with the same test data. The decision tree algorithm provides a tree capable of classifying test data with a high degree of accuracy. It is then tested with unseen data (*i.e. data not used in the construction of the tree*) for the purpose of fault classification and the performance evaluation. This is dependent on the induction method and cross-validation. A detailed description of decision trees and supervised classification is given in Chapter 4.1. In short, a decision tree is represented as a multi stage decision system in which the classes are sequentially rejected upon the arrival of a feature vector until the destination class (terminal node) is reached. A decision tree is constructed by recursively partitioning a learning sample of data in which the class label and the value of the predictor variables for each case is known

- Each internal node tests an attribute
- Each branch corresponds to an attribute value
- Each leaf node assigns a classification

2.7.1 Neural Networks

System representation, modelling and identification are fundamental to process engineering and other problem domains. It is often required to approximate a real system with an appropriate model given an *input - output* data set.

An Artificial Neural Network (ANN or NN) is an information processing structure that largely mimics and has been inspired by the way in which biological networks (*i.e.* the brain) process information. NNs, like the human brain, have the ability to learn by example and by the surrounding environment, which in turn enables them to perform a classification or recognition task.

NNs have been successfully applied to a number of real-world problems of large scale complexity. Their success can be attributed to their ability to offer a complex non-linear solution to a given problem not suitably accommodated by an algorithmic solution, Ritter et al. (1992).

Wong et al. (1997) survey and review journal articles on neural networks in business applications. The application base is diverse, but the largest problem domain is in production/process operation and the second largest is in finance.

NN's have the to ability to present viable solutions to real-world problems, but in general form an appropriate part of a solution to a large scale problem. Noorossana et al. (2003) present a application of NNs to detection and classification of out-of-control signals.

Whilst supervised learning algorithms can perform well given adequate training data, they have a number of drawbacks. Typically, they require a large training data set. This is of little importance in the data-rich environments of micro-electronics, batch process control and time series data but generation of a suitable training data set can be costly. Furthermore, in general it is not possible to incrementally add to the training data whilst training the classifier. If the training data is changed in any way then the entire training data set must be used to retrain the classifier.

2.8 Chapter summary

This chapter introduces the concept of quality, in terms of both process and product. It also differentiates between stochastic and assignable variation. Statistical Process Control (SPC) is described along with its multivariate counterpart (MSPC). Univariate methods to monitor complex data rich processes are quite limited and the extension to multivariate methods give a better representation of the underlying process. Unsupervised and supervised machine learning methods are introduced and some examples are given.

Chapter 3

Unsupervised Learning Methods

Contents

"Make the model as simple as possible, but not simpler."

(Albert Einstein, 1879-1955)

3.1 Exploratory Data Analysis

Statistical methods of analysis are only capable of *'data modelling'* not *'actual process modelling'* and this is an important realisation in any model development. Simply put, models are mathematical abstractions of a system which can vary greatly in complexity and usefulness. The primary goal is to capture the behaviour and to organise this information into a concise set of rules or metrics.

In a multivariate framework, the structural complexity of a system is changed into a problem of high dimensionality. Abbott (1884) wrote of this very problem of dimensionality and visualisation in *'Flatland, A romance of many dimensions'*.

The *'curse of dimensionality'* Bellman (1961) is the description of a problem that occurs when working with data in a high dimensional space. The complexity is exponential to the number of dimensions, which in statistical terms are the degrees of freedom. This dimensionality problem is illustrated in Figure 3.1 where Euclidean $\mathbb{R}^3$ space is defined by (x_1, x_2, x_3). Modelling a non-linear relationship through a set of input variables $\mathbf{x_i}$ and output y can be achieved on the basis of training data and partitioning the sample space. This however, is not an optimal solution. Firstly, the input variable is split into a number of intervals so that the value of a variable can be specified by indicating in which interval it lies. This leads to the division of the entire input space into a large number of cells (or blocks). Each of the training examples corresponds to a point in one of the cells and carries with it an associated value of the output variable y. Given a set of new input vectors $\mathbf{x_{new}}$ the corresponding output y_{new} could be determined by finding which cell $\mathbf{x_{new}}$ falls in and returning the average for the the training points in that cell. Increased precision requires the increase of the number of divisions along each axis.

The fundamental problem is however, exponential growth of the sample space. If each input variable is divided into M divisions then the total number of cells is M^d, where d is the dimension. With limited quantities of data, increasing the dimensionality of the space rapidly leads to sparsity where the input-output mapping is very poorly represented. Figure 3.2 represents a input space with 10 divisions in 3 dimensions, which has 10^3 cells.

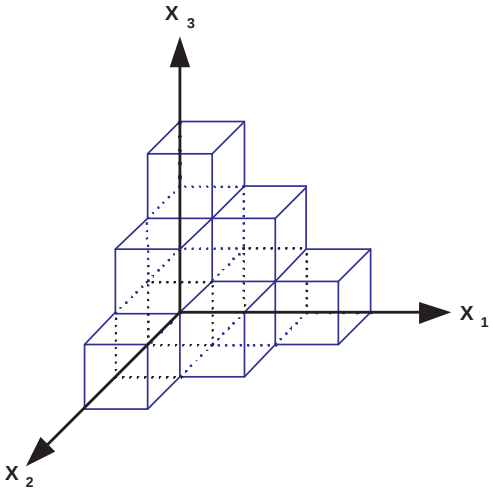


Figure 3.1. d- dimensional Euclidean space $\mathbb{R}^d$. *One method of mapping a d-dimensional space,* $(x_1, x_2, ..., x_d)$*, to an output variable y is to split the space into a number of cells (or blocks) and specify a value of y for each of the cells. One major drawback of this method is the exponential growth of the space with respect to d.*

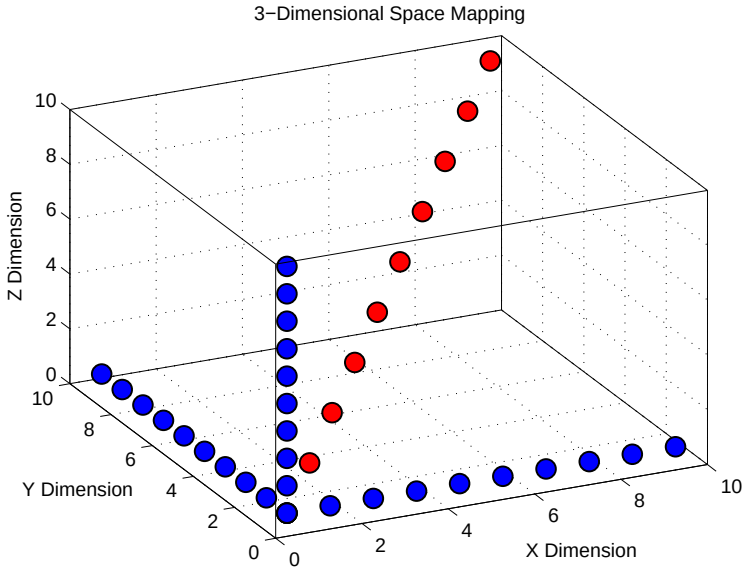
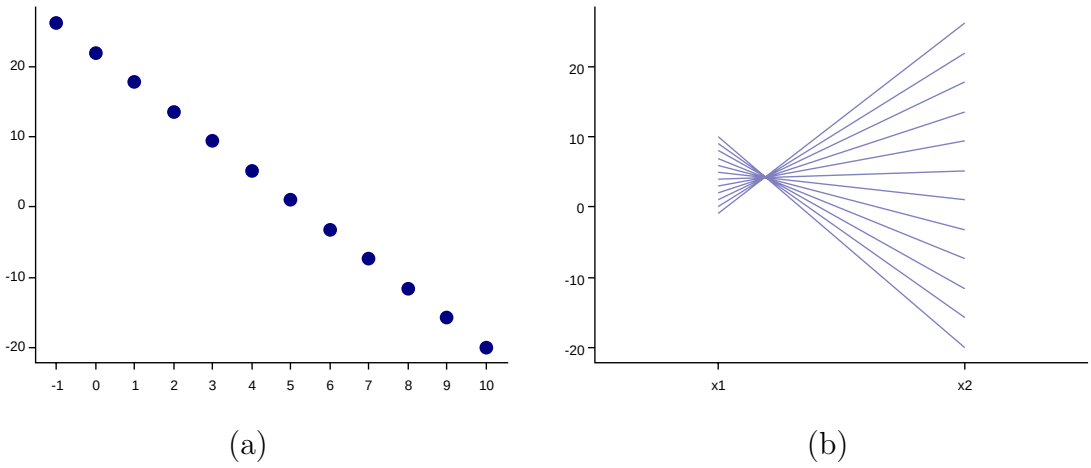


Figure 3.2. Feature Space Mapping. *Representing* $\mathbb{R}^3$ *space through a* $(10 \times 10 \times 10)$ *feature map. Cells are shown for the* x*,* y *and* z *axes and along the principal diagonal. For any* (x_i, y_j, z_k) *triplet, an output* y *is available.*



(a) (b)

Figure 3.3. Methods of data representation in $\mathbb{R}^2$. (a) *Cartesian* (b) *Parallel Coordinate Plot.*

3.1.1 Parallel Coordinates Analysis

Parallel Coordinates Analysis (parallel-coords) was introduced by Inselberg (1981), in an attempt to rationalise high dimensional data structures. Parallel-coords is a two-dimensional technique for multidimensional data visualisation. It has properties of low representational complexity, uniform treatment of all variables, works for any d-dimension and the display intuitively conveys information about the d-dimensional object it represents. Parallel-coords is an exploratory technique without any *a priori* bias. Any d-dimensional tuple $(x_1, x_2, ..., x_d)$ can be visualised as a *polyline* in parallel-coords, connecting the points $x_1, x_2, ..., x_d$ in d parallel ordinates.

Figure 3.3 shows the representation of two vectors, $\mathbf{x_1}$ and $\mathbf{x_2}$, in both cartesian and parallel-coords. This is a trivial example, but it does convey the latter's ability to extend beyond $\mathbb{R}^2$. Figure 3.3 (a) shows a cartesian or scatter plot of the two vectors. Figure 3.3 (b) shows two ordinates along the abscissa (x-axis) each displaying a vector. The transition from $x_1 \rightarrow x_2$ is through a polyline and each polyline passes through an axis at a location that indicates the observation's value relative to all other values. Inselberg (1997) introduces the idea of parallel-coords as a space efficient method for representing large data sets and modelling relations between variables.

Parallel-coords and other clustering methods can break down a high dimen-

sional space into correlated subsets that are more readily analysed.

3.1.2 Cluster Analysis

Cluster analysis is a pattern search method used in multivariate data structures and it can be applied to data that is quantitative (*numerical*), qualitative (*categorical*) or a mixture of both. The goal of clustering is to find an optimal grouping for which the variables within each cluster are similar but the clusters are dissimilar. This, in effect, has the tendency to find the natural groupings within the data which can be of significant benefit to the researcher or analyst. Clustering is a method where each data point is associated with its next closest point and likewise onwards until a specified amount of cluster centres are formed. There are many different clustering methods which attempt to identify and group similar data. It is necessary to have a measure of similarity or distance between the vectors in order to achieve this. Since distance increases as two points diverge, distance is actually a measure of *'dissimilarity'*, *i.e. distance* $= similarity^{-1}$. Assuming a $\mathbb{R}^d$ Euclidean space, the distance $d(\mathbf{x}, \mathbf{y})$ between two vectors $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ and $\mathbf{y} = (y_1, y_2, ..., y_n)^T$ can be defined using one of the following distance measures

$$\text{Euclidean} \quad = \sqrt{\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{y}_i)^2} \tag{3.1a}$$

$$\text{Manhattan} \quad = \sum_{i=1}^{n} \mid (\mathbf{x}_i - \mathbf{y}_i) \mid \tag{3.1b}$$

$$\text{Minkowski} \quad = \left[\sum_{i=1}^{n} (\mid (\mathbf{x}_i - \mathbf{y}_i) \mid)^p \right]^{\frac{1}{p}} \tag{3.1c}$$

The Minkowski metric, Equation 3.1 c, becomes the Euclidean distance when $p = 2$ and the Manhattan or *'city block'* distance when $p = 1$. The distance is called a metric if it satisfies to the following four axioms

- $d(i, j) \geq 0$ (non-negative distance)
- $d(i, j) = 0$ (when $i = j$)
- $d(i, j) = d(j, i)$ (commutative)

- $d(i,j) \leq d(i,k) + d(k,j)$ (triangle inequality)

Another distance metric which accounts for the differing variances and covariance amongst the variables in the data set is Mahalanobis distance and it is defined as

$$\text{Mahalanobis} = \sqrt{(\mathbf{x}_i - \mathbf{y}_i)^T \sum^{-1} (\mathbf{x}_i - \mathbf{y}_i)} \tag{3.2}$$

where $\sum$ is the covariance matrix.

Tryon & Bailey (1970) outline the concept of cluster analysis which was first introduced by Tryon in 1939 although an earlier reference to clustering and the concept of measuring likeness was given by Karl Pearson in his 1926 paper "On the Coefficient of Radical Likeness".

Clustering techniques and algorithms are well documented in literature. Bhatia & Deogun (1998) and Carpineto & Romano (1996) describe information retrieval and document classification using conceptual clustering techniques. Judd et al. (1998) use clustering in data mining and image analysis. Hartigan (1984) describes data clustering and unsupervised learning in multidimensional data. An indepth and thorough review of clustering is given in Kain et al. (1999). The term *clustering* is used widely in different research communities to describe methods of grouping unlabelled data. Humans perform competitively with automatic clustering procedures in two dimensions, but most real problems involve clustering in higher dimensions, Kain et al. (1999). Successful clustering is achieved when there is high *within* class similarity (homogeneity) and low *between* class similarity (heterogeneity) between the cluster groups. Clustering also has the ability to discover some hidden patterns in the data and organise large amounts of data both quickly and efficiently. Clustering performance is dependent upon the distance metric and its implementation.

A distance matrix is constructed using the inter-variable distances (or dissimilarities) from an appropriate metric. This matrix, $\mathbf{D}$, is a square symmetric matrix with its principal diagonal elements equal to zero. Two common clustering methods are hierarchical and non-hierarchical algorithms. Hierarchical clustering creates a hierarchical decomposition of the data set using a particular criterion. Non-hierarchical clustering, more commonly known as partitioning methods, construct various partitions and evaluate them with respect to a particular criterion.

3.1.3 Hierarchical Clustering

Hierarchical methods work by grouping the data into a tree of clusters using a computationally efficient technique. When the dimension of the data set is large, it is generally not feasible to examine all possible clustering possibilities. The number of ways of partitioning a set of n items into g clusters is given by

$$N(n,g) = \frac{1}{g!}\sum_{k=1}^{g}\binom{g}{k}(-1)^{g-k}k^n \tag{3.3}$$

in Seber (1984). This can be further approximated by

$$N(n,g) \approxeq \frac{g^n}{g!}$$

which is large even for moderate values of n and g. For example, the number of ways of partitioning 20 items into 10 groups, $N(20,10)$ from Equation 3.3 is $\approxeq 2.76 \times 10^{13}$. Hierarchical methods therefore permit the analyst to search for reasonable solutions without having to look at all the possible clustering arrangements. This grouping can be either *agglomerative* or *divisive*. Agglomerative clustering methods is a bottom-up approach and starts with each object forming a separate cluster. This cluster successively merges the items (or groups of items) until a stopping criterion is reached. Divisive clustering is a top-down approach and starts with all items clustered together. This cluster is iteratively split into smaller clusters until a stopping criterion is reached. Of the two, agglomerative methods are more commonly used.

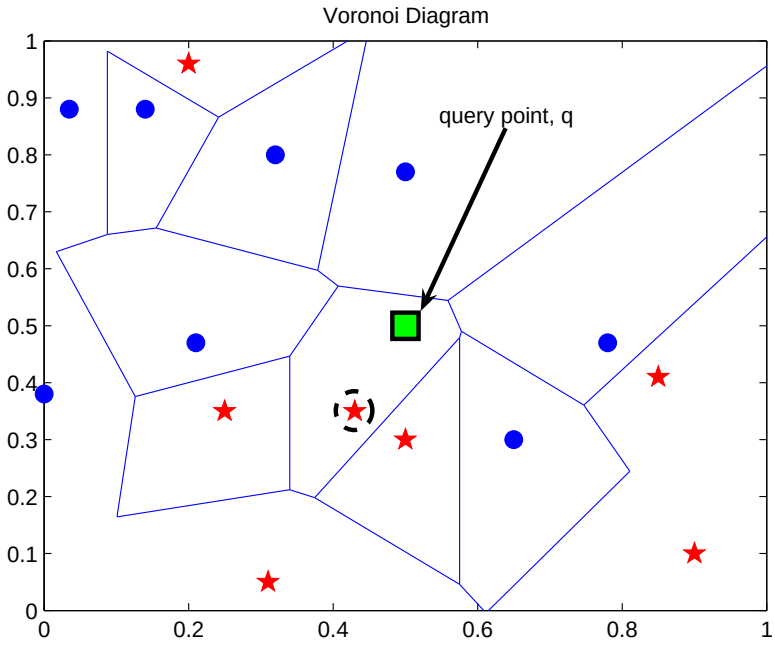
Agglomerative Clustering

One of the most commonly used agglomerative clustering methods is simple linkage, more commonly known as *Nearest Neighbour* clustering. In essence, this is the minimum distance between a point y_i in cluster A and a point y_j in cluster B and is defined as

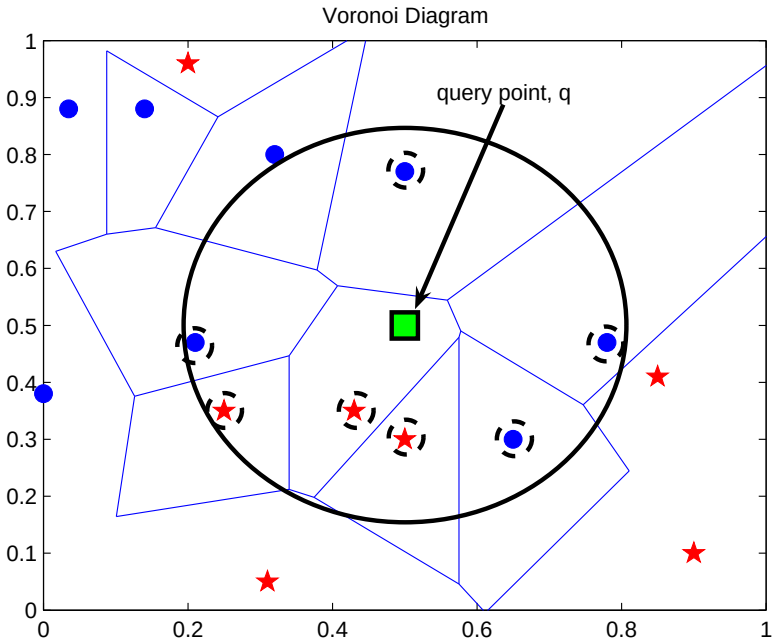
$$D(A,B) = min\{\, d(y_1,y_j), \quad \text{for } y_i \text{ in A and } y_j \text{ in B}\} \tag{3.4}$$

where $d(y_1,y_j)$ is a distance metric such as the Euclidean distance in Equation 3.1 a. At each step in the nearest neighbour method, the distance in Equation 3.4 is calculated for each pair of clusters and the two clusters with the

smallest distance are merged. This is an iterative process and the final result is achieved when the pair with the minimal distance is merged into a single cluster. It is perhaps useful to visualise the concept of nearest neighbour clustering with an example. Figure 3.4 (a) shows a Voronoi diagram on a two class data set $\longrightarrow\{\bullet,\bigstar\}$. This is a simplified classification problem given a new vector (query point q) to classify. A voronoi diagram is constructed by partitioning a plane with n points into convex polygons such that each polygon contains exactly one generating point and every point in each given polygon is closer to its generating point than to any other. Query points outside the convex polygons are closer to some other training example. In Figure 3.4 (a), the 1-NN algorithm classifies q as '$\bigstar$' whereas in Figure 3.4 (b) the 7-NN algorithm classifies q as '$\bullet$' from a majority count in the decision region. This region is depicted by '$\bigcirc$'. Mitchell (1997) states that the hypothesis space H is not implicity considered by the k-NN algorithm, but rather, the algorithm computes the classification of each new query point or instance as needed. The data set used to construct the voronoi diagram can be represented through a connection dendrogram. As clustering is an unsupervised learning technique, the data are without any class labels. This is shown in Figure 3.5, where each sample is treated as a singleton cluster at the onset of the analysis and subsequently pooled with its nearest neighbour to form a new cluster, C_{new}. The colouring of lines in the dendrogram in Figures 3.5 (a) and (b) is to illustrate the clustering effect inherent in the data. Figure 3.5 (a) shows two main cluster groups, indicated by the **blue** and **red** lines. The vertical lines indicate which samples are linked and the horizontal lines indicate the length of a link, *i.e.* the distance between the linked groups. Of interest to note is that data points 11 & 13 have the smallest distance metric (*i.e. they are the closest together*) and points 12 & 14 are somewhat removed from both cluster centres. Figure 3.5 (b) shows the result from the k-means algorithm. It can be seen that there is a slight difference in cluster assignments between the points (shown by the **green** lines). Clustering techniques are widely used for pattern recognition and it is possible to use a cluster dendrogram to assign a class to an unknown sample based on the k-nearest neighbours or $k$means methods but not without the risk of misclassifications.

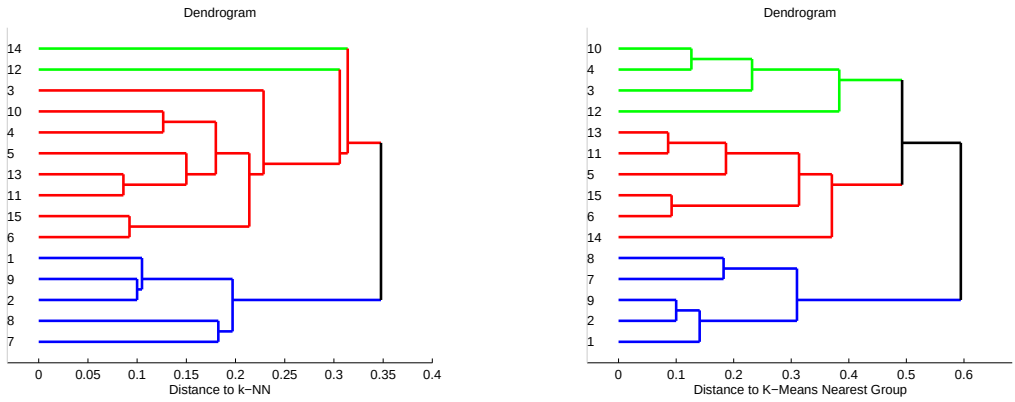


(a)



(b)

Figure 3.4. Voronoi Tesselation in $\mathbb{R}^2$ showing Query Point, q with Nearest Neighbours (NN). (a) *1-NN classifies q as* ★ (b) *7-NN classifies q as* ●, *from a {4,3} count of {●,★} in the decision region* ◯.



(a) (b)

Figure 3.5. Cluster Dendrogram. (a) *A connection dendrogram constructed using the k-NN algorithm* and (b) *The k-means algorithm, using the same data set as Figure 3.4. The '|' lines indicate which samples are linked together, and the '—' lines indicate the length of the link (i.e. the distance between two linked groups).*

3.1.4 Non-Hierarchical Clustering

A common non-hierarchical method is *k-means* clustering. This method allows items to be moved from one cluster to another, a reallocation not applicable in hierarchical clustering. Initially, k items are chosen as *seeds* and data is assigned to the cluster with the nearest seed based on a distance metric (Euclidean). When the cluster has more than one member, the cluster seed is replaced by the centroid. This procedure is sensitive to the initial seed choice, but can be an improvement in clustering performance due to the reallocation of points. Clustering techniques can be problematical for very high dimensional data and time series data as many clustering algorithms are computationally expensive.

3.2 Dimension Reduction Methods

Dimension reduction methods (DRM) are commonplace when dealing with high dimensional multivariate data. Also known as latent variable modelling techniques, they are applicable when a reduced set of variables are required for process monitoring. The rationale behind using DRM is to reduce stochastic noise com-

ponents, monitor fewer process variables, uncorrelate data and assess process performance. Perhaps the most widely used technique is Principal Component Analysis.

3.2.1 Multivariate Data Modelling

Choosing the correct procedure to successfully represent and model multivariate data structures is a difficult task. Martens & Martens (2001) define Bi-Linear Modelling (BLM) as a multivariate method of information extraction which is dependent upon both the data type and modelling algorithm. BLM is built on a combination of multivariate analysis and statistical regression theory, where a methodology is used to extract relevant information from input data and combined with cross-validation and graphical analysis. Models can be used to give a concise and simplified representation of an otherwise complex system (or process) and these allow for quantitative interpretation and prediction. There is a trade off between model complexity and cognisance, therefore a trade-off between the two is important.

The methods used in modelling data are multi-disciplinarian as they can be analogously used in Chemometrics, Econometrics, Psychology, Biology, Statistics, Mathematics and Engineering. The factors that are different between these disciplines are data structures (*i.e.* numerical, continuous, discrete, categorical or mixed mode) and the required *'input-output'* relationships. There are other factors that are very significant to the outcome such as philosophical and technical issues that also need consideration.

3.2.2 Principal Component Analysis

Principal Component Analysis (PCA) is derived from the hypothesis that data variance carries with it information. This is an underlying assumption of PCA. The basic methodology of PCA is used in many different disciplines, each with their own definitions and conventions. In statistics it is known as PCA, in numerical analysis as Singular Value Decomposition (SVD) or Eigenanalysis and in Pattern Classification and Signal Processing as Karhunen-Loéve transform.

PCA is a mathematically efficient technique and its roots and development took place throughout the 20^{th} century. PCA is one of the most heavily used

multivariate techniques which has found wide spread application in a variety of substantive areas, Krzanowski (2002). PCA was first introduced by Pearson (1901) in his paper entitled "On lines and planes of closest fit to systems of points in space", *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science.* Pearson, through a geometrical viewpoint, introduced the concept of *'line of best fit'* and the *'line of worst fit'* through the data, subsequently known as *Principal Components*. Some thirty years later, Hotelling (1933) independently made reference to principal components, and in contrast, took an algebraic approach that established Pearson's principal components were also the orthogonal directions in space that successively maximised the variance of the data. Therefore it holds that the d-dimensional subspace of closest fit to the data is also the subspace in which the variance is maximised. Much has been built on the earlier contributions to PCA and Anderson (1963) developed the distributional theory underlying PCA from a statistical perspective.

PCA is concerned with explaining the variance-covariance structure of a data set through a reduced set of *linear combinations* of the original variables. In short, the objective of PCA is to represent a variable in terms of several underlying factors. Succinctly, PCA can perform:

Data Reduction Although the original data set may contain p variables, it is often the case that much of the variability can be accounted for by a smaller number (m) of principal components.

Data Interpretation Relationships that were previously unsuspected can commonly be identified through PCA.

In Montgomery (2001), PCA is referred to as a Latent Structures Modelling technique because of the analogy with photographic film where a hidden or latent image resides as a result of light interacting with the chemical surface of the film.

PCA is similar in nature Factor Analysis and these two techniques *'look inside'* a set of variables and attempt to assess the structure of the data. The most simple theoretical model for describing a variable in terms of several other variables is a linear one thus, PCA linearly transforms an original set of variables $X = [X_1, X_2, ..., X_p]^T$ into a substantially smaller set of uncorrelated variables Z. The new variable Z represents most of the information contained in the original set

of variables. This implies

$$Z = a_1X_1 + a_2X_2 + ... + a_pX_p \tag{3.5}$$

where $a_1, a_2, ..., a_p$ are loading weights or eigenvectors assigned to the original X variables. Equation 3.5 is similar to the technique used in Multiple Linear Regression (MLR). MLR can be expressed as

$$y = \beta_0 + \beta_1X + \varepsilon \tag{3.6}$$

where y is analogous to the dependent variable or regressand Z, β_1 is the weight a_1 and X are the independent variables or regressors. In Equation 3.5 however, there is no intercept, β_0 and no residual ε.

PCA is used extensively within the field of Chemometrics, where the complement data reduction and latent variable extraction abilities are most useful techniques in dealing with highly dimensional, correlated data matrices. Wise & Gallagher (1996) define Chemometrics as *'the science of relating measurements made on a chemical system to the state of the system via application of mathematical and statistical methods'*. From this definition, it can be inferred that Chemometrics is a data-based methodology and the goal of many techniques is the production of a data derived empirical model that permits an estimate of one or more properties of a system from measurements.

PCA is an overall technique that drawns upon SVD matrix processing routines to encompass data reduction, latent variable extraction, score and loading vectors, PC modelling and regression. The mathematics of PCA are expanded upon in Section 3.2.3.

3.2.3 Singular Value Decomposition

Singular Value Decomposition (SVD) is a matrix processing procedure rather than a direct statistical technique. SVD is an extension to the Eigenvalue Decomposition (ED) technique for non-square matrices, *i.e* it shows any real matrix can be diagonalised using two orthogonal matrices. ED on the other hand, works only on square matrices and uses one matrix (and its inverse) to achieve diagonalisation. If a matrix is square and symmetric, the two orthogonal matrices of SVD become equal, thus SVD and ED become one and the same thing.

Any real $(m \times m)$ symmetric matrix $\mathbf{A}$ can be decomposed into

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda^2}\mathbf{U^T}$$

where $\mathbf{U}$ is orthonormal ($\mathbf{U^T U} = \mathbf{I}$) and $\mathbf{\Lambda^2}$ is diagonal. This gives the normal eigenvalue equation

$$\mathbf{A}\mathbf{u_i} = \mathbf{u_i}{\lambda_i}^2 \tag{3.7}$$

where $\mathbf{u_i}$ is the i^{th} column in $\mathbf{U}$ and ${\lambda_i}^2 = \mathbf{\Lambda}^2_{i,i}$ or principal diagonal value.

It follows that decomposition on any *rectangular* $(m \times n)$ matrix can be given by

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V^T} \tag{3.8}$$

where $\mathbf{X}$ is a $(m \times n)$ matrix, $\mathbf{U}$ is a $(m \times n)$ column-orthonormal matrix (*i.e.* orthogonal and normalised) containing the eigenvectors of the symmetric matrix XX^T (*left singular vectors*), $\mathbf{S}$ is a $(n \times n)$ diagonal symmetric matrix containing the singular values of matrix $\mathbf{X}$ and $\mathbf{V^T}$ is a $(n \times n)$ row-orthonormal matrix containing the eigenvectors of the symmetric matrix X^TX (*right singular vectors*). It is useful to give an explanatory example of SVD and its intrinsic link with PCA. Consider the matrix

$$\mathbf{X} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 5 & 8 \\ 2 & 4 & 7 \end{pmatrix} \tag{3.9}$$

SVD of $\mathbf{X}$ yields

$$\mathbf{U} = \begin{pmatrix} -0.278 & -0.108 & -0.955 \\ -0.736 & -0.615 & 0.284 \\ -0.617 & 0.782 & 0.091 \end{pmatrix} \tag{3.10}$$

$$\mathbf{S} = \begin{pmatrix} 13.444 & 0 & 0 \\ 0 & 0.48 & 0 \\ 0 & 0 & 0.155 \end{pmatrix} \tag{3.11}$$

$$\mathbf{V} = \begin{pmatrix} -0.277 & -0.81 & 0.518 \\ -0.499 & -0.34 & -0.798 \\ -0.821 & 0.479 & 0.31 \end{pmatrix} \tag{3.12}$$

The Trace($\mathbf{S}$) = 14.08. Dividing each diagonal element (*eigenvalue*) of $\mathbf{S}$ by the Trace($\mathbf{S}$) gives its % contribution.

reconstructing the original matrix $\mathbf{X}$, using the two largest singular values (eigenvalues) of $\mathbf{S}$ gives

$$\widehat{\mathbf{X}} = \begin{pmatrix} 1.08 & 1.88 & 3.045 \\ 2.98 & 5.04 & 7.99 \\ 1.99 & 4.01 & 6.995 \end{pmatrix} \tag{3.13}$$

The residual elements between Equation 3.9 and Equation 3.13 are due to fact that the third singular value of $\mathbf{S}$ was not included when reconstructing the matrix. The significance of this result is that two of the three singular values or *'factors'* can reproduce the original data matrix to a reasonable degree. The trace of the symmetric matrix $\mathbf{S}$ (tr ($\mathbf{S}$)) outlines the data variance captured by each singular value. In this case, the cumulative variance of two factors is 98.9%. This is an important concept in PCA.

PCA can identify combinations of variables that describe major trends in the data, and as mentioned previously, PCA relies upon SVD of a data matrix. This methodology is based upon explaining the variance-covariance structure of the original data matrix in terms of a minority of linear combinations of the original variables.

The principal component decomposition of an m-dimensional data set $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_m}]$ can be defined as

$$\mathbf{X} = \mathbf{TP^T} + \mathbf{E} = \sum_{i=1}^{k} \mathbf{t_i p_i^T} + \mathbf{E} \tag{3.14}$$

where k=min(m, n), n is the number of samples, $\mathbf{T} = [\mathbf{t_1}, \mathbf{t_2}, ..., \mathbf{t_k}]$ is the matrix containing the principal component scores, $\mathbf{P} = [\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_k}]$ is the matrix containing the principal component loadings and $\mathbf{E}$ is the residual matrix. The nomenclature in Equation 3.14 is used frequently within the Chemometrics field.

The score vectors, $\mathbf{t_i}$, contain information on how the *samples* relate to each other and the loading vectors, $\mathbf{p_i}$, contain information on how the *variables* relate to each other. The successful implementation of PCA is dependent upon the data structure (*i.e.* raw data, variance-covariance matrix, correlation matrix) and the scaling parameters used. The variance-covariance matrix and the correlation matrix are equal when the original data has been standardised but are not equal with the raw data matrix.

The majority of analysis is performed on variance-covariance and correlation matrices of the original process variables rather than raw data matrices. Given a

data matrix $\mathbf{X}$ with m rows and n columns, the covariance matrix of $\mathbf{X}$ is defined as

$$\mathbf{S} = cov(\mathbf{X}) = \frac{\mathbf{X}^T\mathbf{X}}{m-1} \tag{3.15}$$

where $m-1$ in the denominator of Equation 3.15 is the degree of freedom and it is used to give an unbiased estimate of the covariance matrix, from a sample population. The true covariance matrix, $\mathbf{\Lambda}$ is unknown and therefore an estimate $\mathbf{S}$ is calculated.

As PCA is scale dependent, the raw data must be scaled in a meaningful way. This can be achieved by mean centering, variance scaling, logarithmic scaling or combinations of these. A frequently used method is *'autoscaling'* whereby the columns of the original data matrix $\mathbf{X}$ are adjusted to zero mean and unit variance. This is commonly known as the Z-score normalisation routine, which creates a new data matrix (of the same dimension as $\mathbf{X}$) with zero mean and unit variance. In short, this is often expressed as $\mathbf{N} \sim (0, 1)$. The reasoning behind scaling is to alleviate different magnitudes between variables and remove the effect of numerically large values. This difference has a significant impact on the analysis.

The PCA decomposition in Equation 3.14 can be rewritten in terms of the score vectors $\mathbf{t_i}$

$$\mathbf{T} = \mathbf{XP} \tag{3.16}$$

where $\mathbf{X}$ is the normalised data matrix, $\mathbf{P}$ is the matrix of loading coefficients which provide information as to which variables influence the direction of individual principal components and $\mathbf{T}$ is a matrix of principal component scores which act *ad interim* for the process data. Equation 3.16 holds as the score vector $\mathbf{t_i}$ is a linear combination of the original data $\mathbf{X}$ defined by $\mathbf{p_i}$.

The loadings ($\mathbf{P}$) are the eigenvectors of the variance-covariance matrix and from the normal eigenvalue relation (Equation 3.7), are related to the eigenvalues of the variance-covariance matrix

$$\mathbf{Sp_i} = \lambda_\mathbf{i}\mathbf{p_i} \tag{3.17}$$

where λ_i is the eigenvalue associated with the eigenvector $\mathbf{p_i}$. The eigenvalues of the variance-covariance matrix are a measure of the variance explained by each individual principal component and in this context, variance can be thought of

as *information*. The scores $\mathbf{t_i}$ form an orthogonal set $\mathbf{t_i^T t_j} = 0$ for $i \neq j$, and the loadings $\mathbf{p_i}$ form an orthonormal set $\mathbf{p_i^T p_j} = 0$ for $i \neq j$ and $\mathbf{p_i^T p_j} = 1$ for $i = j$. Maximising Equation 3.17 and rewriting in standard matrix notation yields

$$[\,\mathbf{S} - \lambda\mathbf{I}\,]\mathbf{p_i} = 0 \tag{3.18}$$

where the inclusion of an identity matrix $\mathbf{I}$ is to allow matrix subtraction. Equation 3.18 is solved for non-trivial solutions, *i.e.* $\mathbf{p_i} \neq 0$. Jackson (1991) outlines computational methods used to obtain principal component scores and loading vectors as well as characteristic scaling and PCA implementation.

The score and loading pairs $(\mathbf{t_i}, \mathbf{p_i})$ are arranged in descending order according to the associated eigenvalue (λ_i), therefore the λ_i are a measure of the amount of variance described by the $\mathbf{t_i}, \mathbf{p_i}$ pair. The greatest amount of variance is captured by the first $\mathbf{t_i}, \mathbf{p_i}$ pair, the second greatest amount of variation that is orthogonal is captured by the second $\mathbf{t_i}, \mathbf{p_i}$ pair and so on. Sign inversion is well known in PCA and it comes from the equality

$$\begin{aligned} \mathbf{X} &= \mathbf{u_1} s_1 \mathbf{v_1}^{'} + \mathbf{u_2} s_2 \mathbf{v_2}^{'} + \mathbf{E} \\ &= (-\mathbf{u_1}) s_1 (\mathbf{v_1})^{'} + \mathbf{u_2} s_2 \mathbf{v_2}^{'} + \mathbf{E} \end{aligned} \tag{3.19}$$

and this means that any pair of scores and loadings can be mirrored. It also means that small differences between algorithms that perform PCA may give mirrored results but favourably, both the scores and loadings are mirrored together.

After the transformation, there are as many principal components as original variables. As they are computed in descending order, the lower order components constitute lesser quantities of information and hence can be regarded as process noise (a stochastic component). In practice, it is rarely necessary to compute all $\mathbf{p_i}$ eigenvectors, since the majority of the variance is contained in the first few principal components and generally it is found that the data can be adequately described using far fewer principal components (k) than original variables (m), *i.e.* $k \ll m$.

The proportion of variability in the original data explained by k principal components can be readily calculated by computing the cumulative sum of k eigenvalues and dividing by the total amount of eigenvalues, *i.e.* the sum of the

principal diagonal or trace of (S),

$$\frac{\lambda_k}{tr\,(\mathbf{S})} \tag{3.20}$$

3.2.4 Multivariate Fault Detection

Kourti (2002) addresses process analysis and abnormal situation detection from both a theoretical and applications viewpoint. If there is significant correlation amongst variables, univariate monitoring of process variables does not provide an adequate process monitoring scheme as only a few underlying events are likely to be driving a process at any time Kourti & MacGregor (1995) and these measurements are simply different reflections of the same underlying events Kourti et al. (1996). Therefore, by treating the variables uniquely and independently, significant deviations from normal are difficult to detect as such methods only look at the magnitude of the deviations in each variable independently of all others.

Conventional SPC charting techniques (see Section 2.4) are well established statistical procedures for monitoring stable (and unstable) univariate processes. The difficulty in using independent charts is highlighted in a simplified process schematic, Figure 3.6, where two quality variables X_1 and X_2 are shown for a certain number of samples. Supposing each variable is drawn from a normal distribution and treating each variable independently, it is clear that the stochastic variation is well contained by 3σ Upper and Lower Control Limits (U / L CL). This is not the case in a bivariate or multivariate sense (whereby the control region is shown by an ellipsoidal confidence region). Since process (and product) quality are multivariate properties, many constituent variables may be correlated and simultaneous monitoring of these can show any significant departure from the multivariate normal control region. The univariate probability that either X_1 of X_2 exceed the 3σ control limits is (1-0.9973) or 0.0027. The joint probability that both variables exceed their control limits simultaneously when they are both in control is $(0.0027)^2$ which is considerably less than 0.0027. Conversely, the probability that both X_1 and X_2 will simultaneously plot inside the control limits when the process is in control is $(0.9973)^2$ or 0.9964. In summary, an indication of an out of control process can be seen by the ● point outside of the main ellipsoidal region but it is not possible to detect an assignable cause by maintaining independent control charts.

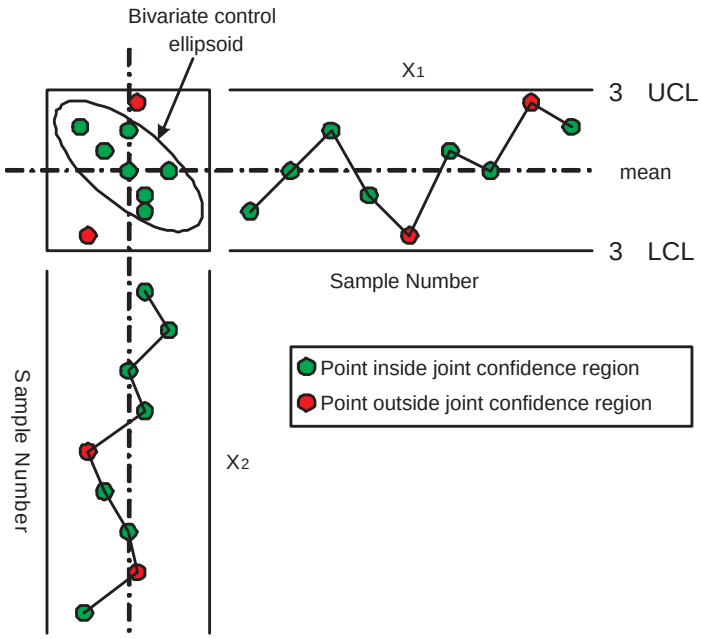


Figure 3.6. Bivariate Control Region. *Quality control of two variables illustrating both the ellipsoidal joint confidence region and the misleading nature of univariate control charting.*

If the X_1 and X_2 variables are independent the correlation between them is zero, *i.e.* $\sigma_{12} = 0$, and the principal axes of the ellipsoidal control region are parallel to the original X_1 and X_2 axes. If however, the X_1 and X_2 variables are dependent, *i.e.* $\sigma_{12} \neq 0$, then the principal axes of the ellipsoidal region are no longer parallel to the X_1 and X_2 axes or equal, thus the ellipse undergoes a rotation. This is shown in Figure 3.6. There are some difficulties with the control ellipse however, as it removes the temporal sequence of the sample run and it is significantly more inefficient to calculate for more than two variables.

Fault detection in multivariate subspaces can be achieved in many different ways, but perhaps the two most commonly used and well known methods are the Hotelling's $\mathbf{T}^2$ statistic and the squared prediction error (SPE) or $\mathbf{Q}$ statistic, He et al. (2004). The $\mathbf{T}^2$ statistic is a measure of variation *within* a PCA model and it is calculated as the sum of the normalised squared scores. The $\mathbf{Q}$ statistic on the other hand indicates how well each sample conforms to the PCA model and is calculated by the projection of a sample vector on a residual space, Chen et al. (2004).

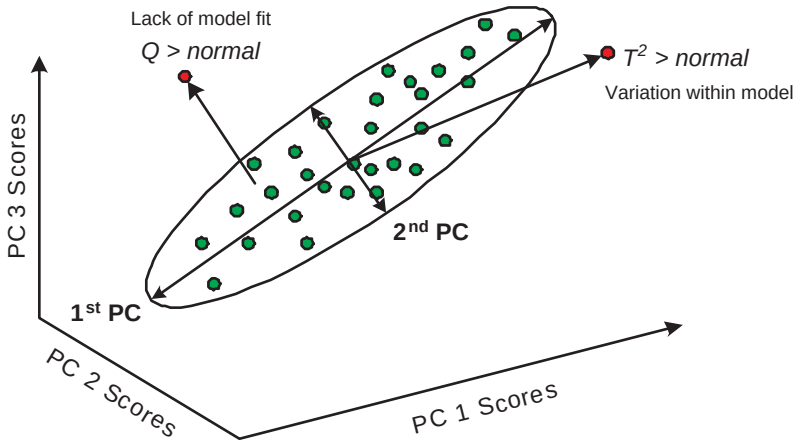


Figure 3.7. Principal Component Model. *PCA representation of a 3-dimensional data set showing control ellipsoid, $\boldsymbol{T}^2$ sample outlier and $\boldsymbol{Q}$ sample outlier. The points shown by ● are abnormal for PCA model. The data in this case are adequately described by a 2 PC model. Adapted from Wise et al. (1999).*

Wise et al. (1999) outline the concept of PCA monitoring with multivariate indices on a three-dimensional data set, lying primarily in a single plane. Figure 3.7 depicts a three dimensional principal component subspace (with arbitrary axes) with a 3-σ control ellipsoid. The major axis of the ellipsoid is the 1^{st} PC and the minor axis is the 2^{nd} PC. The data are shown to be adequately described by a two component model. Included in the Figure are two outliers, one a $\mathbf{T}^2$ outlier which is a measure of the Mahalanobis distance in the principal component subspace between the position of a sample and the average behaviour of the process (*i.e.* the locus on the ellipsoidal confidence region in the multidimensional subspace, Jackson (1991) and Ku et al. (1995)), the other a $\mathbf{Q}$ outlier where the sample is not adequately described by the model. A nominal threshold limit for both indices signals whether the process is operating normally or if a disturbance has occurred.

The diagnosis of abnormal process behaviour can be greatly enhanced if similar process conditions and plant performance can be located in historical databases, thus essentially reducing this to a pattern recognition problem. Although not all of the faults will have occurred in historical records, the most common ones may reside in a similar process subspace, which in turn may assist fault

identification. Diagnosis of disturbances for low-dimensional processes can be achieved visually with cylindrical or ellipsoidal control regions, but the sensitivity decreases as the number of dimensions increase.

Fault identification can be inferred from a PCA model, with both the $\mathbf{T}^2$ and $\mathbf{Q}$ statistics producing an *'out-of-control'* signal when a fault occurs. In a process, it is possible to distinguish between two classes of change. The first class is a change in process operation which may result in greater variation in some process variables. The relationship between the variables remains the same however, the result being a shift in the mean value of one or more process variable. The metric used to detect this change is Hotelling's $\mathbf{T}^2$. The second class is associated with a change in the correlation structure of the process variables and the metric used to signal this change is the $\mathbf{Q}$ statistic, Martin et al. (2002). Neither of these, however, provide any information about the cause of the fault, *i.e.* they are non-causal, Ündey & Çinar (2002). Contribution plots indicate which variable(s) are responsible for the deviation from normal process behaviour and are used to successfully diagnose a fault condition. Raich & Çinar (1997) introduce statistical distance and angle measures in fault diagnosis, where the trimmed PCA model, *i.e.* model with redundant dimensions removed, is tested for outlier sensitivity using both indices.

Hotelling's $\mathbf{T}^2$ Statistic

A traditional multivariate SPC approach to process monitoring is achieved through the use of the $\mathbf{T}^2$ statistic. Given a vector of measurements or quality characteristics z on n normally distributed variables, with an in-control variance-covariance matrix, the current mean of the multivariate process can be compared to the population mean μ by computing the chi-squared statistic (χ^2) through

$$\chi^2 = n(\mathbf{z}-\mu)'\sum\nolimits^{-\mathbf{1}}(\mathbf{z}-\mu) \tag{3.21}$$

where $\mu = (\mu_1, \mu_2, ..., \mu_p)'$ is a $(p\times 1)$ vector of in-control means, $\sum$ is the variance-covariance matrix. The χ^2 statistic can be plotted against time with an upper control limit (UCL) given by $\chi^2_{\alpha,n}$, where α is an appropriate significance level (e.g., α=0.01 or 0.05). The χ^2 statistic in Equation 3.21 represents the Mahalanobis distance of any point from μ, Montgomery (2001) and Jackson (1991).

In practice it is usually necessary to estimate the mean and variance-covariance from a sample n of an entire population P. This sample is assumed to be extracted from the process operating under steady state conditions and follow a multivariate normal distribution. The mean and variance of a population sample are calculated in the normal manner

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and

$$\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x_i} - \overline{\mathbf{x}})(\mathbf{x_i} - \overline{\mathbf{x}})'$$

$\mathbf{S}$ is often expressed in variance-covariance matrix form (with $n = 3$) Maesschalck et al. (2000)

$$\mathbf{S} = \begin{pmatrix} \mathbf{S_{11}} & \mathbf{S_{12}} & \mathbf{S_{13}} \\ \mathbf{S_{21}} & \mathbf{S_{22}} & \mathbf{S_{23}} \\ \mathbf{S_{31}} & \mathbf{S_{32}} & \mathbf{S_{33}} \end{pmatrix}$$

then the average of the sample variance-covariance matrix $\mathbf{S}$ is an unbiased estimate of $\sum$ when the process is in-control.

Replacing μ with $\overline{\mathbf{x}}$ and $\sum$ with $\mathbf{S}$ in Equation 3.21 yields Hotelling's test statistic

$$\mathbf{T^2} = n(\mathbf{z} - \mu)'\mathbf{S^{-1}}(\mathbf{z} - \mu)$$

commonly the subgroup size n is equal to one, thus

$$\mathbf{T^2} = (\mathbf{z} - \mu)'\mathbf{S^{-1}}(\mathbf{z} - \mu) \tag{3.22}$$

$\mathbf{T^2}$ is directly related to the F distribution, and depends upon the degrees of freedom in $\mathbf{S}$, Jackson (1991) and Jolliffe (1986). An upper limit, $\mathbf{T^2}_{UCL}$ can be calculated from

$$\mathbf{T^2}_{UCL} = \frac{(m^2-1)n}{m(m-n)}F_{\alpha,n,m-n} \tag{3.23}$$

where m is the number of samples used to develop the PCA model and n is the number of principal components (pc's) retained, $F_{\alpha,n,m-n}$ is the upper $100(1-\alpha)\,\%$ critical point of the F distribution with $(n, m-n)$ degrees of freedom. As this statistic is squared, the lower limit is equal to zero, *i.e.* $\mathbf{T^2}_{LCL} = 0$.

$\mathbf{T^2}$ in Equation 3.22 can also be expressed in terms of the first i^{th} pc's of the PCA model, Kourti & MacGregor (1996) and Jackson (1991), thus

$$\mathbf{T^2} = \sum_{i=1}^{n} \frac{t_i^2}{\lambda_i} \; = \; \sum_{i=1}^{n} \frac{t_i^2}{s_i^2} \tag{3.24}$$

where λ_i are the eigenvalues of $\mathbf{S}$ and t_i are the scores from the pc transform. s_i is the estimated variance of the corresponding latent variable t_i as the variances of the pc's are the eigenvalues of $\mathbf{S}$.

Q Statistic

The $\mathbf{Q}$ statistic is a scalar measurement of the *'goodness-of-fit'* of a sample $\mathbf{x}$ to a PCA model. Once a model has been developed from nominal data (a NOC model) using a reduced set of principal components, the squared prediction error (SPE) can be calculated. The SPE provides the facility to identify the onset of a new event not previously captured within the data. This is significant for identifying incipient drift, small dynamic differences and ill-fitting models.

A special event will generate new pc's and this has the effect of moving the new observation, x_{new} off the plane described by the k pc's. Such special events can be detected by computing the SPE of the residuals of x_{new}

$$\mathbf{Q} \; = \; SPE_x \; = \; \sum_{i=1}^{n} (x_{new,i} - \widehat{x}_{new,i})^2 \tag{3.25}$$

where the predicted values are given by $\widehat{\mathbf{X}} = \mathbf{TP^T}$, MacGregor et al. (2005). In matrix notation this becomes

$$\mathbf{Q} = \mathbf{x_i^T}(\mathbf{I} - \mathbf{P_k}\mathbf{P_k^T})\mathbf{x_i} \tag{3.26}$$

where $\mathbf{I}$ is an identity matrix, $\mathbf{P_k}$ is a matrix of k loading vectors retained in the PCA model and $\mathbf{x_i}$ is the i^{th} sample in $\mathbf{X}$.

A confidence limit for $\mathbf{Q}$ can be established from the standard normal deviate which corresponds to the $100(1-\alpha)$ percentile

$$c = \theta_1 \frac{\left[(\frac{Q}{\theta_1})^h - \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} - 1 \right]}{\sqrt{2\theta_2 h_0^2}} \tag{3.27}$$

where c is $\mathbf{N} \sim (\mathbf{0}, \mathbf{1})$, Jackson & Mudholkar (1979). Consequently, the critical value $\mathbf{Q}_\alpha$ is

$$\mathbf{Q}_\alpha = \theta_1 \left[\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} + 1 \right]^{\frac{1}{h_0}} \tag{3.28}$$

where

$$\theta_i = \sum_{j=k+1}^{n} \lambda_j^i \qquad \text{for } i = (1, 2, 3) \tag{3.29}$$

and

$$h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2} \tag{3.30}$$

where k is the number of pc's retained in the model and n is the total number of pc's.

Subsequent analysis is required in order to ascertain the variables that account for process change and this is achieved through contribution plots and this concept is further developed in Section 5.2.6.

3.3 Chapter summary

The application of unsupervised learning is reviewed as a preconditioning methodology for input to a control strategy. This chapter describes unsupervised learning methods. Clustering and parallel coordinate analysis are introduced for exploratory data analysis. Multivariate data modelling in the form of Principal Component Analysis (PCA) is introduced as a dimension reduction method. The mathematics of singular value decomposition are outlined with a small example. The concept of multivariate fault detection is introduced along with some common fault indices.

Chapter 4

Supervised Learning

Contents

"Statistics in the hands of an engineer are like a lampost to a drunk- they're used more for support than illumination."
(B. Sangster)

4.1 Supervised Learning Methods

Supervised learning requires one to be confident of the true classes of the original data used to build the model. Classification and prediction algorithms are in general computational means for reducing the amount of information in data.

The input is likely to be information rich sequence data but the output may be a class or predicted number, or in the simplest case a dichotomy representing two classes.

The definition of machine learning differs according to each author, but in general, a widely accepted definition is *'A computer program is said to* ***learn*** *from experience E with respect to some class of tasks T and performance measure P, it its performance at task T, as measured by performance P, improves with experience E',* Mitchell (1997). Witten & Frank (2003) outline a more general description of *'things learn when they change their behaviour in a way that makes them perform better in the future'.*

4.1.1 Machine Learning Terminology

Machine learning is a mutli-disciplinary field of research devoted to the formal study of learning systems. As this is a synergistic combination of many sub-communities, machine learning has roots in statistics, computer science, artificial intelligence, engineering, control theory, optimisation theory, philosophy and many other disciplines of science and mathematics. Therefore, the terminology associated with each communities is analogous to one another. For clarify it is useful to define the terminology most commonly associated with machine learning.

Instance A single example in a data set. A row vector in a data matrix.

Attribute An attribute is a characteristic (*or property*) of an instance and can take on either categorical or numeric values. Alternatively known as *vector, pattern, case, sample, dimension and observation.* These can be used interchangeably without a loss of generality.

Value Either categorial or numerical.

Concept The 'thing' to be learned.

4.2 One-Rule Algorithm

Simplicity is an important concept in statistical model building. Many authors recommend a 'simplicity-first' methodology when analysing practical data sets

Witten & Frank (2003) and Fielding (1999). Surprisingly, many of the 'simple' algorithms will perform well and some interesting information can be gleaned from the process. Often, latent structures exist within the data and a rudimentary algorithm is sufficient for good classifier performance. The rules induced by machine learning systems (*e.g. classification, regression, etc.*) are judged on two criteria:

1. Classification accuracy on independent test sets (*accuracy*),
2. Complexity component

The one-rule (1R) algorithm generates a set of rules that all test on one particular attribute. This is equivalent to a one-level decision tree. The aim is to infer a rule that predicts a class given certain attribute values. The basic version, assuming nominal attributes, has one branch for each of the attribute's values and each branch assigns the most frequent class, Holte (1993). The error rate associated with 1R is the proportion of instances that do not belong to the majority class of their corresponding branch. The attribute with the lowest error rate is chosen. The pseudo code for the implementation of 1R is shown. This is taken from Nevill-Manning et al. (1995).

```
Pseudo code for 1R
-------------------------------------------------------------
For each attribute a, form a rule as follows:
    For each value, v, from the domain of a,
        count class frequency
        find the most frequent class
        make the following rule:
            if a has a value, v, then class is c
    Calculate the accuracy/error rate of the rule
Choose rules with the greatest accuracy/smallest error rate.
=============================================================
```

1R ranks attributes according to their accuracy/error rates as opposed to entropy based measures such as those used in decision tree induction. The algorithm treats all valued attributes as continuous and uses a straight forward method to divide the range of values into several disjoint intervals. It is common practice in

Outlook	Temp	Humidity	Windy	Class
sunny	hot	high	false	Don't Play
sunny	hot	high	true	Don't Play
overcast	hot	high	false	Play
rain	mild	high	false	Play
rain	cool	normal	false	Play
rain	cool	normal	true	Don't Play
overcast	cool	normal	true	Play
sunny	mild	high	false	Don't Play
sunny	cool	normal	false	Play
rain	mild	normal	false	Play
sunny	mild	normal	true	Play
overcast	mild	high	true	Play
overcast	hot	normal	false	Play
rain	mild	high	true	Don't Play

Table 4.1. Weather data from Quinlan (1993). *Both temperature and humidity are nominal representations of the Fahrenheit scale (^oF) and % humidity. Both these attributes can be expressed on a continuous scale.*

Rule	Attribute	Rules	Errors	Total Errors
1	outlook	sunny → `don't play`	$\frac{2}{5}$	$\frac{4}{14}$
		overcast → `play`	$\frac{4}{4}$	
		rain → `play`	$\frac{2}{5}$	
2	temp	hot → `don't play`*	$\frac{2}{4}$	$\frac{5}{14}$
		mild → `play`	$\frac{2}{6}$	
		cool → `play`	$\frac{1}{4}$	
3	humidity	high → `don't play`	$\frac{3}{7}$	$\frac{4}{14}$
		normal → `play`	$\frac{1}{7}$	
4	windy	false → `play`	$\frac{2}{8}$	$\frac{5}{14}$
		true → `don't play`*	$\frac{3}{6}$	

Table 4.2. 1R evaluation of attributes. *The incidence of* ***`don't play`***** denotes a random choice made between 2 equally likely outcomes. The rule with the lowest error rate is chose as the classifier.*

data mining and machine learning to make use of an established data set for test purposes and benchmarking. Table 4.1, Quinlan (1993), is an illustrative data set in which there are four attributes (two nominal and two continuous) and two classes. A decision is made on whether to play golf or not based on four different input attributes.

To classify on the outcome (*play* or *don't play*), 1R considers four sets of rules, one for each of the attributes. There are two outcomes, the number of errors per rule and the total number of errors for the rule set. These rules are shown in Table 4.2. 1R chooses the attribute that produces rules with the smallest number of errors, and for Table 4.2, these are the first and the third rule sets. The first rule is determined through the attribute 'outlook', and forecasts whether to play or not only on this attribute. Inference from this yields the following precedent:

```
outlook:  sunny    --> don't play
          overcast --> play
          rain     --> play
```

This data set is a hypothetical one often used to demonstrate machine learning algorithms. Using outlook as the key attribute, the 1R algorithm success-

fully classifies six out of a total of fourteen instances, thus giving an accuracy of 42.86%. Application of 1R to the weather data is for demonstration only and certain improvements in algorithm performance can be achieved through attribute discretisation, Witten & Frank (2003).

4.3 Decision Tree Induction

Decision tree induction is a popular method of classification (*and prediction*) in supervised learning. It is a hierarchical model whereby the local region (*or problem space*) is identified in a sequence of recursive splits. The theoretical framework can be traced back to the seminal work of Hoveland and Hunt in the late 1950's. Hunt et al. (1966) describe the implementation of Concept Learning Systems (CLS).

The concept of decision trees as a learning paradigm was invented separately in two different fields of research. Classification And Regression Trees (CART) were developed in 1984 by researchers in the statistical sciences, Breiman et al. (1984), and the ID3 tree was invented in the machine learning community, Quinlan (1986). Quinlan refined this algorithm and released C4.5, Quinlan (1993).

In Hunt's CLS algorithm, Hunt et al. (1966), a decision tree is grown in a recursive manner by partitioning the training cases into successively purer subsets. This method for constructing a decision tree is quite elegant and simple. If D_t is the set of training cases that are associated with a node t and $y = \{y_1, y_2, ..., y_c\}$ are the class labels, a recursive definition of Hunt's algorithm can be outlined:

Step 1 If all the records in D_t belong to the same class y_t, then t is a leaf node labeled as y_t.

Step 2 If D_t contains instances that belong to more than one class, an **attribute test condition** is selected to partition the instances into smaller subsets. A child node is created for each outcome of the test condition and the instances in D_t are distributed to the children based on the outcomes. (C4.5 uses the most frequent class as parent of this node).

Step 3 This algorithm is then recursively applied to each child node.

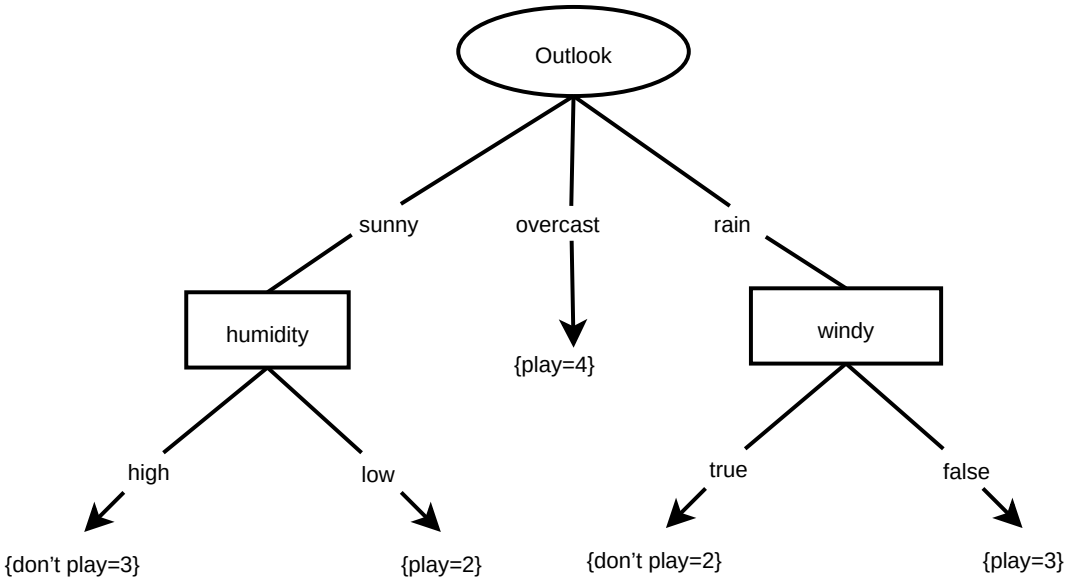


Figure 4.1. Decision tree representation of weather data, Quinlan (1993)

Decision trees are the single most commonly used paradigm in data mining, Mitchell (1997). One advantage that decision tree modelling has over other classification techniques lies in the interpretability of the constructed model, Myles et al. (2004). Some other advantages are ease of implementation and use, cost to construct (*inexpensive*), speed of classification and computational load.

Perhaps the best method of illustrating a decision tree is to give an example. Figure 4.1 shows a decision tree generated from the weather data, Quinlan (1986). The temperature and humidity attributes are described in nominal form but extensibility to continuous attributes is possible in C4.5.

```
Weather decision tree rules.
-------------------------------
outlook = sunny
|   humidity = low:--> yes (2.0)
|   humidity = high:--> no (3.0)
outlook = overcast:--> yes (4.0)
outlook = rain
|   windy = TRUE:--> no (2.0)
|   windy = FALSE:--> yes (3.0)
-------------------------------
```

```
Number of Leaves: 5
Size of the tree: 8
```

In Figure 4.1, the **root node** is `outlook` which has no incoming graph edges and three outgoing edges. `Humidity` and `windy` are **internal nodes**, each of which have one incoming edge and two or more outgoing edges. **Leaf** or **terminal** nodes have one incoming edge and no outgoing edges and these are shown by the `yes (play)` and `no (don't play)` decisions. Each leaf node is assigned a class label and the **non-terminal** nodes contain attribute test conditions to separate instances that have different characteristics. For example, in Figure 4.1, the attribute test condition for the root node is to separate `outlook` into three subcategories `sunny, overcast` and `rain`. In summary, any decision tree is composed of the following nodes:

- A **root node** that has no incoming graph edges and zero or more outgoing edges.
- **Internal nodes**, each of which has exactly one incoming edge and two or more outgoing edges.
- **Leaf** or **terminal node**, each of which has exactly one incoming edge and no outgoing edges.

A decision tree is constructed by recursively partitioning the feature space of the training set. The objective is to find a set of decision rules that naturally partition the feature space to provide an informative and hierarchical classification model. Decision tree partition rules can differ but certain scoring criteria have been established to evaluate the partition rules as the induction algorithm must address how to best split the training instances, Murthy (1998). Each recursive step of the decision tree growing process selects an attribute test condition to divide the instances into smaller subsets and provide a measure for evaluating the goodness of each test condition, Loh & Shin (1997)

4.3.1 Attribute Types

Decision tree induction algorithms have to be extensible to different attribute types. A **Binary** attribute test condition generates two potential outcomes. A

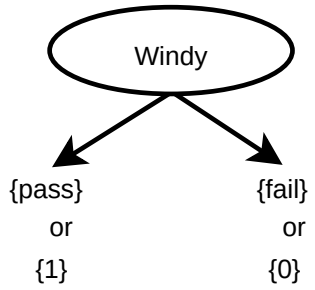


Figure 4.2. Binary attribute test condition

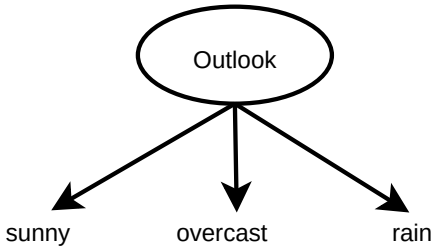


Figure 4.3. Nominal attribute *multiway split* test condition

binary attribute is a special case of a discrete attribute, and is expressed in dichotomous terms, *i.e. true/false*, *pass/fail* or *0,1*. This schema is shown in Figure 4.2. A nominal attribute may have many values, and therefore, subsequent test conditions are expressed in two ways. Figure 4.3 shows a multiway split from the attribute `outlook`. Alternatively, Figure 4.4 shows three different ways of grouping the nominal attribute `outlook` into binary subsets. Ordinal attributes can also be handled in this manner. Continuous attributes contain a

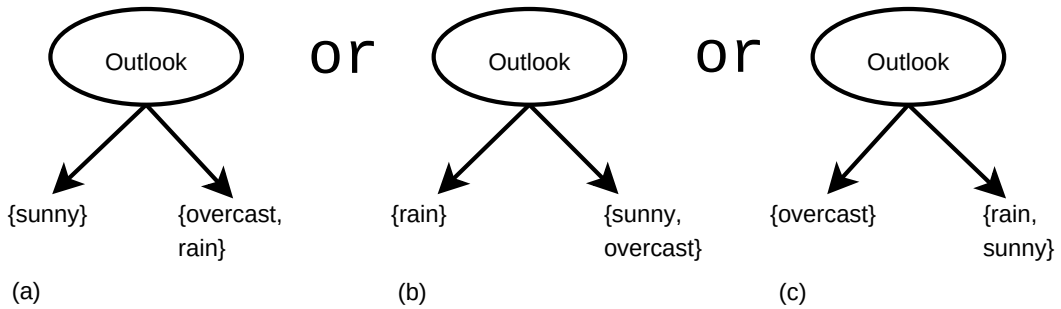


Figure 4.4. Nominal attribute *binary split* test condition. *Attributes are shown grouped together.*

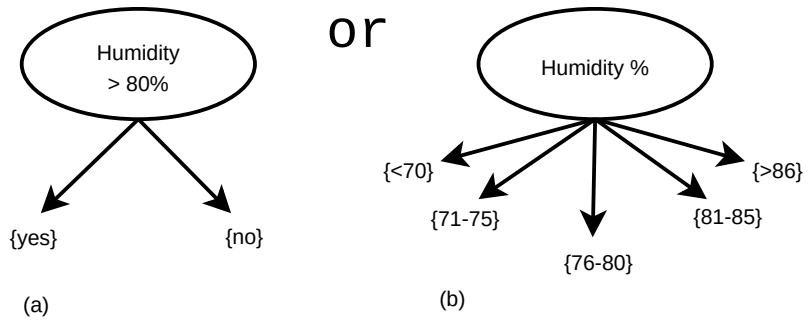


Figure 4.5. Test conditions for a continuous attribute. *These can be expressed as either a comparison test* (a) *or a range query* (b).

test condition that is either a comparison or range query. The difference between these two is shown in Figure 4.5.

4.3.2 Selecting The Root Node Attribute

The key idea in decision tree induction is to determine which attributes best classify the data. The central choice is to select certain attributes at each node of the tree. The root node of the decision tree contains all the data. There are many measures that can be used to determine the best split and they are often based upon measuring the degree of purity (*or impurity*) at each node in the tree Frank et al. (1997). This continues until no further split is possible and thus a terminal node is reached. At each node, the data are split according to the values of one particular feature and the splits are chosen to maximise the *gain* in information. This was originally proposed by Shannon in 1948 in his seminal paper 'A Mathematical Theory Of Communication', Shannon (1948). The most basic of decision tree learning algorithms, ID3 Quinlan (1986), is a top-down approach that searches through the given sets to test each attribute at every tree node. This approach of ID3 implementation is summarised in Table 4.3.

In order to introduce the statistical property, *information gain*, is used to define a common measure in information theory, *entropy*. Entropy, H, is a measure of the (*im*)purity of an arbitrary collection of instances. It is the reciprocal of homogeneity and is measured in binary digits or *bits*. Letting S represent the set of all instances which contain positive ($p_{\oplus}$) and negative ($p_{\ominus}$) examples of a

Loop	Algorithm description
1.	$A \leftarrow$ the 'best' decision attribute for next *node*
2.	Assign A as decision attribute for *node*
3.	For each value of A, create new descendent of *node*
4.	Sort training examples to leaf nodes
5.	IF training examples perfectly classified THEN stop
6.	ELSE iterate over new leaf nodes

Table 4.3. Top-Down Induction of Decision Trees (TDIDT). *This algorithm determines the most useful attribute for classifying instances. This process continues until the tree perfectly classifies the instances or until all the attributes have been used.*

target class, the entropy of S relative to the boolean classification is

$$Entropy\,(S) \equiv H(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus} \tag{4.1}$$

where $p_{\oplus}$ is the proportion of positive examples in S and $p_{\ominus}$ is the proportion of negative examples in S. $H(S) = 0\,bits$ if the sample is pure (*i.e.* either all $p_{\oplus}$ or $p_{\ominus}$) and $H(S) = 1\,bit$ if equally distributed (*i.e.* $p_{\oplus} = p_{\ominus}$). Naively computing the function $f(p) = -p \log_2 p$ with $p = 0$ is problematical as dividing a finite number by zero leads to an infinite number. Therefore in all calculations, entropy is defined to be 0 when $p = 0$, *i.e.* $0 \log_2 0 = 0$.

In Figure 4.6, H is monotonically increasing in the domain $\{0 < p_{\oplus} < 0.5\}$, and decreasing in the domain $\{0.5 < p_{\oplus} < 1\}$. When all the members of S belong to the same class, $H = 0$. Likewise, with an equal number of classes, $(p_{\oplus} = p_{\ominus})$, $H = 1$. With an unequal numbers of classes $(p_{\oplus} \neq p_{\ominus})$, H is between $0 \rightarrow 1$. This is summarised in Table 4.4.

To illustrate this the weather data, Quinlan (1986), is used once more. Let S be the boolean concept of {*play or don't play*}. There are nine positive examples and five negative examples. Mitchell (1997) adopts the following notation [9+,5-], to indicate boolean class memberships. The entropy of S relative to this classification can be determined from Equation 4.1,

$$\begin{aligned} H(S:\,[9+,5-]) &= -\left(\frac{9}{14}\right) \log_2 \left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) \log_2 \left(\frac{5}{14}\right) \\ &= 0.9403\,bits \end{aligned}$$

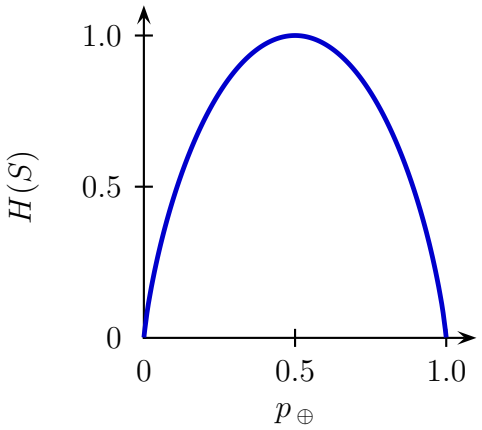


Figure 4.6. Entropy function of a boolean classification. *This shows the entropy function, H(S), relative to a boolean classification, as $p_\oplus$ varies between 0 and 1.*

Class membership	Entropy (H)
$\frac{p_\oplus}{p_\oplus + p_\ominus} = 0$	0
$\frac{p_\oplus}{p_\oplus + p_\ominus} = 0.5$	1
$\frac{p_\oplus}{p_\oplus + p_\ominus} = 1$	0

Table 4.4. Class membership and entropy from Figure 4.6. *H is monotonically increasing with class membership in the range $0 \rightarrow 0.5$ and decreasing in the range $0.5 \rightarrow 1$.*

The information theoretic interpretation of H is the number of bits required to encode the classification of an arbitrary member of S, thus it represents the average information needed for a class distinction. A more general form of entropy when the target attribute can take on c different values is given as

$$H\,(S) \equiv \sum_{i=1}^{c} -p_{\,i} \log_2 p_{\,i} \tag{4.2}$$

Information gain is the expected reduction in entropy due to sorting on a particular attribute. The information gain, $Gain(S, A)$, of an attribute A, relative to a collection of instances S is defined as

$$Gain(S, A) \equiv H\,(S) - \sum_{v \in Values\,(A)} \frac{\mid S_v \mid}{\mid S \mid} H\,(S) \tag{4.3}$$

where $Values(A)$ is the set of all possible values for attribute A, S_v is the subset of S for which attribute A has value v, *i.e.* $S_v = \{s \in S \mid A(s) = v\}$, Quinlan (1993), Mitchell (1997), Mantaras (1991) and Witten & Frank (2003).

4.3.3 Computing Attribute Information Gain

Information gain is used by the ID3 algorithm to select the best attribute at each stage in growing the decision tree. There are four attributes in the data set so therefore, four information gain calculations are carried out and the one which gains the most information is split on. The class membership for the attribute `outlook` is shown in Figure 4.7. Information gain on the attribute `outlook`, $Gain(S, outlook)$, is calculated from Equation 4.3.

$$\begin{aligned}
Gain(S,\, \texttt{outlook}) &= H(S) - \sum \frac{\mid S_v \mid}{\mid S \mid} H(S_v) \\
Gain(S,\, \texttt{outlook}) &= H(S) - \left[\frac{\mid S_{sunny} \mid}{\mid S \mid} H(S_{sunny}) + .. + \frac{\mid S_{rain} \mid}{\mid S \mid} H(S_{rain})\right] \\
Gain(S,\, \texttt{outlook}) &= 0.9403 - \left[\left(\frac{5}{14}\right) 0.971 - \left(\frac{4}{14}\right) 0 - \left(\frac{5}{14}\right) 0.971\right] \\
Gain(S,\, \texttt{outlook}) &= 0.2464\, bits \qquad (4.4) \\
& \qquad (4.5)
\end{aligned}$$

Calculation of information gain for the other attributes is achieved analogously. These are succinctly shown in Table 4.5. The result shows that `outlook` provides

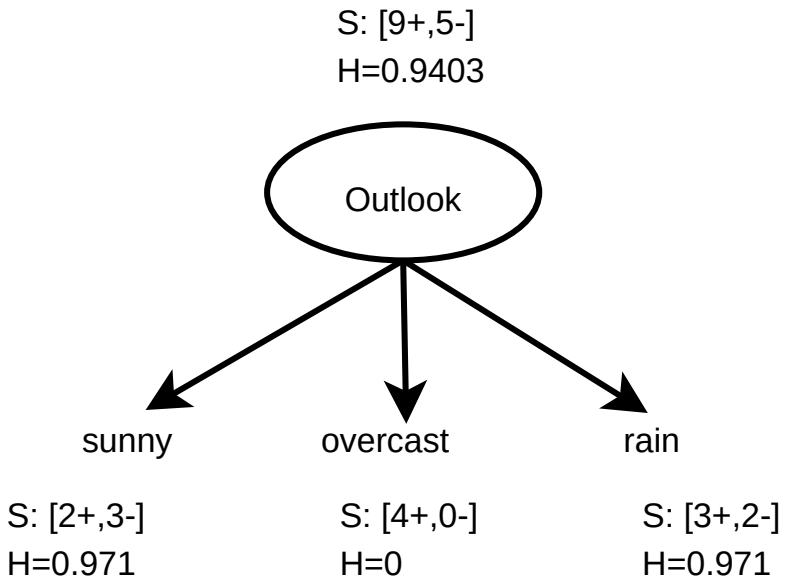


Figure 4.7. Information gain for the `outlook` attribute

the best prediction of the target class as it gains the most information and thus is chosen as the root node. `Outlook` is a nominal attribute and the branches emanating from the root node are represented by multiway split to each of the possible values, (*i.e.* `sunny, overcast` and `rain`). Figure 4.1, the original decision tree and Figure 4.7 both show this concept. Intuition suggests that the root attribute should be `outlook` as this would result in one of the internal nodes (*child nodes*) being completely pure.

The process of selecting a new attribute is recursive repeated and attributes

`Attribute`	Gain (bits)
$Gain(S, \texttt{outlook})$	**0.2464**
$Gain(S, \texttt{humidity})$	0.1519
$Gain(S, \texttt{wind})$	0.0481
$Gain(S, \texttt{temperature})$	0.0292

Table 4.5. Information gain for all 4 attributes. *According to the result,* `outlook` *provides the best prediction of the target class,* {`play, don't play`}.

that have been incorporated higher in the decision tree are excluded. Termination occurs when the data can not be split any further. This is most easily visualised in Figure 4.1.

4.3.4 Issues in Decision Tree Induction

There is a certain inductive bias in ID3 that favours the shorter decision trees. This is derived from the fact that there are many more complex hypotheses that fit the training data, but fail to generalise correctly and accurately with subsequent data. Simpler trees do not partition the feature space into too many small boxes. This was first proposed circa 1320 by William of Occam, a medieval philosopher, apparently whilst shaving. This bias therefore is known as Occam's razor and it states 'to prefer the simplest hypothesis that fits the data' and simpler explanations are more plausible and unnecessary complexity should be 'shaved off', Alpaydin (2004).

If the algorithm perfectly classifies the training data, then there is a risk that the decision tree *overfits* the data and thus will perform less accurately on unseen, new instances. Larger trees tend to have a better training data accuracy than test data accuracy. Termination of the tree before perfect classification and post-pruning are two methods commonly used to avoid overfitting. This minimises the risk of over-training the classifier and incorporating too much noise.

The original algorithm, ID3, originated for discrete attributes only. Extensibility to continuous attributes was made possible with C4.5, Quinlan (1986) and the java version J4.8, Witten & Frank (2003).

There are other methods of measuring information gain. In Classification and Regression Trees, CART, Breiman et al. (1984), the impurity is defined as the Gini index. This statistic is defined as

$$Gini = 1 - \sum_{j=1}^{c} \left(\frac{n_j}{n}\right)^2 \tag{4.6}$$

where n_j is the number of objects from class j present in the node.

4.3.5 Measuring Error

Evaluation of classifier performance is based on the number of correctly and incorrectly classified instances. Counts of misclassifications are tabulated in a

confusion matrix and this is a commonly used error metric. Table 4.6 depicts the confusion matrix for the dichotomous *pass/fail* classification problem.

		Predicted Class	
		PASS	FAIL
Actual	PASS	`True Positive (TP)`	`False Negative (FN)`
Class	FAIL	`False Positive (FP)`	`True Negative(TN)`

Table 4.6. Summary confusion matrix. *The matrix is comprised of 4 indices which outline the classifiers predictive ability. Ideally the off-diagonal elements should be equal to 0 for zero misclassification.*

where `True Positive (TP)` and `True Negative (TN)` entries are correct classification predictions made by the model and `False Positive (FP)` and `False Negative (FN)` are incorrect classification predictions.

- `TP` is the number of correct predictions that an instance is a PASS
- `TN` is the number of correct predictions that an instance is a FAIL
- `FP` is the number of incorrect predictions that an instance is a PASS
- `FN` is the number of incorrect predictions that an instance is a FAIL

Although the confusion matrix provides all the information needed to determine how well a classifier performs, summarising the information into a single performance metric is convenient and allows direct comparison of different models. There are a number of standard terms for a two class matrix. Most classification algorithms seek to attain the highest accuracy, or equivalently, the lowest error rate when applied to a test data set. Accuracy and error rate are therefore common metrics.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{\texttt{TP} + \texttt{TN}}{\texttt{TP} + \texttt{TN} + \texttt{FP} + \texttt{FN}} \tag{4.7}$$

and

$$\text{Error rate} = \frac{\text{Number of Incorrect Predictions}}{\text{Total Number of Predictions}} = \frac{\texttt{FP} + \texttt{FN}}{\texttt{TP} + \texttt{TN} + \texttt{FP} + \texttt{FN}} \tag{4.8}$$

4.4 Chapter Summary

This chapter describes supervised learning methods. A brief introduction to machine learning terminology is given and then the chapter describes, with examples, one-rule classification and decision tree induction. Methods of selecting attributes and computing the information gain are discussed. Error metrics are discussed along with the presence of false positives and false negatives in a confusion matrix.

In Chapter 5, exploratory data methods are evaluated with semiconductor batch test data. Parallel coordinate monitoring plots are constructed and a PCA model is developed to reduce the dimension of the monitored variables. Fault detection and classification methods are discussed. Decision tree induction is also described for the semiconductor batch test data. Chapter 6 outlines two process states and shows how fault detection and classification is performed.

Chapter 5

Results

Contents

"There are three kinds of lies: lies, damned lies, and statistics"
(Mark Twain (1835-1910)

5.1 Exploratory Data Analysis

Traditional methods of monitoring via control limits can work well for smaller data sets but when dimensions are large, the requirements for an accurate monitoring scheme become more arduous.

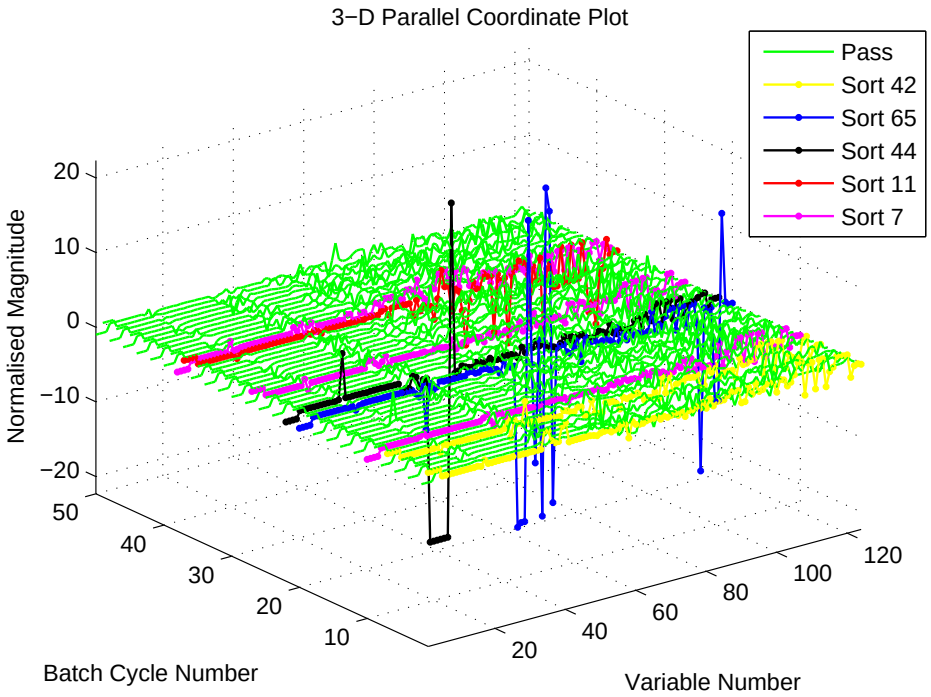
One of the most fundamental stages in data analysis is interpretation of results and this *understanding* is heavily influenced by exploratory data analytical techniques (EDA). EDA can be used to signal data quality, formulate ideas and turn raw data into information. One of the greatest challenges in modelling a data set with a large number of dimensions is spatial representation. Parallel coordinate analysis (parallel-coords) is a visualisation technique that has been universally recognised as an effective method of dealing with multidimensional numerical data.

5.1.1 Parallel-coords Monitoring Plots

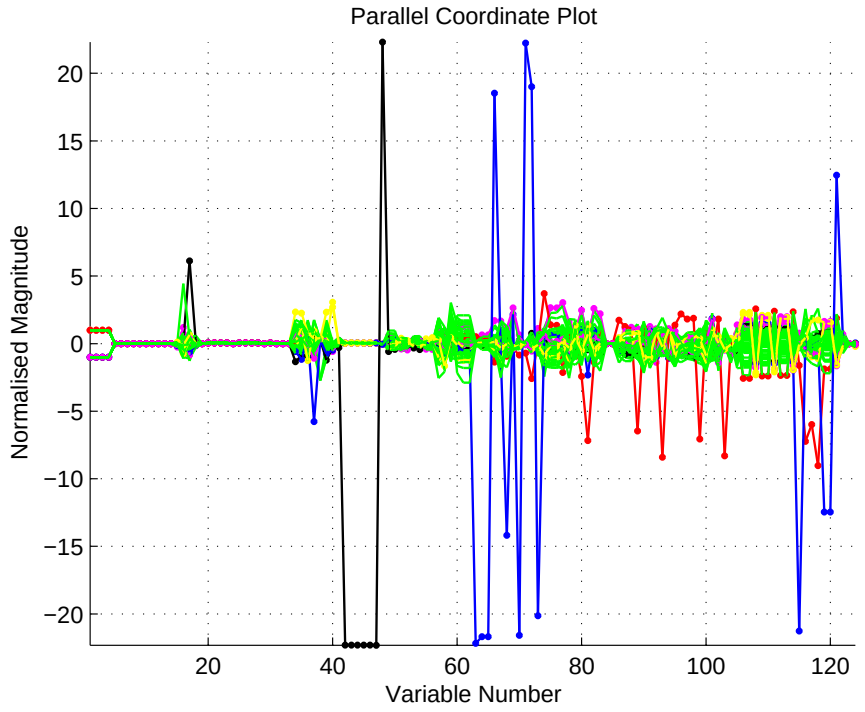
A slight adaption to the method of parallel-coords is shown in Figure 5.1 (a). A third dimension allows for each successive cycle, $\mathbf{n}$ ($\mathbf{n} \approx 50$), to be plotted and generates a three dimensional representation of a multivariate test batch. Each cycle represents a device under test (DUT) from a batch. The importance of this visualisation method is that the outcome (*pass or fail*) can be quickly identified as testing is in operation. It can serve as a low level monitoring system, with a control window, $\mathbf{n}$, representing the process in real-time with all of the constituent variables. If there are too many successive fails, the test can be stopped and the rogue variables can be identified through the coordinate plot.

The semiconductor data contains a outcome sort index which is known as the bin-sort index and the results are 'binned' into common groups in the range between 0 to 100. Test passes are binned as 1 and test fails take on a value greater than 1, *i.e.* fail > 1. This enables specific targeting of faults and fault conditions. Developing this idea further is Figure 5.2, where the response surface of a normally operating process is compared with the response surface of an abnormally operating process. The differences shown represent actual fault conditions throughout a test. The benefit of this technique is that it rapidly provides a graphical perspective of the problem domain.

Figure 5.3 shows the normal behaviour of a tester complete with operational



(a)



(b)

Figure 5.1. A monitoring Parallel Coordinate plot. (a) *50 cycles of semiconductor batch data visualised in 3-d clearly showing particular fails and significant failure outliers.* (b) *X-Z axes of (a) showing similar parallel-coords plot to that in Figure 5.6 (a).*

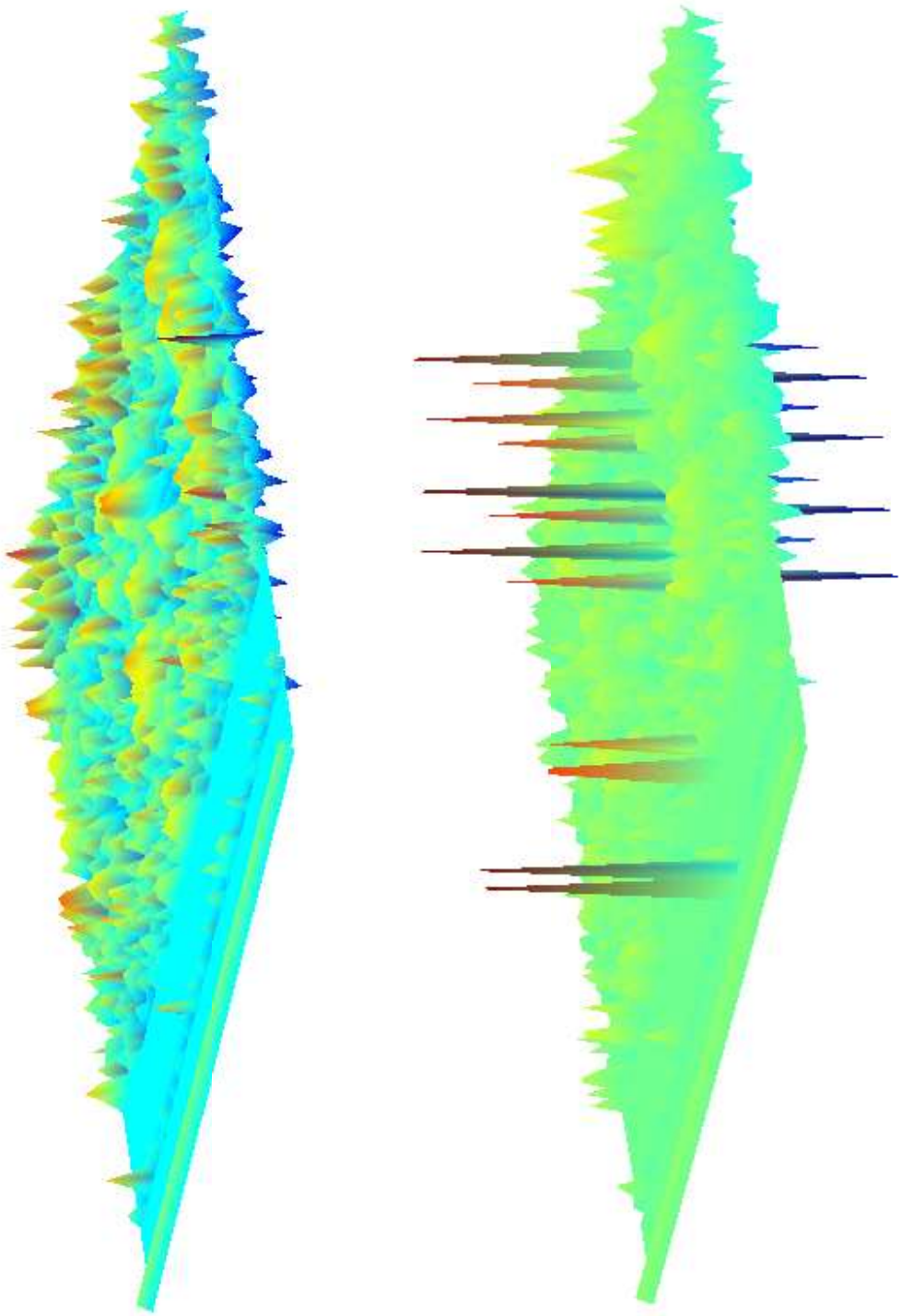


Figure 5.2. Response surface monitoring plots. *Comparison of tester response surfaces. The top plot shows test passes only and the bottom plot shows both passes and fails. As both figures are scaled identically, the incidence of a fail is clearly visible.*

limits. It is important to stress that the upper and lower limits in this Figure are derived from the batch statistics and not from an arbitrary external source. There are three main clusters in this data set, which are populated by the majority of the process variables. However, a distinction between pass and fail is not evident in all of the variables. This is a limitation in this method of analysis, but it does serve as a useful direct data visualisation tool. Subsequent analysis based on these assumptions are developed in the next Sections, 5.2 and 5.3. A parallel-coord plot is a generalisation of a two-dimensional Cartesian plot. The idea is to reduce the number of orthogonal axes by drawing the axes parallel to each other in order to obtain a planar diagram in which each **d**-dimensional point has a unique representation. This allows interpretation of these plots in a manner analogous to two-dimensional Cartesian scatter plots. A major benefit of parallel-coords analysis is the ability to visualise dense information with each dimension being of equal importance.

The fundamental reason for this type of analysis is to identify different system operational states though the raw data structures. It also provides the opportunity for human *Pattern Recognition*, which is by far one of the most efficient (in two dimensions). Visualisation of variable correlation in parallel-coords is achieved by observing the polylines that connect two variables together. Any occurrence of a *cross-over point*, Wegman (1990), or an intersection between two ordinates gives is an indication of correlation. When $\mathbf{x_1} \propto \frac{\mathbf{1}}{\mathbf{x_2}}$, the intersection point takes place between the two ordinates, its exact position determined by the slope, $\mathbf{m}$. If $\mathbf{m} = -1$, then this occurs at midpoint. If however, $\mathbf{x_1} \propto \mathbf{x_2}$, the intersection point lies outside that of the two ordinates. When $\mathbf{m} = 1$, in Euclidean projective geometry, coincident lines meet at infinity and hence all coincident lines with the same slope will meet at the same *ideal point*. This is illustrated in Figure 5.4.

The transition between the test variables gives a dynamic view of the process and of the combination of variables yielding conditions of *pass* and *fail*. It is relatively easy to isolate any variables that do not change significantly or indeed at all. These are known as a *redundant* test variables in modelling because the information they yield is neither useful in distinguishing between class or in describing the process. An example of this redundancy is shown in Figure 5.5 where the data are pseudo labelled from $X_1 \rightarrow X_{124}$. It is clear that certain regions of

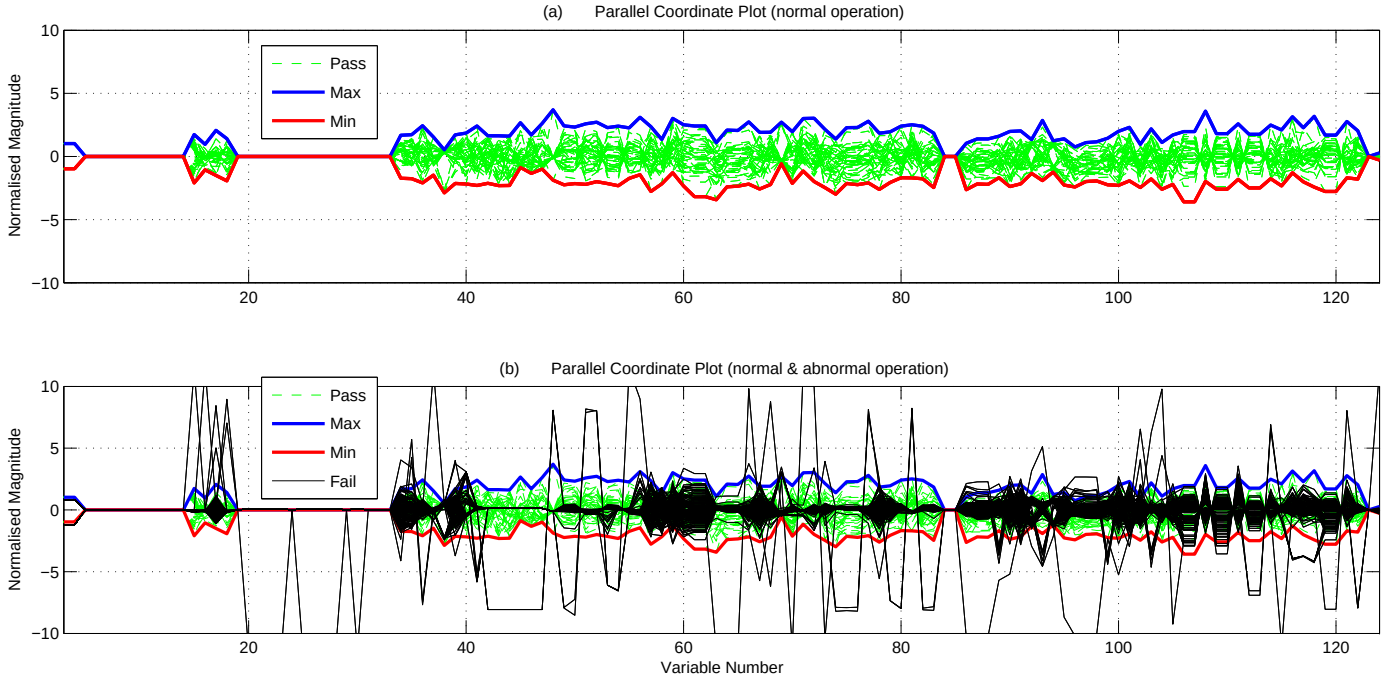


Figure 5.3. Normal and abnormal process operation states. (a) *The blue and red lines represent normal operational limits for the process variables* and (b) *the black lines indicate test failure and abnormal operation. It is evident that the incidence of a fail can occur within the normal operational limits for some variables. Variables that require more consideration in a model are depicted by the departure from normal.*

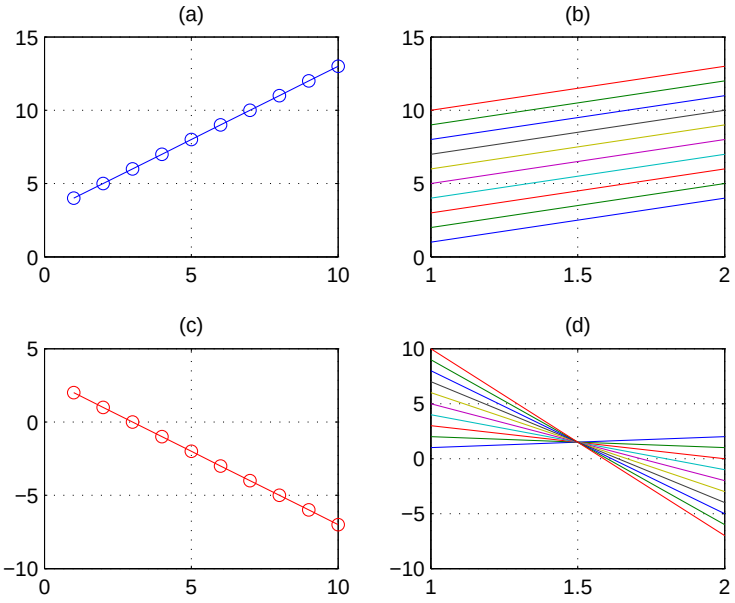


Figure 5.4. Correlation and intersection characteristics. (a) $\mathbf{x_1}$-$\mathbf{x_2}$ *scatter plot, with slope* $\mathbf{m} = 1$, (b) *Parallel coordinate representation of (a),* (c) $\mathbf{x_1}$-$\mathbf{x_2}$ *scatter plot, with slope* $\mathbf{m} = -1$, (d) *Parallel coordinate representation of (c).*

sparsity exist in the data and three such regions are readily identifiable

1. Variables $X_1 \Rightarrow X_{14}$
2. Variables $X_{19} \Rightarrow X_{33}$
3. Variables $X_{84} \Rightarrow X_{85}$

This is illustrated with semiconductor batch test data. Figure 5.5 (a) and (b), detail a high dimensional data set ($d = 125$) mapped to a two dimensional space. The process dynamics become apparent and can be further emphasised through variable *brushing*. *Brushing* is an interactive way of identifying trends and common characteristics amongst variables and for visualisation purposes. Brushed variables are shown in 'green'. In Figure 5.5 (a), the brushed polylines are representative of a normally operating process. This delineation can be thought of as the process operating normally (sometimes referred to as *steady state*) subject to stochastic disturbances. In Figure 5.5 (b), the brushed polylines represent abnormal process operation and test failures. This concept is reinforced through careful scrutiny of Figure 5.5 (a) and (b) and Figure 5.8. It is evident that certain variables do not contribute to a distinction between *pass* and *fail* or to the systematic trend of the process. The dependent variable in this case is test outcome (*i.e. pass* or *fail*) and there are a number of redundant test vectors in the data. From this *'eye-balling'* approach, it is reasonable to assume that certain variables will not contribute towards a model and therefore can be excluded in the building phase. On subsequent analysis, the majority of the redundant variables were digital (*i.e. discrete*) tests which held a constant value. Parallel-coord plots have the ability to show either commonly or independently scaled variables. Common or global scaling represents an overall picture of variability between each variable, but can be heavily skewed by differences in variable magnitude. Independent scaling removes the requirement for data normalisation and allows each ordinate to represent the full variable range. This is shown in Figure 5.6. In Figure 5.6 (a), common scaling of the variables reveal definite outliers and variable behaviour when a test *fail* occurs. The dynamics of the process are not as evident however, as the variables with the largest magnitudes dominate the plot. The trend depicts test passes. In Figure 5.6 (b), the influence of individual scaling can be seen.

(a)

(b)

Figure 5.5. Parallel-coord plot. The data are pseudo labelled $X_1, X_2, .., X_d$ for brevity. (a) *Semiconductor batch data visualised through parallel-coords analysis. The 1st variable is test sort (pass = 1 and fail > 1) and all subsequent 124 variables shown. Bin-sort 1 (test pass) is shown through the 'green highlighting' or brushing of observations.* (b) *The brushed polygonal lines indicating test failures and abnormal situations.*

With parallel-coords, it is possible to select high yield batches and directly compare them to low yield batches. This helps identify the variables that are more likely to have an effect on the outcome of the test.

To further explore the use of parallel-coords, the summary statistics of eighteen semiconductor batch tests are plotted together. These data, as depicted in Figure 5.7 (a), are extracted from the same tester over differing time periods and it can be seen that there is considerable variation in batch yields (84.2% → 98.8%). Prediction of yield is a difficult task in this non-deterministic system. Ideally, the null hypothesis states that all batches tested will yield identically on the same tester, but Figure 5.7 (b) reveals this not to be the case as there is significant differences in the variable contributions which determine yield. Therefore, in order to successfully predict yield, the stochastic component is needed in order to understand what is a likely outcome and what is not. This emphasises that a test *fail* may be due to the tester and not the device under test (DUT). The top three yielding batches are identified through *brushing* in the reduced subsection of variables.

It is important to mention that the tester constraints are the same on each tester. There is a certain similarity in the performance of three different testers in examining different batches of the same product. This is shown in Figure 5.8. So, whilst there is similarity between the testers, there is an inherent variation within the actual test itself. This was expected, but to what degree was unknown.

5.2 Unsupervised Learning and Fault Detection

The goal of Principal Component Analysis (PCA) is to determine a new set of dimensions (attributes) that better capture the variability of the data and it has several appealing characteristics. Firstly, PCA tends to identify the strongest pattern in the data. Secondly, often most of the variability of the data can be captured by a small number of dimensions. Thirdly, the effect of filtering the data by variability reduction can eliminate noise.

An appealing characteristic of PCA is data compression or a reduction in data dimension. The primary objective of data compression for this application is to minimise the amount of monitored variables for a given process. The term

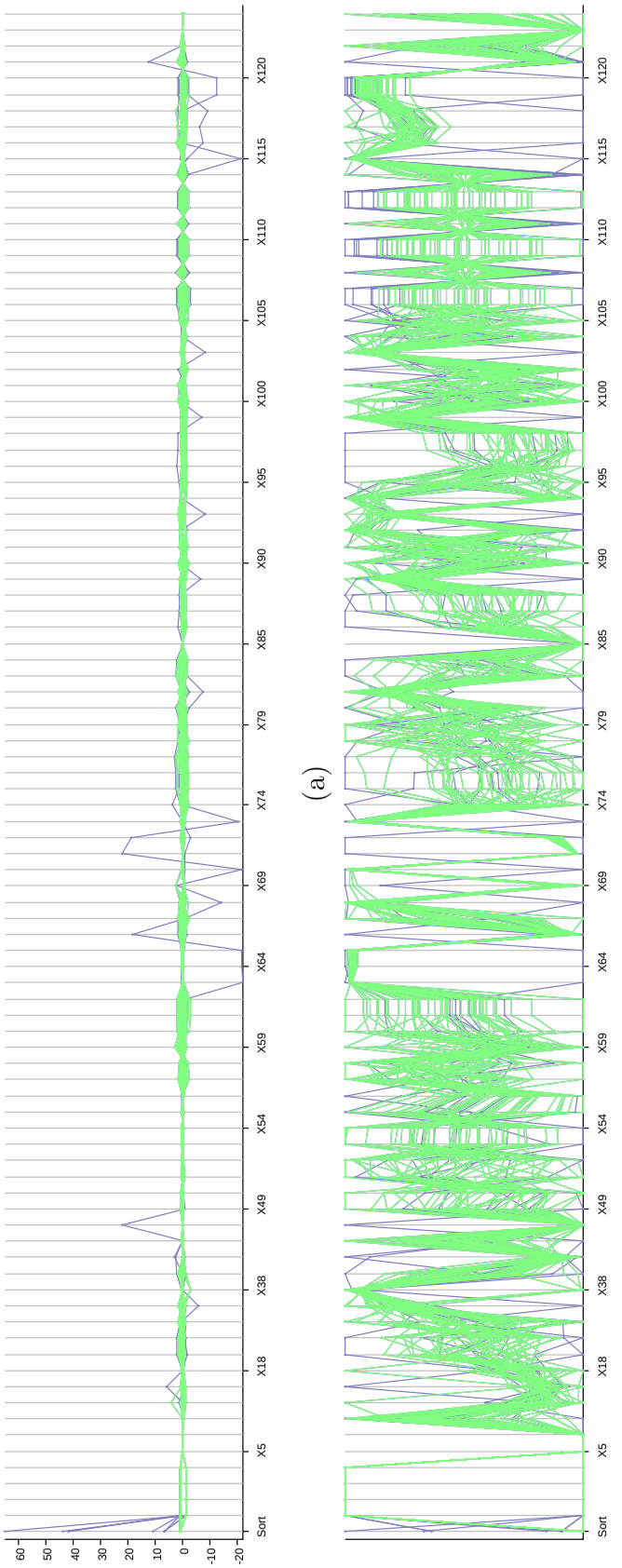


Figure 5.6. Variable scaling in parallel-coords. (a) *Global or common scaling which clearly shows definite outliers (test fails) and variable magnitudes* and (b) *Individual scaling which shows the systematic trend or dynamics of the process. Bin-sort 1 (test pass) is brushed in green.*

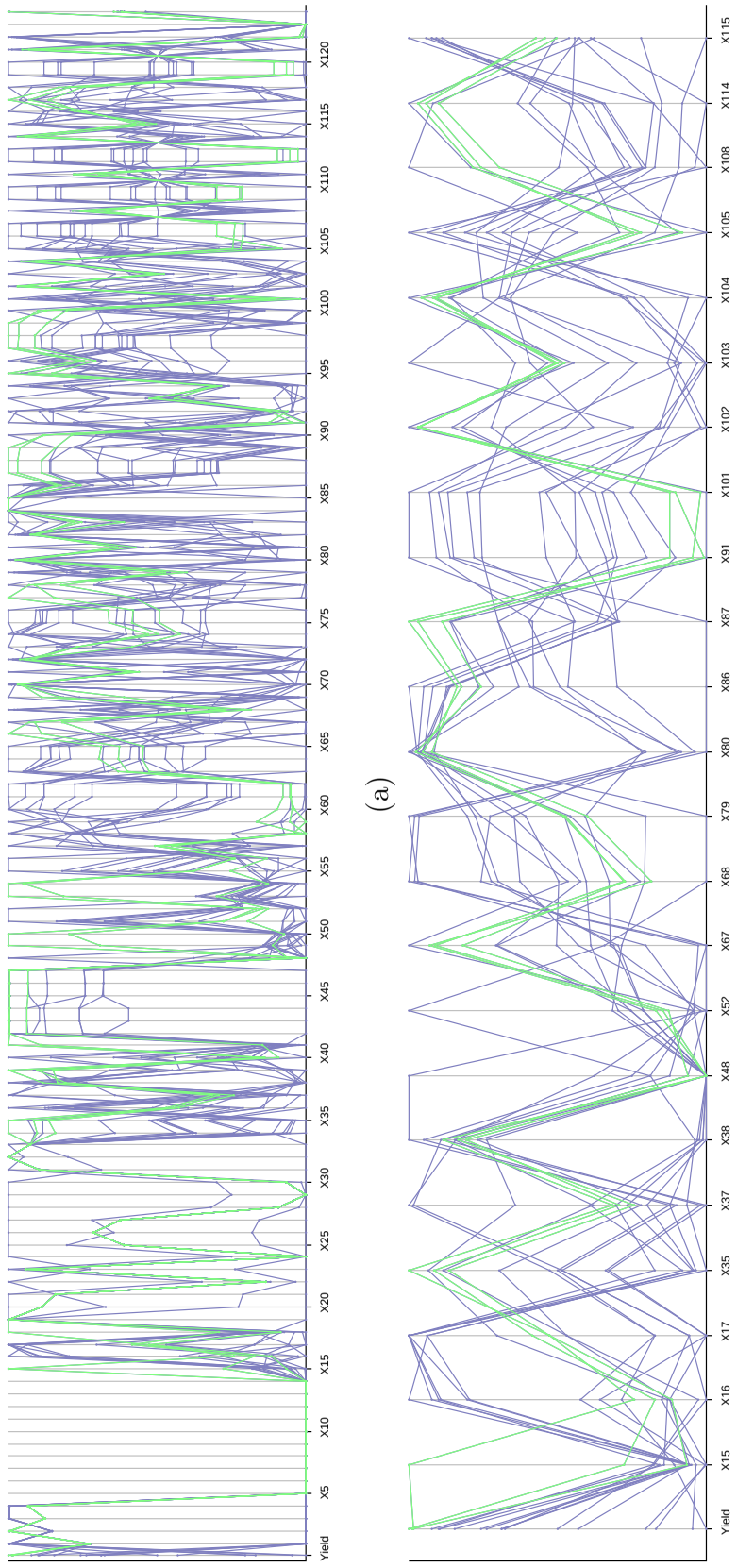
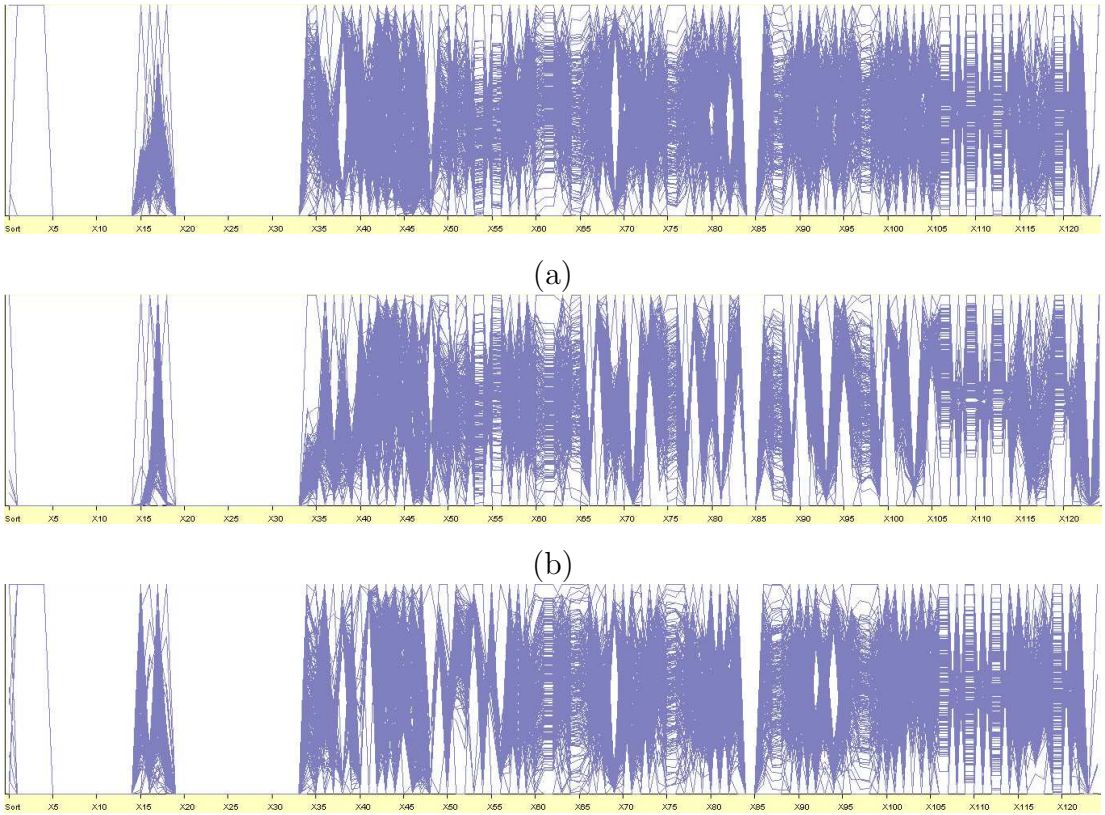


Figure 5.7. Summary parallel coordinate plot. (a) *Summary statistics of 18 batches shown with yield as dependent variable. The 3 highest yielding batches are brushed to show variable contributions* (b) *A reduced data set indicating variable contributions for the 3 highest yielding batches.*



(a)

(b)

(c)

Figure 5.8. Similarity of tester performance. *3 independent batches from 3 different testers, indicating both similar process dynamics and redundant variables. This inspection allows for variable selection in the model building phase.*

dimensionality reduction is often reserved for those techniques that reduce the dimension of a data set by creating new attributes that are a weighted combination of the original variables.

PCA can be used for database exploration for seeking periods of abnormal process behaviour and diagnosing possible causes for such behaviour. Examining the behaviour of the process data in the reduced projected space as defined by the latent variables can identify stable operation, process drifts and sudden, abrupt changes.

5.2.1 Data Preparation

The purpose of data preparation is to manipulate and transform raw data so that the information content can be exposed, or at least be made more accessible. This is based on two *a priori* assumptions,

- the proposed solutions
- the proposed analysis methods

both of which can change significantly. The data referred to in this thesis are multivariate, of high dimension and contain different internal structures. Therefore constructing a suitable model is a continual process which requires many iterative stages and pre-processing. The terms *'data'* and *'data sets'* are used to describe the *attribute -v- instance* relationships in standard matrix or table format. These are obviously intrinsically linked together but at different stages as *data* are used in the initial model building phase and *data sets* are used exclusively in the modelling phase. The initial exploratory work carried out showed the data as both *mixed-mode* (*i.e.* analogue and discrete test vectors) and correlated (*i.e.* positively and negatively). The dichotomous outcome of the process (*i.e. pass* or *fail*) was converted to a yield statistic.

5.2.2 Data Pre-Processing

Data pre-processing is used to filter out any noise components that may influence the analysis, extract features and reduce the dimensionality of the original signal. Pre-processing also attempts to retain as much relevant information as

possible without redundancy. It is found that often the information lies not with the individual variable but rather how the variables change with respect to one another, *i.e.* how they co-vary. Therefore, some sensor measurements are effectively redundant and complementary and PCA seeks to identify this, Gallagher et al. (1997b) and Gallagher et al. (1997a). One main pre-processing task is to ensure high quality data samples to guarantee a valid process description for extracting knowledge and model based diagnosis as well as detecting structured relationships among variables, Sobrino & Bravo (1999).

Pre-processing was initially achieved through data classification. The majority of the variables in the semiconductor test were analogue (*continuous*) and the remainder digital (*discrete*). As the test is preformed digitally, the analogue signals are quantized digital signals, as opposed to traditional analogue/continuous signals. The distinction between the two classes (*continuous and discrete*) is performed on the basis of sampling rate and quantized levels. Not all the digital variables are of binary (0,1) type and some are quantized signals that exist on a number of different levels. The distinction is a question of sampling granularity, with a highly quantized digital signal being a close approximation to a true analogue signal.

Matrix conditioning is important in PCA where the occurrence of a zero (*implying a near singular matrix*) causes problems in singular value decomposition. In order to prevent this, a certain number of digital test vectors were removed from the data in the pre-processing stage. This was achieved by calculating summary statistics of the batch data and eliminating variables devoid of higher order statistical moments (*i.e. Skewness and Kurtosis*).

5.2.3 Normal Operating Condition Model

In developing a process model it is usual to focus on data that have favourable trends (*i.e.'test passes'*). This heuristic approach enables the construction of a *Normal Operating Condition* or NOC model. This model is a representation of the process operating normally and without fault and is similar in concept to the parallel-coord plot in Figure 5.5. It is usual for process data to be compared at certain stages. Commonly, data that is extracted from a process operating normally and within bounds may be compared to that when the process has been

known to drift or behave abnormally. This concept is used frequently in process monitoring and is known as fault (*anomaly*) detection.

A thorough knowledge of the process dictates the overall success of the monitoring and fault diagnosis scheme. An essential starting point is analysis of historical data. The first stage is pre-processing the data, applying limit files (to remove redundancy) and data normalisation $\mathbf{N} \sim (0, 1)$. Next, the data are partitioned and a subset of known goods (*i.e. test passes*) are analysed. PCA is performed on the subset of data, yielding a latent variable representation of the problem domain. Subsequent batch data are normalised with respect to the NOC model statistics (μ, σ), and the result can be used to identify a shift from normal to abnormal operation. Visualisation is achieved through two or three dimensional score plots which contain information on how the *samples* relate to each other. These PC score plots represent linear combinations of the original data. However, when the number of dimensions d increase, *i.e.* $d > 3$, three-dimensional score plots do not fully represent the model, and other combinatorial indices are used. Variable selection can be achieved through contribution plots, which form an integral part of a fault detection and isolation scheme. Table 5.1 outlines each stage in the development and application of a NOC model.

5.2.4 Model Validation and Cross Validation

Cross Validation (CV) is a procedure that is used to fine tune model complexity. It is intended to avoid the possible bias introduced by relying on one particular division into *test* and *training* components. As bias is incalculable, the total error is used. The error on the validation set decreases up to a certain level of complexity, then stops decreasing any further, or even increases if there is significant noise. This error is known as the Root-Mean-Square Error of Cross Validation or RMSECV. The primary purpose of CV is to determine the number of Principal Components (PCs) used in the model. Normal procedure in PCA is to list a table of eigenvalues or PCs along with percentage variance for each eigenvalue. This summarises how well the model describes the original data through a given number of PCs.

Data Warehouse $\leftrightarrow$ Database

$\Downarrow$

Data matrix $[m\,rows, n\,cols]$

$\Downarrow$

Pre-process & partition $[pass\,\&\,fail]$

$\Downarrow$

Reduced data set, k $(k \leq min\,[m, n])$

$\Downarrow$

Normalisation, k_N $(k_N \sim (0, 1))$

$\Downarrow$

Covariance matrix

$\Downarrow$

Singular Value Decomposition

$\Downarrow$

Model Eigenvalue %

$\Downarrow$

Score & loading vectors

$\Downarrow$

Apply NOC stats $[\mu, \sigma]$ to batch data

$\Downarrow$

PC score plots $[2\,\&\,3D]$

$\Downarrow$

$\mathbf{T}^2$ & $\mathbf{Q}$ indices

$\Downarrow$

Contribution plots

$\Downarrow$ $\Downarrow$ $\Downarrow$

Fault Detection & Isolation

Table 5.1. NOC model flowchart. *Flowchart outlining each constituent stage in the development and application of a NOC model. Grouping outlines the similarity between stages. Multivariate quality control charts form an inherent part of any FDI scheme.*

Principal Component Number	Eigenvalue of Cov Matrix	% Variance Captured per PC	% Variance Captured Total
1	2.31e+001	24.09	24.09
2	1.33e+001	13.87	37.95
3	1.05e+001	10.91	48.87
4	6.76e+000	7.05	55.91
5	4.71e+000	4.91	60.82
6	3.68e+000	3.83	64.65
7	2.78e+000	2.89	67.54
8	2.34e+000	2.44	69.98
9	1.96e+000	2.04	72.03
10	1.83e+000	1.91	73.93
11	1.60e+000	1.67	75.60

Most commonly, as a first indication of the model, the eigenvalues are plotted in a plot. Raymond B. Cattell is credited with the name scree plot and it is essentially an eigenvalue fallout and it takes the name *'scree'* from the rubble or fallout at the bottom of a cliff. Figure 5.9 shows a combined eigenvalue scree and RMSECV plot. In the scree plot, the line stabilises after the 8^{TH} PC and a subjective decision is required as to the number of PCs used. A more thorough method is to calculate the RMSECV, and plot the error function. The error decreases until a point of inflection (*saddle point*) where it begins to rise again. Although a subjective process, the amount of PCs to include will influence the overall result. Too many PCs will over parameterise the model and add noise and too few will give an incomplete, inaccurate representation of the process. Jackson (1991) outlines four ways of choosing the number of PCs in a model:

1. Scree plots to identify *elbow/knee* point in plot
2. Disregard lower order eigenvalues *i.e.* $\lambda < 1$
3. Include 70 - 90% cumulative variation
4. RMSECV to identify *saddle point*

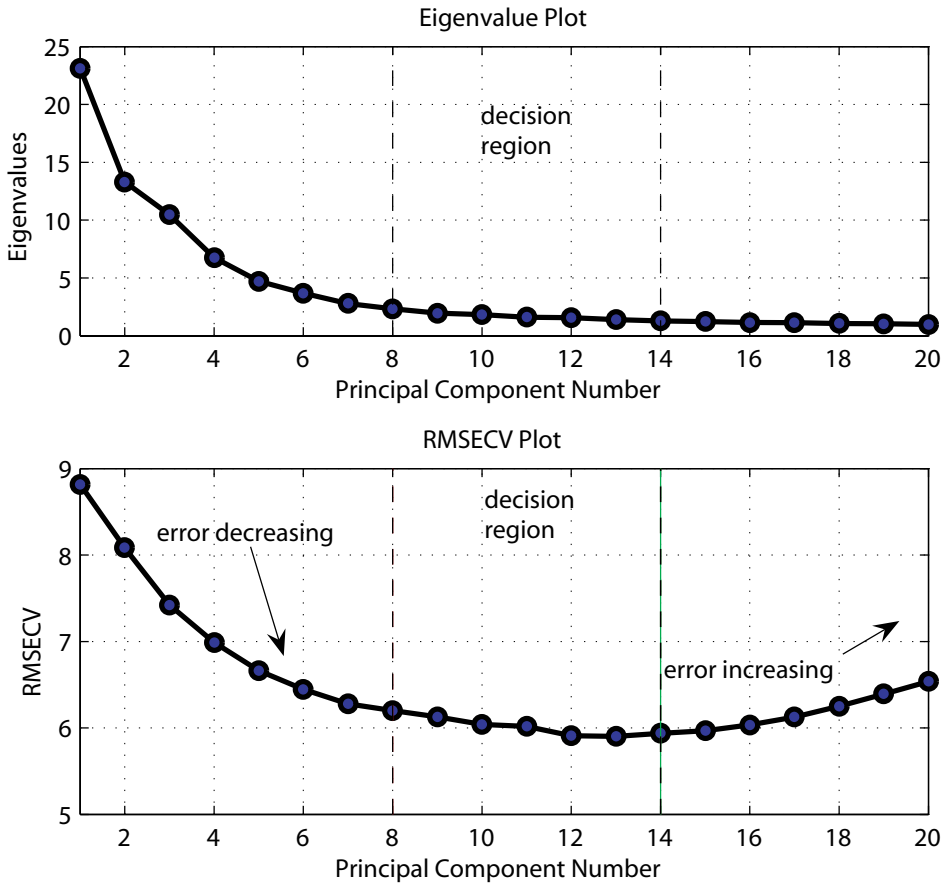


Figure 5.9. Combined scree and RMSECV plot. *A scree plot of the eigenvalues is shown including a subjective decision region. In the RMSECV plot, the error decreases but subsequently increases with the number of PCs. The number of PCs, based on the RMSECV projection, should be between 8 and 14.*

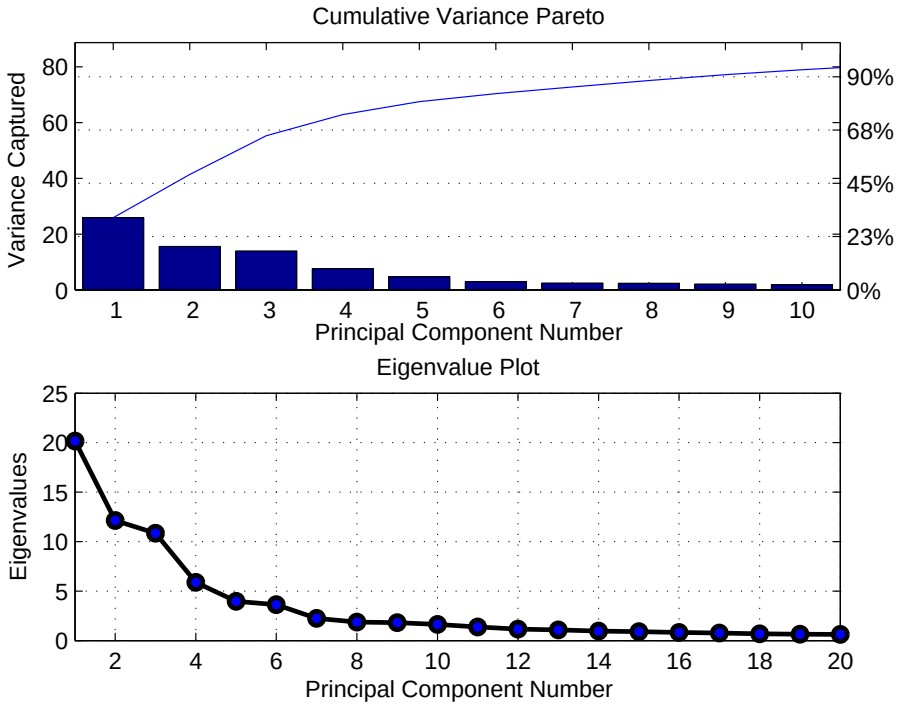


Figure 5.10. Cumulative variance and eigenvalue plot. *This shows the cumulative variance in a combined barchart-pareto plot. The eigenvalue contributions are shown below.*

5.2.5 PCA Score Plots

Figure 5.10 shows a cumulative variance Pareto plot and eigenvalue contribution input to the model. Figure 5.11 shows normal operating condition batch data projected on the first three PC score vectors with accompanying 2σ and 3σ confidence ellipses. This model has been created with batch data extracted under normal process conditions and can be used to test subsequent batch data to indicate significant departure from the normal model. This indicates a change in process condition or development of an abnormal situation. It is important to note that although Figure 5.11 is a very detailed and useful visualisation tool, a 3-dimensional score plot will only fully represent a NOC model with three score loading vectors. For instances with more than three loading vectors multivariate monitoring indices are more frequently employed. For this model, there were six loading vectors which explain 72.6% of the variation in the data. This reduces the dimension of the problem domain from $d = 80 \rightarrow d = 6$. Figure 5.12 details a

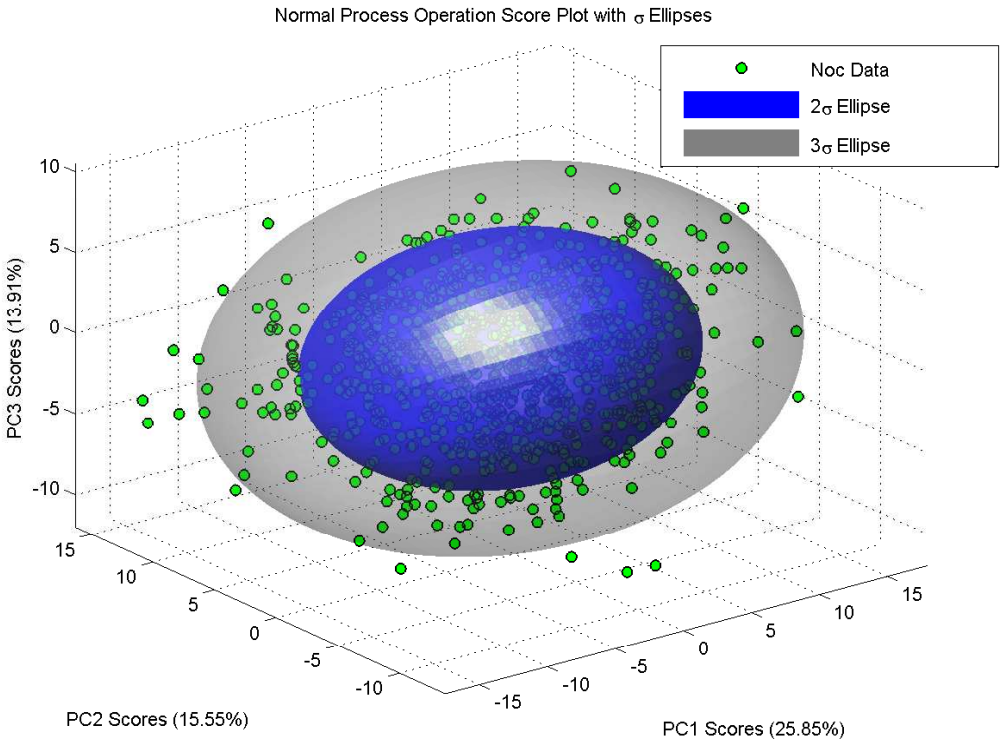


Figure 5.11. NOC score plot. *A visual representation of the NOC model including* 2σ *and* 3σ *confidence ellipses. This is constructed from the first 3 PC's, each showing their particular % contribution to the model. The PC1-PC2-PC3 space describes 57.4% variation in the data.*

steady state NOC model used with batch data. This is a 2-dimensional abstraction as PC scores 1 and 2 are plotted together. This shows batch data projected onto a NOC model and it is apparent that there are instances of significant departure from the NOC cluster (indicated as '•' in Figure 5.12). Disassociation with the NOC cluster is reflective of underlying differences in the data and gives an indication of anomalous patterns and these findings are further verified through the use of a Hotelling $\mathbf{T}^2$ chart.

Figure 5.13 shows both $\mathbf{T^2}$ and $\mathbf{Q}$ multivariate indices. These control charts are derived from a six component PCA model which represents 71% of the original variation within a semiconductor batch test data of 50 observations of 145 process variables. These control charts effectively represent the batch run as a univariate metric for quick fault identification. Figure 5.13 (a) shows a $\mathbf{T^2}$ chart

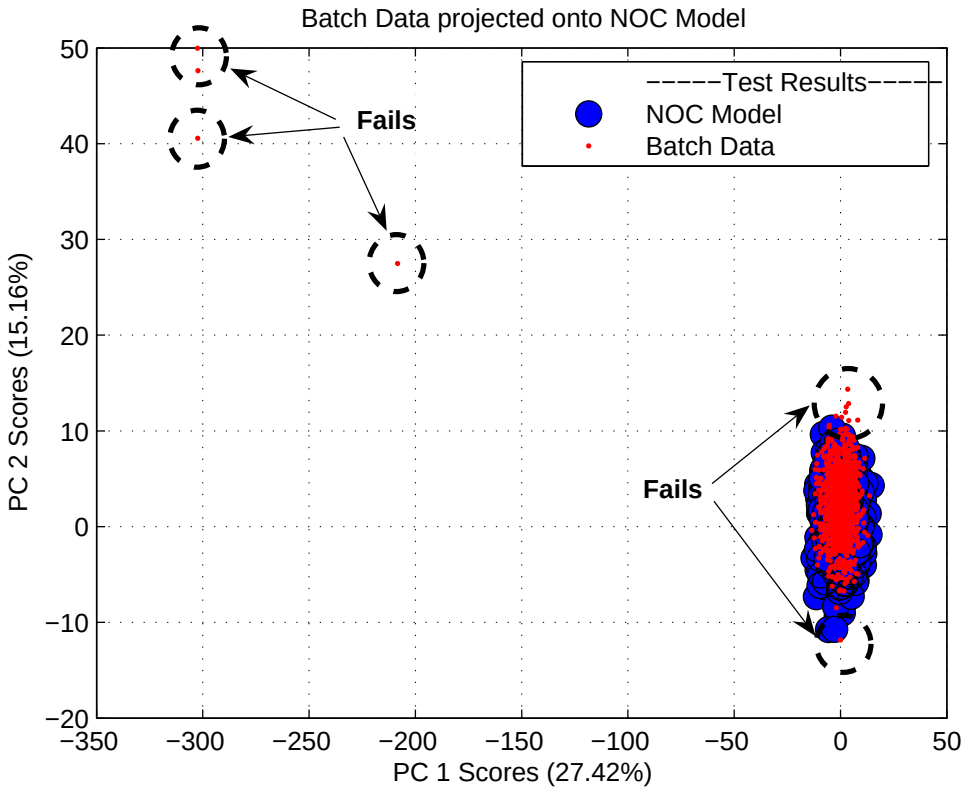


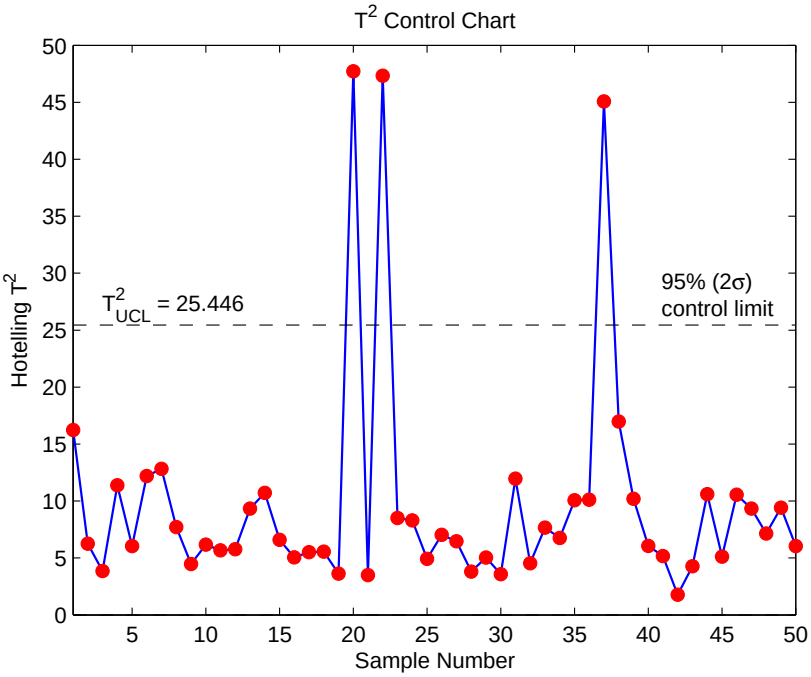
Figure 5.12. Batch Data and NOC Model. *This shows batch data projected onto a 6 PC NOC Model. Outliers from the NOC cluster are test failures, as shown in the Figure. The PC1-PC2 space describes 42.6% variation in the data.*

with three points exceeding the 95% (2σ) confidence limit. This is an indication that variation within the plane of the first k PCs of the model has exceeded that of normal stochastic or common cause variation. However, it may be insensitive to a totally new type of event, *i.e.* assignable causes. Figure 5.13 (b) shows a $\mathbf{Q}$ chart where one sample exceeds the 95% (2σ) confidence limit. This is an example of a new *assignable* or special event occurring. Together these two indices signal four faults within the batch duration of fifty DUT samples and on further examination of the original data, the yield sort code reveals this to be correct. A major benefit is the original dimension of the data is reduced and faults (*fails*) are identified by the two charts. Invariably, Hotelling's $\mathbf{T^2}$ statistic is used in conjunction with the $\mathbf{Q}$ statistic as together they characterise two orthogonal subspaces of the transformed data.

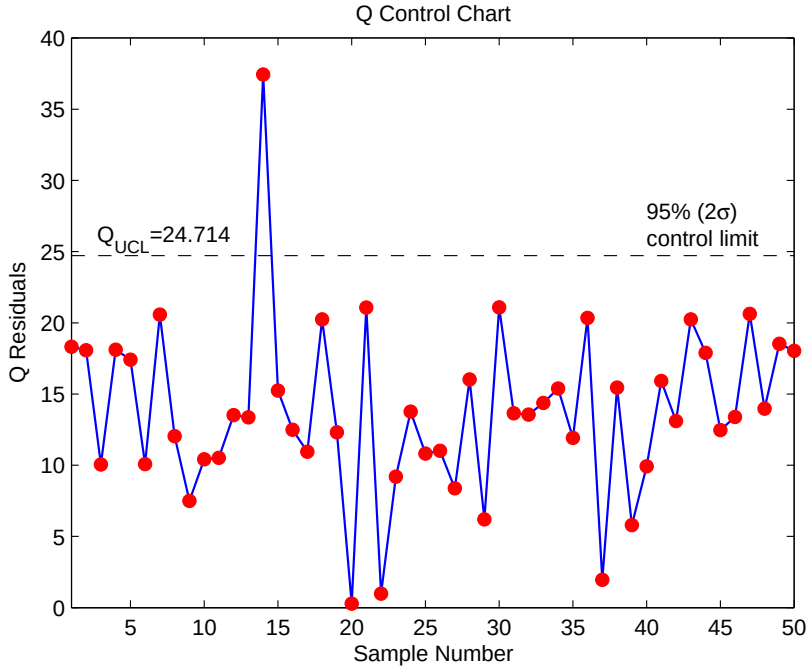
5.2.6 Contribution Plots

Although both the $\mathbf{T}^2$ and $\mathbf{Q}$ indices detect deviations from normal behaviour, they do not indicate reasons for such deviations. Variable contribution plots are required to determine the root cause of an anomaly (*fail condition*). A contribution plot provides an overview of the *'weighting'* for each of the k variables in the original data. The contributions that exhibit the greatest change are typically indicative of process variables which may be exhibit non-conforming behaviour. In essence, the application of contribution plots complete the loop from 'fault detection' to 'fault classification'. In Martin & Morris (2002), contribution plots are also known as *'root-cause'* plots as they are used to analyse the score contributions to determine variables exhibiting non-normal behaviour. Contribution plots are best described figuratively and this is done using a smaller data set.

Firstly, a fail matrix is shown which outlines a particular DUT failure (the sample number of the batch), the bin-sort code (in square brackets) and the multivariate signalling index ($\mathbf{Q}$ or $\mathbf{T}^2$). The corresponding variable contribution plots are shown in Figure 5.14. In each sub-figure, the dominant process variables are representative of a particular state or fault condition. The contribution plots are derived from the multivariate control charts in Figure 5.13. The sign of each variable represents whether the shift away from the mean (*i.e. normality*) for a



(a)



(b)

Figure 5.13. $\mathbf{T^2}$ and $\mathbf{Q}$ multivariate fault indices. (a) *The* $\mathbf{T^2}$ *statistic which shows 3 samples clearly exceeding the 95% (2σ) control limit.* $\mathbf{T^2}_{UCL}$ *as derived from Equation 3.23 is shown,* $\mathbf{T^2}_{UCL}$*=25.445* and (b) *The* $\mathbf{Q}$ *statistic which shows 1 sample exceeding the 95% (2σ) control limit.* $\mathbf{Q}_{UCL}$ *as derived from Equation 3.28 is shown,* $\mathbf{Q}_{UCL}$*=24.714. These data are extracted from a 6 component PCA model (cum var = 71%) based upon 50 observations of a semiconductor batch test, with 145 test vectors.*

given fault was in a positive or negative direction for each of the variables.

$$Fails\ (Figure\ 5.13) = \begin{pmatrix} 14 & \rightarrow & [4] & \rightarrow & \mathbf{Q} \\ 20 & \rightarrow & [7] & \rightarrow & \mathbf{T^2} \\ 22 & \rightarrow & [65] & \rightarrow & \mathbf{T^2} \\ 37 & \rightarrow & [11] & \rightarrow & \mathbf{T^2} \end{pmatrix}$$

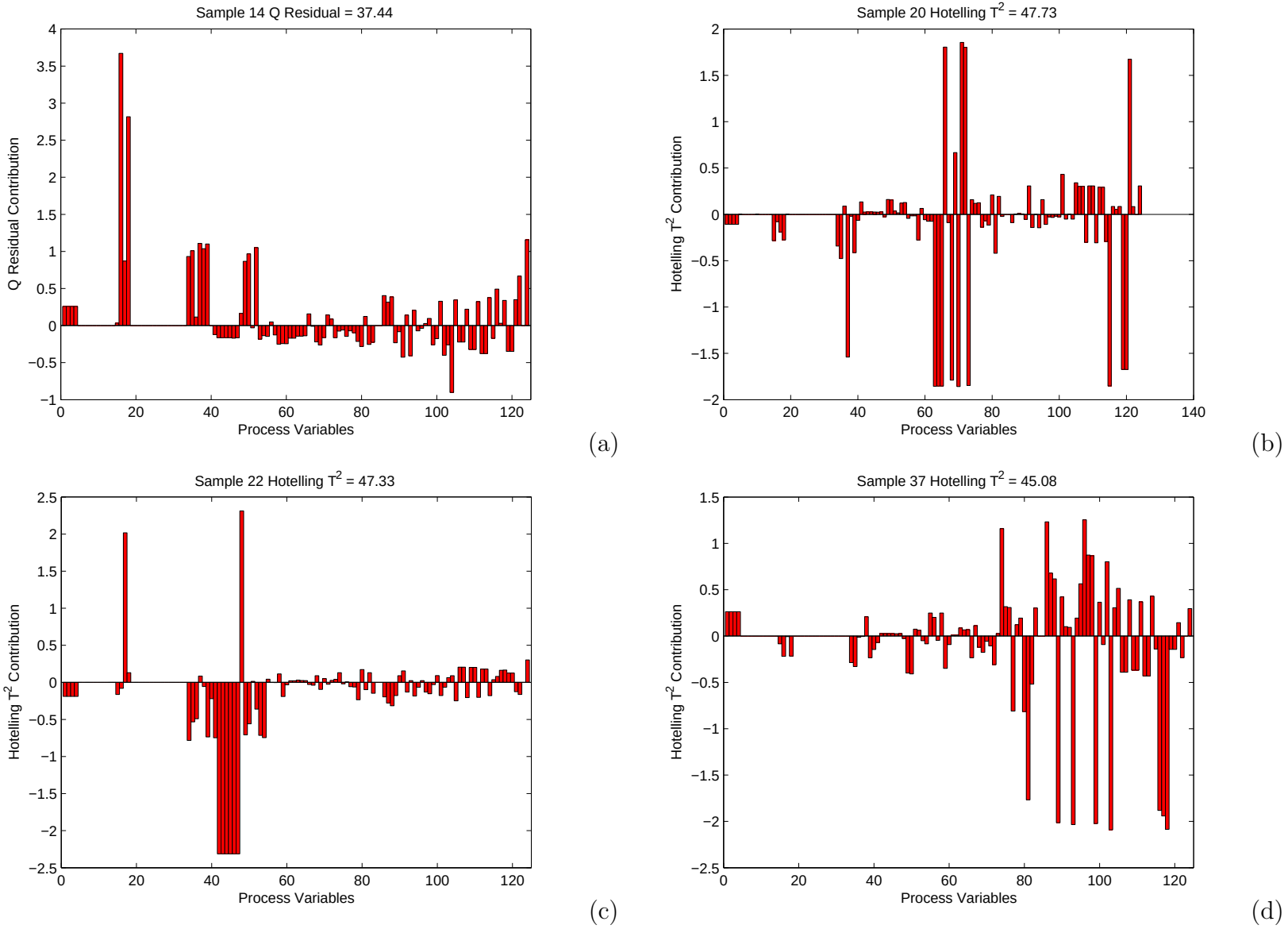


Figure 5.14. Contribution plot. *This figure shows variable contributions to each of the 4 faults. In* (a) *sample 14* **Q** *residual is shown, In* (b) *sample 20* $\mathbf{T}^2$*, In* (c) *sample 22* $\mathbf{T}^2$ *and* (d) *sample 37* $\mathbf{T}^2$*. Each subplot details a different fault condition and the variables which account for that fault, signalling a shift from normal → abnormal operation.*

5.3 Supervised Learning and Decision Tree Induction

In the previous section, no class label information was used in the analysis of the data. In supervised learning however, class label information is used as an integral part of the analysis. Supervised learning in its simplest form is generalising a dichotomous class (*i.e.* both positive and negative class examples) and finding a description that is shared by all the positive examples and none of the negative examples. This is termed *class learning*. In doing this, a prediction can be made on new data to provide or assign a class label. New data is commonly referred to as *unseen* data in supervised learning. In essence, unseen data is quite simply data that has not been used to construct the supervised learning model. In the semiconductor batch test data, a class label was available through the bin-sort code index. Although different classes were assigned to each particular fault condition, the overall classification was a dichotomous index of *pass* or *fail*. Techniques of supervised learning and anomaly detection require the existence of a training set with both normal and abnormal operation class information. This task of assigning objects to a particular predefined category is known as *Classification*.

5.3.1 Supervised Learning Through Test Constraints

In order to use the raw data to understand the process operation, the test limits were not considered in the analysis. These test limits are arbitrarily prescribed based on historical data, process capability and vendor constraints. Therefore, to consider the process without bias, these constraints were removed. However, as a visualisation exercise, it is possible to derive a test result matrix from the data using the constraints. An example of this dichotomous test matrix is given in Figure 5.15. This matrix illustrates any variable that exceeds a prescribed high or low limit during the test process. The batch size (*x-axis*) is plotted against the number of variables (*y-axis*) in matrix style format. *Blue circles* represent a high level failure and *red circles* represent a low level failure. This perspective of the data and can be used to determine the existence of failure patterns and expose fault clusters. It can also signal certain variables that consistently fail at

A	Low level failure of variables 113 & 120 ⇒ Sort 7
B	Low level failure of variable 40 ⇒ Sort 42
C	Low level failure of variable 48 ⇒ Sort 50
D	High level failure of variables 95 & 109 ⇒ Sort 87
E	Low & high level failure of multiple variables ⇒ Sort 65
F	High level failure of variable 118 ⇒ Sort 19

Table 5.2. Fault identification. *This table lists the variable & fault combinations shown in Figure 5.16. These combinations are consistent through the test.*

either level, or in a cluster with other variables. In Figure 5.16, the correlation between variable failure (both high and low limits) and tester fail sort code is shown. The '★' represent actual test failures. The test failure sort code for each '★' can be read directly from the *y-axis*. For clarity, a non exhaustive list is show and Table 5.2 details these combinations.

This information can be used in developing IF-THEN rules from which it is possible to classify alternative data sets. This is shown in the sample association rule base.

```
Sample Association Rule Base
-----------------------------------------------------
IF variables 113 & 120 fail low ---> THEN sort code 7
IF variable 40 fail low         ---> THEN sort code 42
..          ..          ..          ..          ..
..          ..          ..          ..          ..
IF variable 118 fail high       ---> THEN sort code 19
```

5.4 Decision Tree Induction

A decision tree is composed of a root node, internal decision nodes and terminal nodes. Each decision node implements a test function whose resultant outcome dictates the split. Decision tree induction is a recursive process and is repeated until a terminal node or *leaf node* is reached, at which point the value of the node

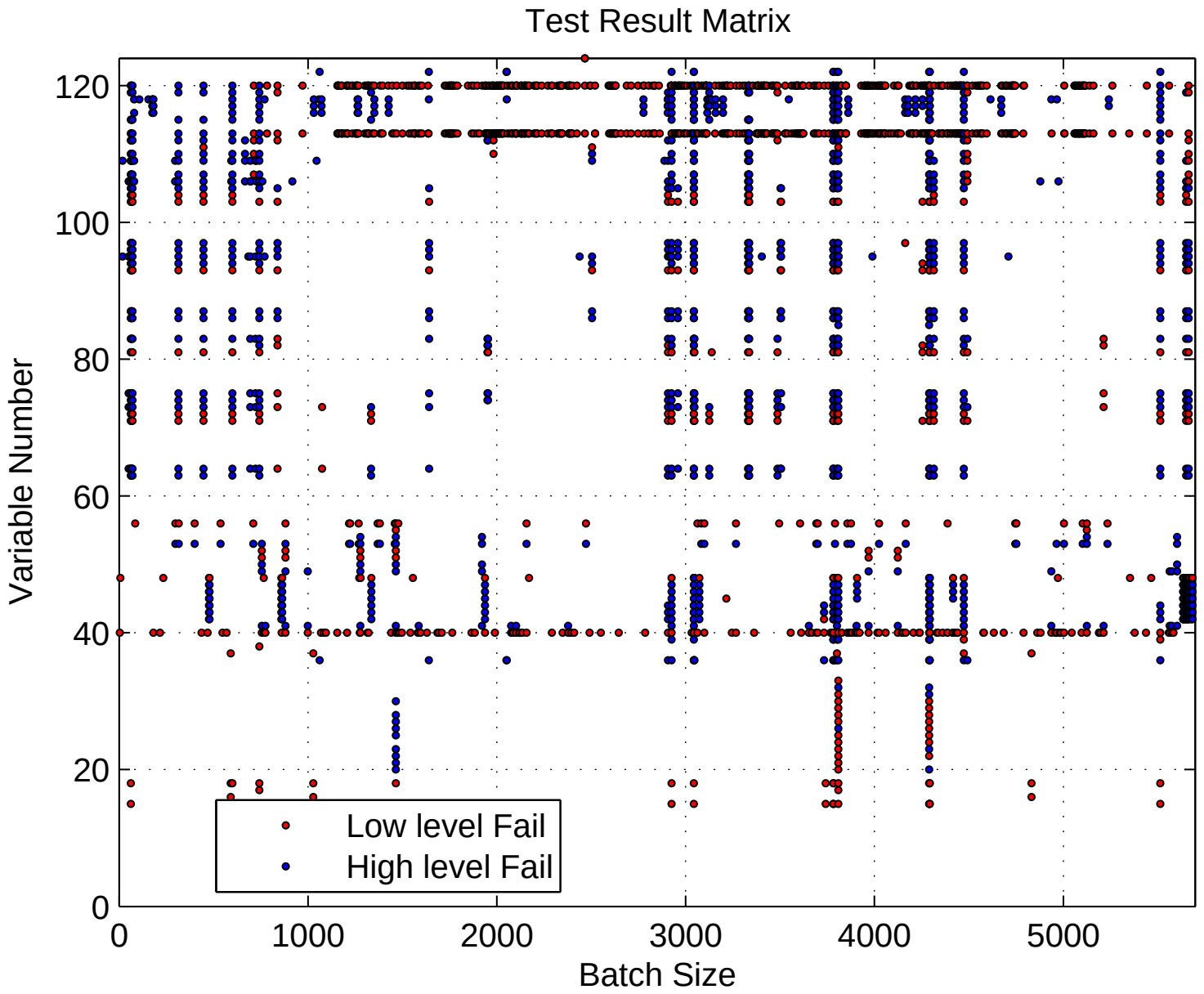
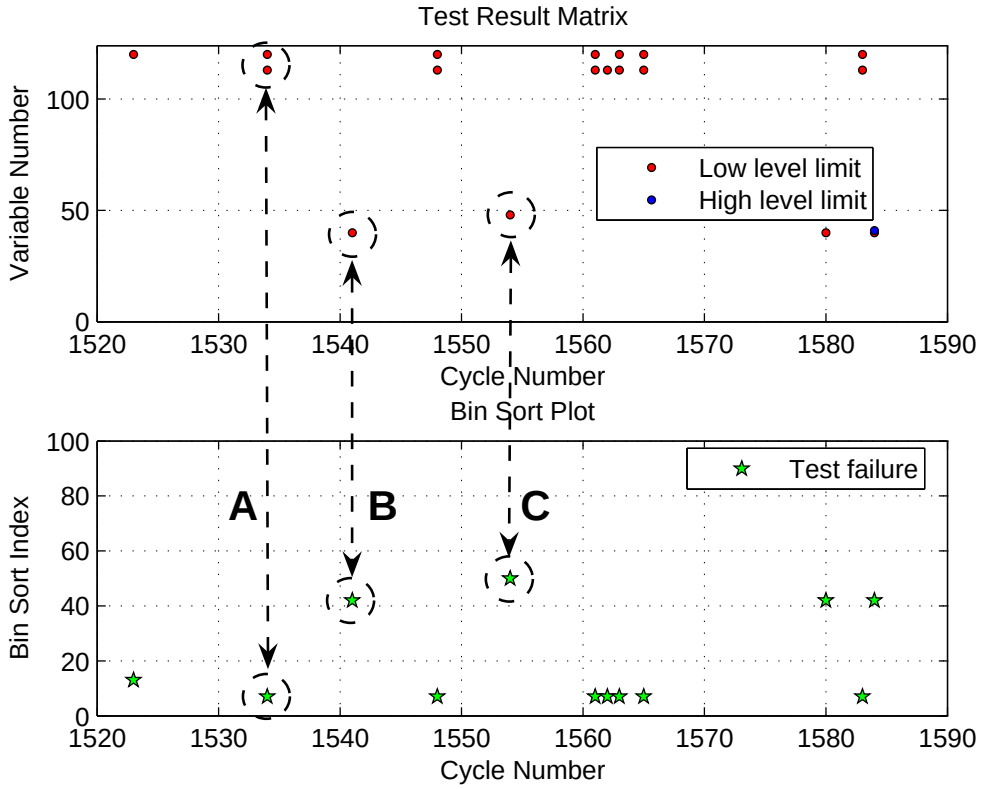
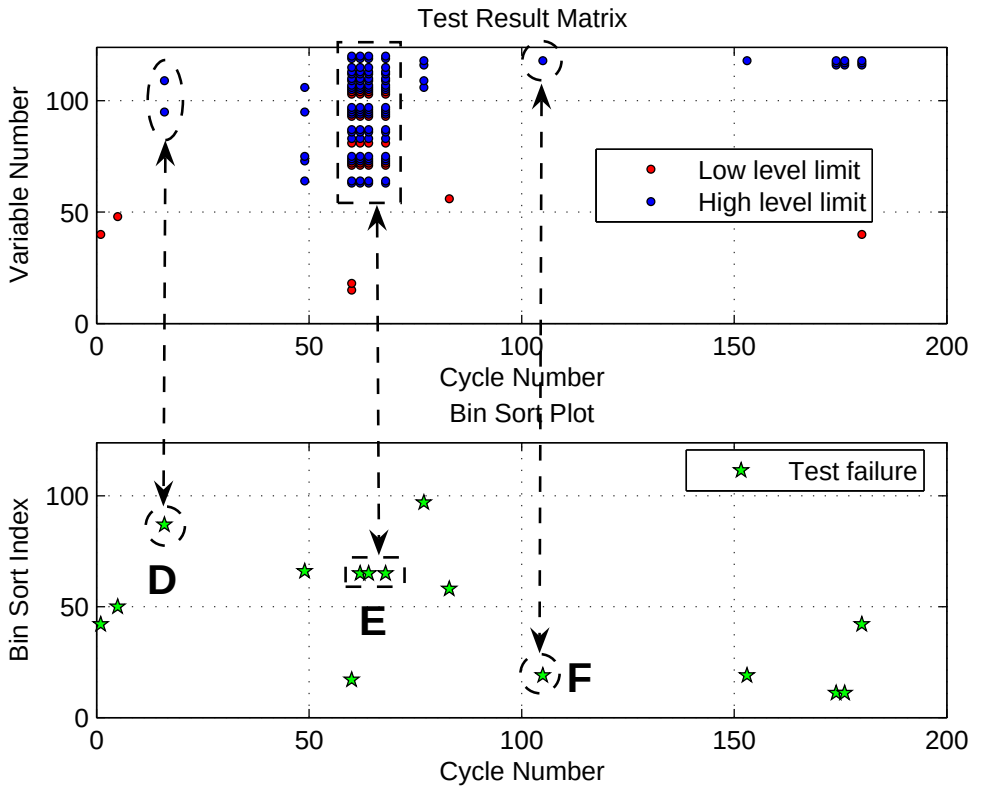


Figure 5.15. Test matrix plot. *A matrix representation of test data, showing low level and high level failures. The plots shows which variables breach either/both of the limits. This plot is useful in exposing failure patterns or fault clusters. Batch size is plotted against variable number .*



(a)



(b)

Figure 5.16. Combination of fails. *A scaled down version of Figure 5.15, showing the instances of* (a) *Low level fails* and (b) *High level fails with associated variable(s). Combination of failures are tabulated in Table 5.2.*

constitutes the output. Each terminal node is labelled with the majority vote of the data contained at that node. More information on this topic is available in Section 4.1.

As with all regression techniques, it is assumed that there is a single response variable and one or more predictor variables. If the response variable is categorical then classification trees are produced whereas if the response variable is continuous regression trees are produced.

5.4.1 Decision Tree Setup

Data pre-processing is fundamental in decision tree construction. In essence, data quality dictates the performance of a decision tree successfully classifying new, unseen data.

The decision tree classifier works to recursively partition data into subsets of similar information content. The open-source GNU machine learning package WEKA was used to construct the decision trees. WEKA implements a java variant of Quinlan's C4.5 algorithm, Quinlan (1993), called J4.8. For more information see http://www.cs.waikato.ac.nz/ml/weka.

The classifier was constructed using 10 fold cross-validation. This is a common evaluation technique to ensure that results are representative of what would be obtained on an independent data set.

Figure 5.17 shows a decision tree representation of a semiconductor batch test. The classifier is trained on the *Pass/Fail* attribute and shows a recursive partitioning schema common to all tree-based methods.

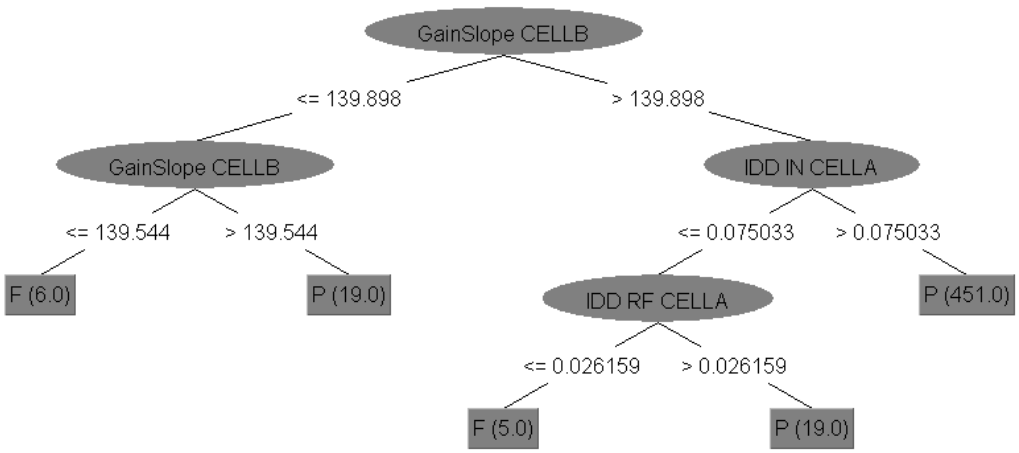


Figure 5.17. Sample decision tree. *A sample classification tree showing pass (P) and fail (F) instances through a series of attribute tests on a subset of semiconductor batch data.*

The tree can also be represented concisely through a series of logical conjunctions or IF-THEN rules (shown in DT Rules).

```
DT Rules.
------------------------------------------
GainSlope CELLB <= 139.898
|   GainSlope CELLB <= 139.544:--> F (6.0)
|   GainSlope CELLB >  139.544:--> P (19.0)
GainSlope CELLB > 139.898
|   IDD IN CELLA <= 0.075033
|   |   IDD RF CELLA <= 0.026159:--> F (5.0)
|   |   IDD RF CELLA > 0.026159:--> P (19.0)
|   IDD IN CELLA > 0.075033:--> P (451.0)
------------------------------------------
Number of leaves:   5
Size of Tree:       9
Correctly classified Instances:     498
Incorrectly classified Instances:   2
```

5.4.2 Decision Tree Test

It is important to make certain that the classifier can handle unseen data and the most common method is to provide new data to the decision nodes of the tree. The result of this analysis is neatly summarised in a confusion matrix. The confusion matrix for Figure 5.17 shows the number of correctly classified instances to be 498 and incorrectly classified instances to be 2.

As a test, a decision tree was derived from a test batch of data and subsequently tested with fresh, unseen batch data. This, in effect, is testing the predictive ability and accuracy of the classification tree. This is shown on Figure 5.18. In order to logically interpret the tree and test alternative data sets, rules are extracted and these logical conjunctions are shown in batch data decision tree rules.

```
Batch data decision tree rules.
------------------------------------------------------
CW OUTPUT 10 CELLB <= -64.4551
|   IDDtotal CELLAPout <= 0.098714
|   |   IDDtotal Idle <= 0.000048
|   |   |   RXNOISE CDMA CELLA <= -133.609:--> P (19.0)
|   |   |   RXNOISE CDMA CELLA > -133.609:--> F (4.0)
|   |   IDDtotal Idle > 0.000048:--> F (2.0)
|   IDDtotal CELLAPout > 0.098714:--> P (950.0/3.0)
CW OUTPUT 10 CELLB > -64.4551
|   CW OUTPUT 10 CELLB <= -64.2989:--> P (7.0)
|   CW OUTPUT 10 CELLB > -64.2989:--> F (18.0)
------------------------------------------------------
Number of leaves:   6
Size of Tree:       11
Correctly classified Instances:     990
Incorrectly classified Instances:   10
```

The confusion matrix is shown in Table 5.3. This represents a measure of the accuracy of the model and gives details of any errors. There are 990 correctly classified instances. This is made up of 971 `True Positives` (*passes*) and 19 `True Negatives` (*fails*). The classifier incorrectly classifies 10 instances, 8 `False`

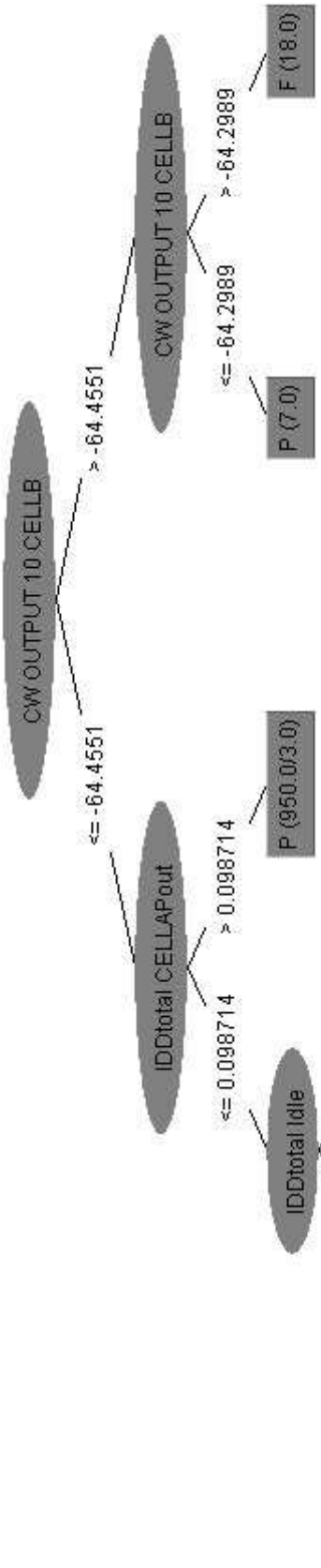


Figure 5.18. Batch data decision tree. *A classification tree derived from an entire batch of data, also showing pass (P) and fail (F) instances through attribute tests. The terminal node represents both class assignment and number of classes.*

		Predicted Class	
		PASS	FAIL
Actual Class	PASS	971	2
	FAIL	8	19

Table 5.3. Classifier confusion matrix. *This matrix is representative of the classifier in Figure 5.18 and outlines its ability to correctly associate a class.*

`Positives` and 2 `False Negatives`. This gives a classification accuracy of 99%. The confusion matrix is summarised in Table 5.4, but a more complete description is given in Section 4.3.4.

		Predicted Class	
		PASS	FAIL
Actual Class	PASS	`True Positive (TP)`	`False Negative (FN)`
	FAIL	`False Positive (FP)`	`True Negative(TN)`

Table 5.4. Summary confusion matrix. *The matrix is comprised of 4 indices which outline the classifiers predictive ability.*

Once the tree has been generated and the logical rules are in place, the next step is to test the classifier with unseen data and hence determine its predictive ability. The classifier, Figure 5.18, is used to test four data sets extracted from the same tester. The results are presented through a series of confusion matrices shown in Tables 5.5 - 5.10.

		Predicted Class	
		PASS	FAIL
Actual Class	PASS	938	3
	FAIL	9	50

Table 5.5. Batch 1 confusion matrix. *This matrix is representative of the classifier in Figure 5.18 testing batch 1. Classification accuracy is 98.8%.*

		Predicted Class	
		PASS	FAIL
Actual Class	PASS	950	3
	FAIL	2	45

Table 5.6. Batch 2 confusion matrix. *This matrix is representative of the classifier in Figure 5.18 testing batch 2. Classification accuracy is 99.5%.*

		Predicted Class	
		PASS	FAIL
Actual Class	PASS	900	5
	FAIL	8	87

Table 5.7. Batch 3 confusion matrix. *This matrix is representative of the classifier in Figure 5.18 testing batch 3. Classification accuracy is 98.7%*

		Predicted Class	
		PASS	FAIL
Actual Class	PASS	942	1
	FAIL	5	52

Table 5.8. Batch 4 confusion matrix. *This matrix is representative of the classifier in Figure 5.18 testing batch 4. Classification accuracy is 99.4%*

		Predicted Class	
		PASS	FAIL
Actual Class	PASS	940	2
	FAIL	45	12

Table 5.9. Tester HP004 confusion matrix. *This matrix is representative of the classifier in Figure 5.18 testing a batch extracted from an alternate tester-handler combination, HP004. Classification accuracy is 95.295%*

		Predicted Class	
		PASS	FAIL
Actual Class	PASS	971	3
	FAIL	12	14

Table 5.10. Tester HP005 confusion matrix. *This matrix is representative of the classifier in Figure 5.18 testing a batch extracted from an alternate tester-handler combination, HP005. Classification accuracy is 98.5%*

The ultimate goal is to create a classifier that has good generalisation performance. In order to determine this robustness, the classifier was tested with two data sets from totally separate testers. The predictive capability of the classifier under these conditions is shown in Tables 5.7 - 5.8. The incidence of `False Negative` detection rate (incorrectly classifying a *pass* as a *fail*) is increased compared to previous results. Conversely, the `False Positive` rate (incorrectly classifying a *fail* as a *pass*) is consistent with previous findings. Of these two prediction errors, a lower `False Positive` rate is more desirable. This decrease in sensitivity is due to differences in tester-handler combinations.

5.5 Chapter summary

This Chapter has given a broad description of methods used in Exploratory Data Analysis (EDA), Unsupervised Learning and Supervised Learning. In order to illustrate how these tools can be used in the context of semiconductor batch test data, two process states are identified and will be discussed in Chapter 6. These

process states are:

- Known good operation
- Normal batch operation (including abnormal situations)

Clustering, parallel-coords and NOC modelling are evaluated with the process in each state. This is presented in Section 6.1.

Chapter 6

Discussion

Contents

"If you can't convince them, confuse them"
(Harry S. Truman (1884-1972))

6.1 Differential Process States

In order to illustrate the different process modelling and classification techniques and their relevance, two process states are identified and subsequent analysis is described and evaluated. These are taken from normal process batch data and are representative of different modes of operation

(a) Known good operation (without anomalies *i.e. pass only*)

(b) Normal batch operation (including anomalies *i.e. pass & fail*)

6.1.1 Exploratory Data Analysis

In this section, the Exploratory Data Analysis (EDA) methods explained in Chapter 5 are outlined and discussed for each of the two process states.

6.1.2 Clustering

To begin, a cluster dendrogram of known good data is constructed through K-Nearest Neighbour *KNN* analysis. Cluster analysis is an unsupervised learning technique used for classification of data and most clustering methods are based around the assumption that samples that are close together in the measurement space are similar and therefore likely to belong to the same class. Figure 6.1 shows a cluster dendrogram of batch data that were extracted when the process was operating normally and without anomalies. The vertical lines in the dendrogram indicate which samples are linked and the horizontal lines in the dendrogram indicate the length of the link (*i.e.* the distance between the linked groups). The data are autoscaled which involved mean centering and standard deviation adjustment. Mean centering ensures that data are interpretable in terms of variation about the mean and normalising to unit variance removes the influence of different variable scales. A dendrogram is a picture of how the data relate to one another, and in general three pieces of information can be extracted from them

Weight. Approximate cluster membership. The weight of each cluster is represented by the number of leaves the dendrogram leads to. As each leaf (endpoint) is equally spaced along the y-axis of the dendrogram, the weight of a cluster is its percentage of the total height of the dendrogram.

Compactness. Within cluster similarity. This represents the minimum distance at which the cluster comes into existence.

Distinctness. Between group dissimilarity. This represents the distance along the x-axis from the point it comes into existence to the point at which it is aggregated into a larger cluster.

In Figure 6.1, the dendrogram shows the existence of similarly clustered samples based on their Euclidean distance. The most significantly different sample in the dendrogram is sample number 9 (the branch leading to the leaf is distinct).

The relative compactness of the clusters is approximately equal as the structure is flat and therefore the distance at which the clusters are formed is similar. The most similar cluster are sample numbers 45 & 29, with the least distance to *KNN*. The data used in this clustering assignment were extracted when the process was operating normally and without any fail conditions or anomalies. In conclusion, the sample sequence of the clustering assignment (*i.e.* the y-axis) would undoubtedly change with respect to the data set, but for normal operating conditions, the distance to *KNN* would be within a certain threshold. This point is illustrated in Figure 6.2. Figure 6.2 illustrates a cluster dendrogram which has been generated using batch data selected from a tester. For the purpose of the illustration, the data contained both *pass* and *fail* conditions and the number of samples is small for clarity purposes. It is evident that there is a significantly different cluster located at the bottom of the Figure. The sample numbers which constitute this cluster are [49, 53, 56, 51] and based on their departure from the normal cluster, one would suspect these samples to be failures. In addition to these, sample numbers [65, 1, 5, 61, 41] at the top of the dendrogram would appear to be located away from the *mean* cluster assignment. As clustering is an unsupervised method, *i.e.* there are no associated classes for the data, the easiest error checking method is to reference the original data set and determine the class assignments for the outlying clusters. Arunajadai et al. (2004) look at a failure model which is based on fault type occurrence and frequency. Figure 6.3 uses the same data as Figure 6.2 except the samples are clustered with respect to their classes instead of their sequence. This is for visualisation purposes and it is seen from Figure 6.3 that these extremities in the dendrogram are indeed test *fails*. The class label information ($P \rightarrow pass$ and $F \rightarrow fail$) is given on the y-axis, and two failure cluster are identified. These clusters are consistent with Figure 6.2 and represent $fails$ in the data. In summary, clustering is a useful method to show which samples in the d-dimensional data space are similar but it does not convey any information as to which process variables are useful for class separation, *i.e.* distinguishing a $pass$ from a $fail$.

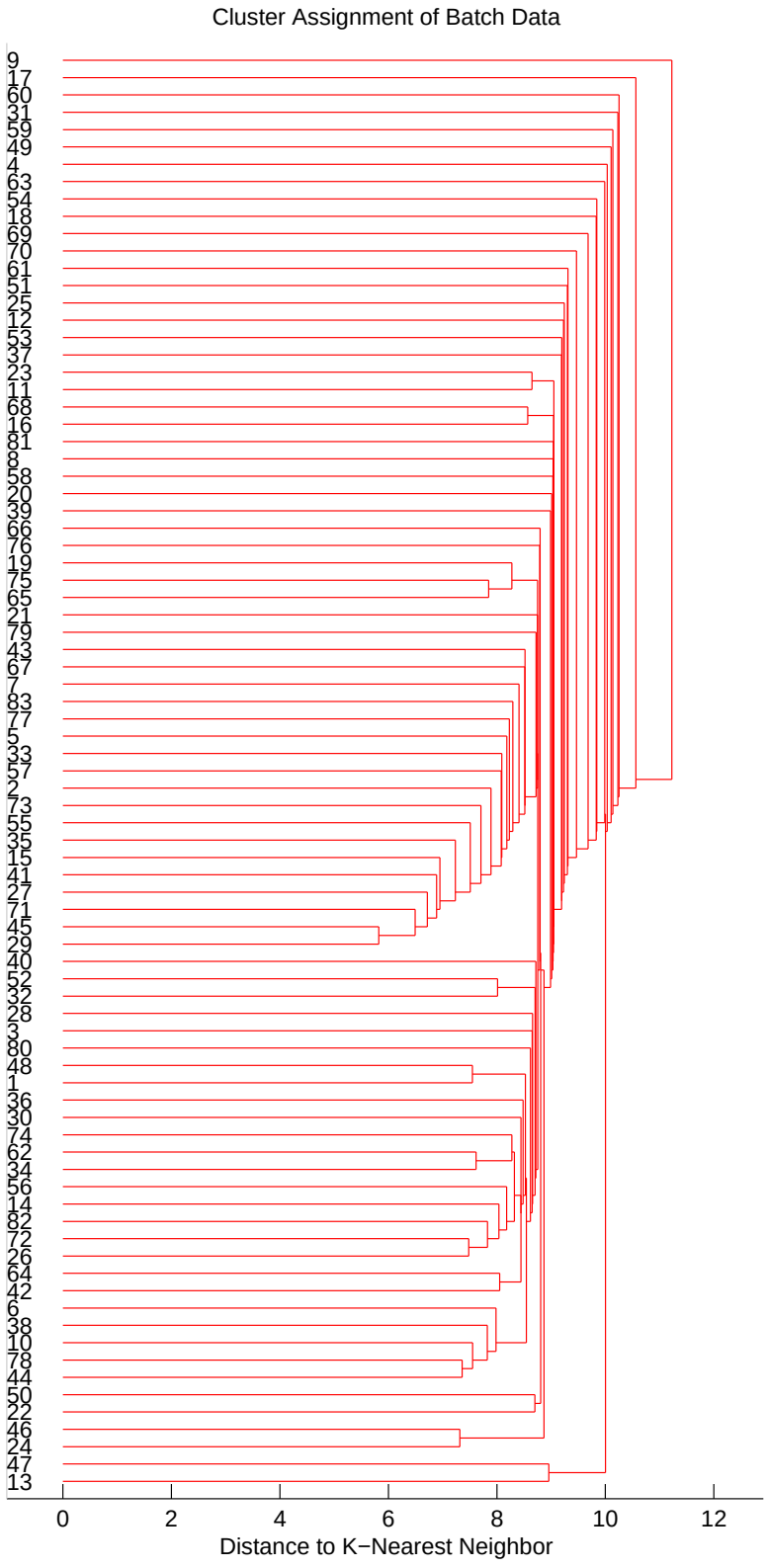


Figure 6.1. Known good operation cluster dendrogram. *The data were extracted for the clustering assignment when the process was performing normally and without anomalies. DUT cycles are shown on the y-axis.*

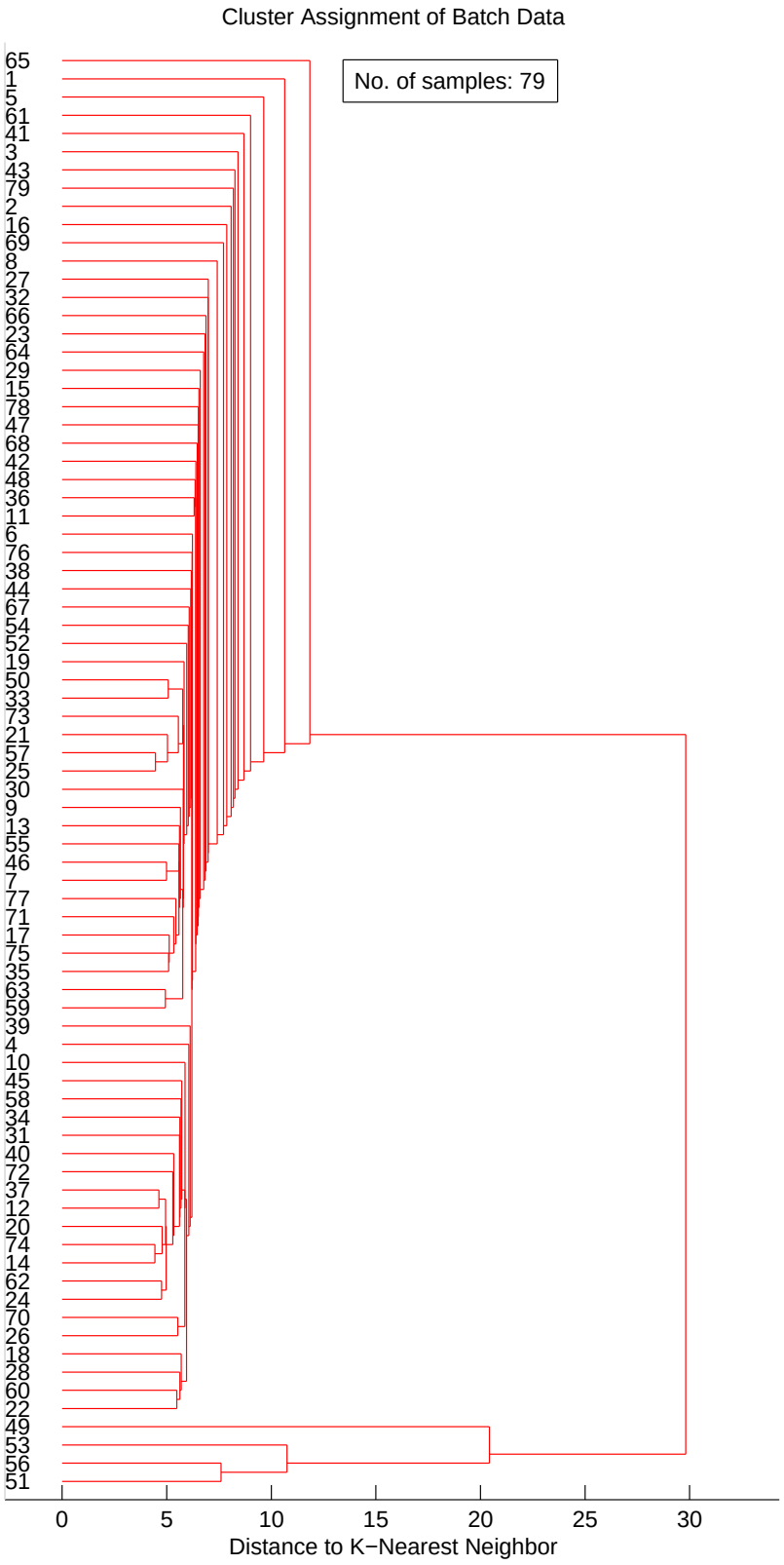


Figure 6.2. Normal process data cluster dendrogram. *A resulting dendrogram for a nearest neighbour cluster assignment on 79 samples. The vertical lines indicate sample linkage and the horizontal lines indicate the distance between groups.*

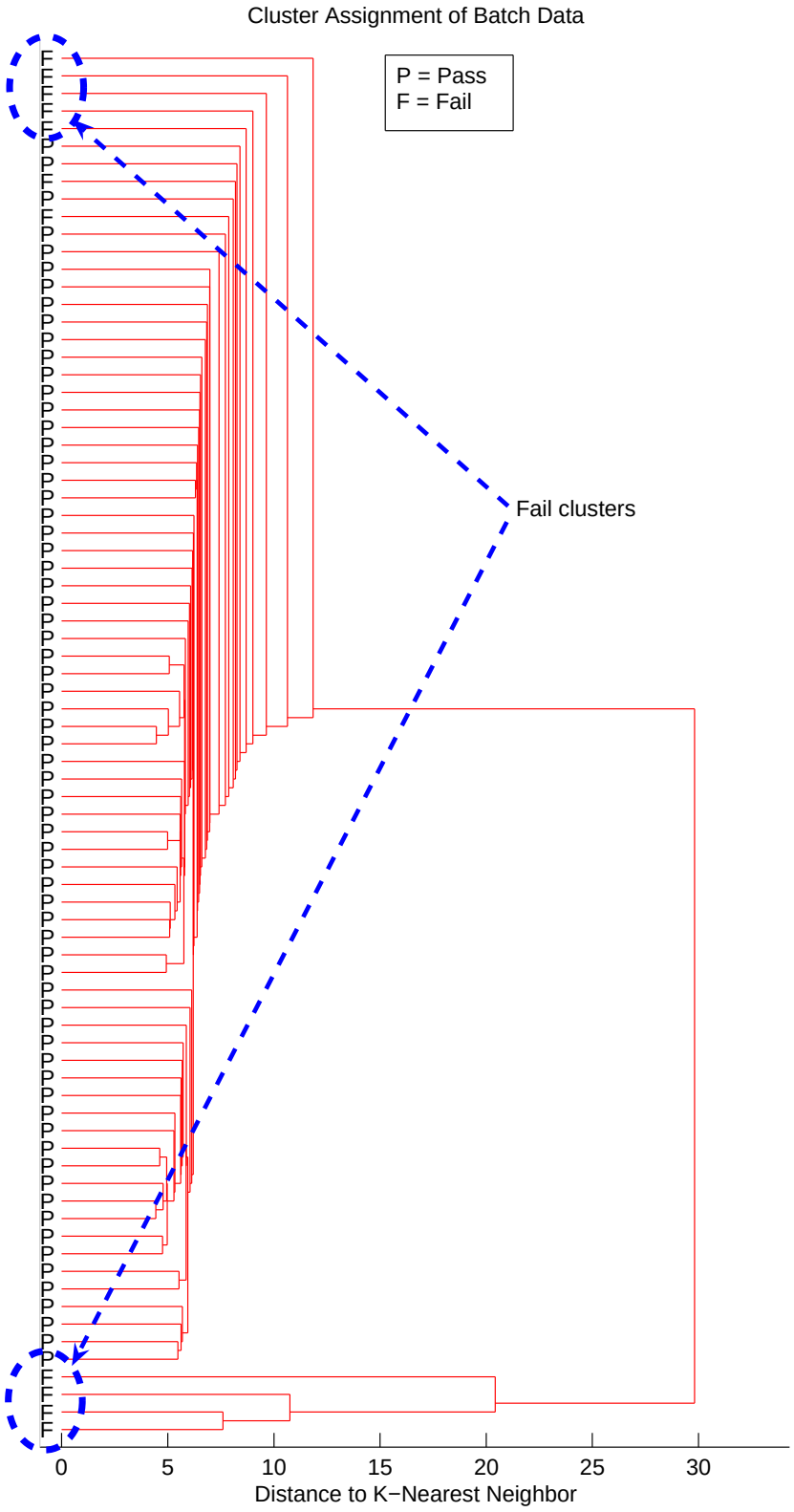


Figure 6.3. Cluster dendrogram including class labels. *This dendrogram is identical to Figure 6.2 save for the class label information ($P \rightarrow$ pass and $F \rightarrow$ fail). 2 fail clusters are apparent, outlined by the dotted ellipsoids.*

6.1.3 Parallel Coordinate Analysis

An exploratory method which gives an indication of the differentiating class variables is parallel coordinates analysis. In this thesis, parallel-coords are used for data visualisation in an attempt to capture and rationalise process dynamics. The method of parallel-coord monitoring plots is discussed more completely in Section 5.1.1 and their inclusion here is to highlight the inherent differences in process states. In Figure 6.4, the process is known to be operating in control and yielding at 100%. The two subplots in Figure 6.4 reveal the dynamics of the process and the stochastic behaviour of constituent variables. One aim of data exploration is to identify correlation between variables and two regions are identified in Figure 6.4 (b). The polylines representing the data in *region 1* and *region 2* are consistent and do not change when standardised. This suggests that they are correlated or redundant. The resulting parallel-coord plot when the data contains anomalies are shown in Figure 6.5. The slight difference in the parallel-coord plots of Figure 6.4 and Figure 6.5 is due to the fact that the data belongs to different testers. In Figure 6.5 , three regions are indicated where the variables remain stationary. This hints at the fact that both test *passes* and test *fails* follow the same path in these regions, and thus a class is indistinguishable. The variables in *regions 1* $\rightarrow$ *3* are discrete test vectors which, on first account, do not contain much information. This finding is the main factor in developing a solution that looks for a smaller representative subspace for the process data, and hence employs data reduction principles. In Section 5.1.1, Figure 5.3 shows a parallel-coord plot with upper and lower confidence limits. This can be compared with univariate control charting techniques when process evolution is viewed dynamically (*i.e.* a batch test) and any significant variable contributions are apparent. The importance of the upper and lower limits are to distinguish between normal process operation and abnormal process operation and have the desired effect of determining process variable contribution in the data. Figure 6.6 shows four different fault conditions together with normal process data. The reason for this type of plot is to identify, if possible, variables that account for *fails*. In Figure 6.6 (a), normal process data is plotted with a Sort 42 *fail*. Although the plot is highly populated, variable number 104 exceeds the minimum level (Min) for normal process operation. This may gives an insight into the *fail* status. This

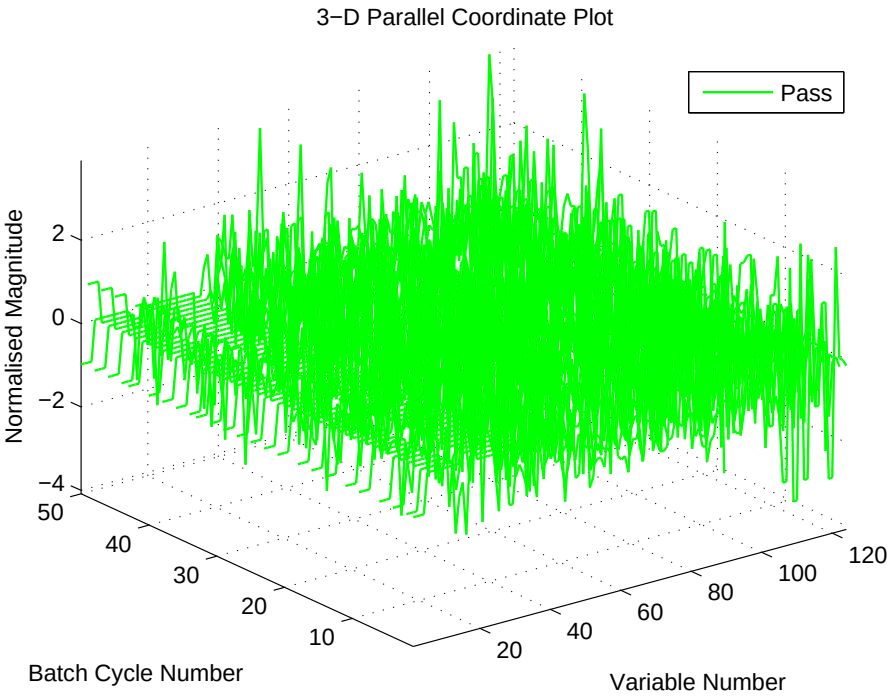
is shown more clearly in Figure 6.7 where the plot is rescaled. Similarly, in Figures 6.6 (b)→(d), the *fail* trends can be interpreted analogously where some are more immediately obvious than others. Figure 6.8 details a *fail* which exceeds both the upper and lower limits.

In Figure 6.9, a response surface plot conveys areas of variation in the data set. The sparse areas represent process stationarity while the undulations, peaks and troughs represent process variation, the magnitude of which is proportional to the shape and severity of the surface. These surface anomalies represent the variables which influence the test outcome.

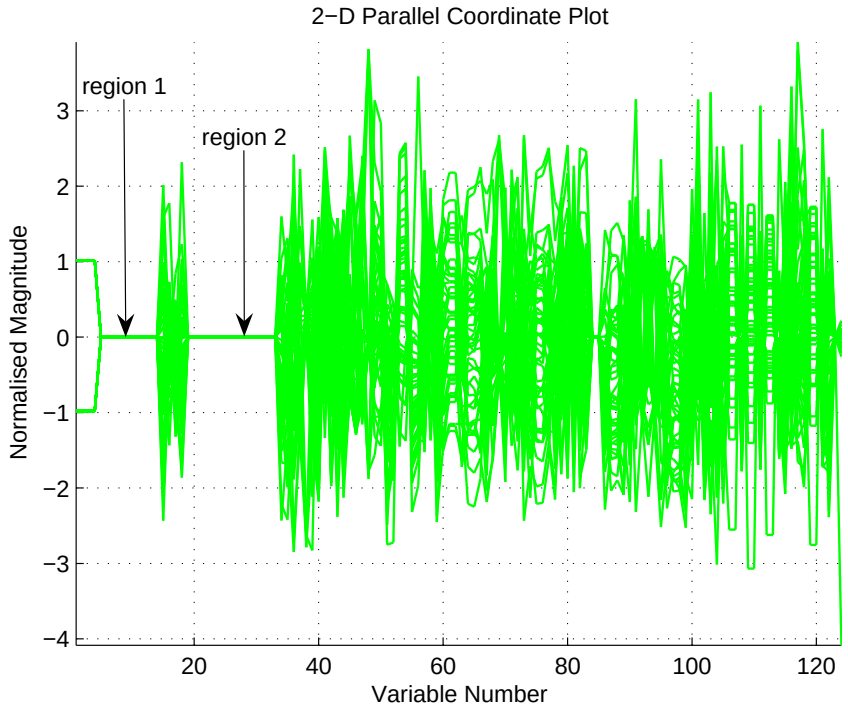
While these data exploration methods can provide an insight into the process dynamics they are not generally useful in determining causation variables. In Multivariate Statistical Process Monitoring (MSPM), one of the most important facets of the analysis is closing the loop between fault detection and fault classification. Therefore, the cause of the fault is as important to detect as the fault itself. Chou et al. (1999) suggest that data visualisation plays an indispensable role in uncovering hidden structures or patterns for information retrieval and parallel-coord plots are indeed important tools in such analysis. One disadvantage of parallel-coord plots is overcrowding when the measurement space becomes too populated. There are many methods to overcome this, such as brushing, opacity plotting, colour plotting, marker plotting and even using higher order star glyphs, Ward et al. (2003) and Chernoff faces, Chernoff (1973) and Chernoff & Rizvi (1975).

6.1.4 Principal Component Analysis

As with all highly automated manufacturing and testing processes, there is a superfluous level of process variables and product characteristics. Therefore, any method that reduces the dimension of the problem domain is a highly applicable and worthwhile investment. PCA is used to determine a systematic pattern of variation in a data set. It is also known as *latent* variable modelling, where the PC score is a non-measurable latent variable that is computed as linear combinations of a set of manifest input variables.

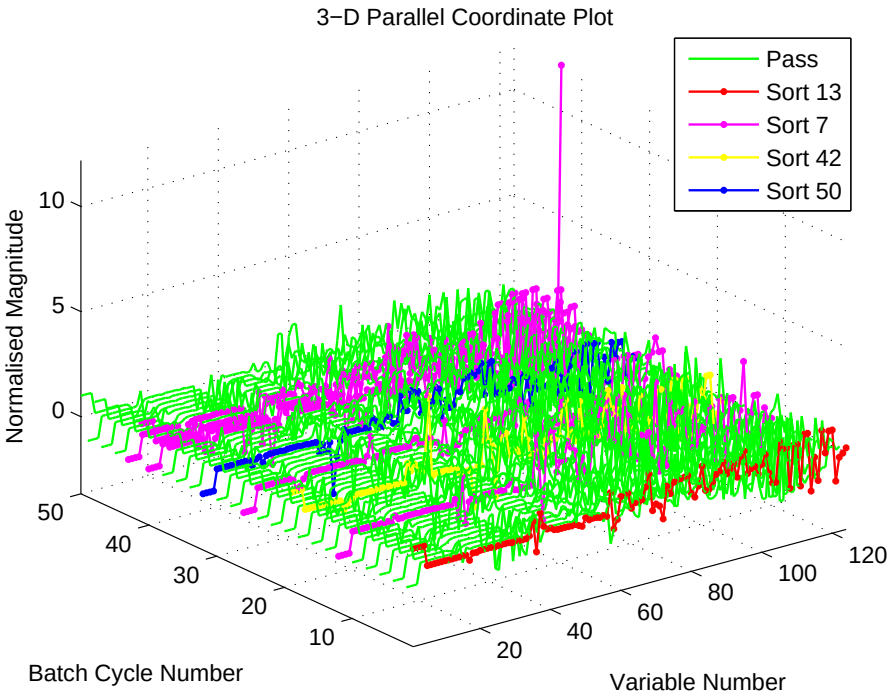


(a)

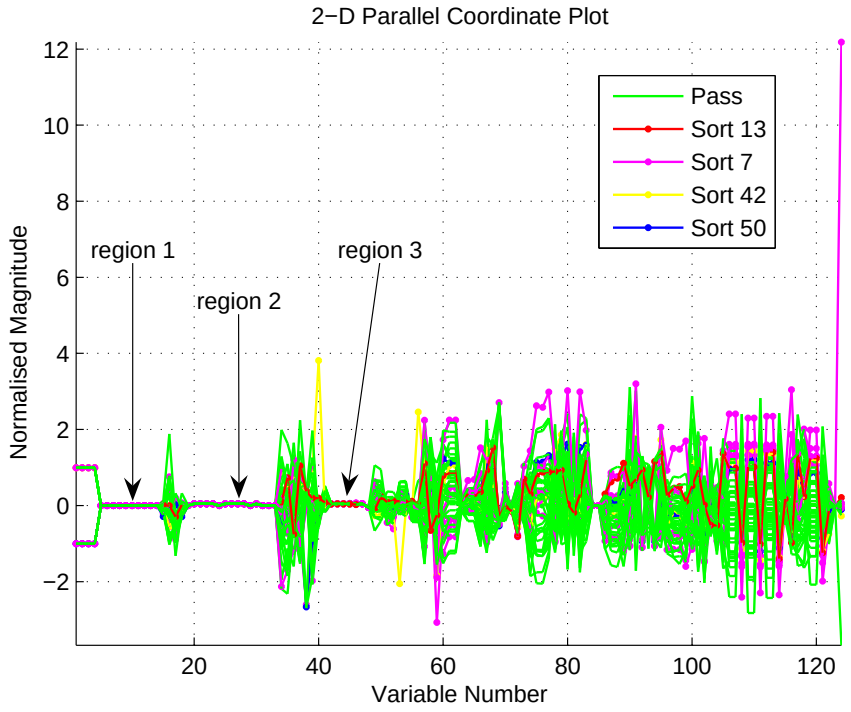


(b)

Figure 6.4. Normal process parallel-coord plot. (a) *A 3-d parallel-coord representation of normal process operation data. The data are known passes.* (b) *X-Z axes of (a) outlining 2-d parallel-coord plot. The stochastic nature of the process can be seen in both subplots.*



(a)



(b)

Figure 6.5. Abnormal process parallel-coord plot. (a) *A 3-d parallel-coord representation of abnormal process operation data. The data is both pass & fail.* (b) *X-Z axes of (a) outlining 2-d parallel-coord plot. 3 stationary regions are identified.*

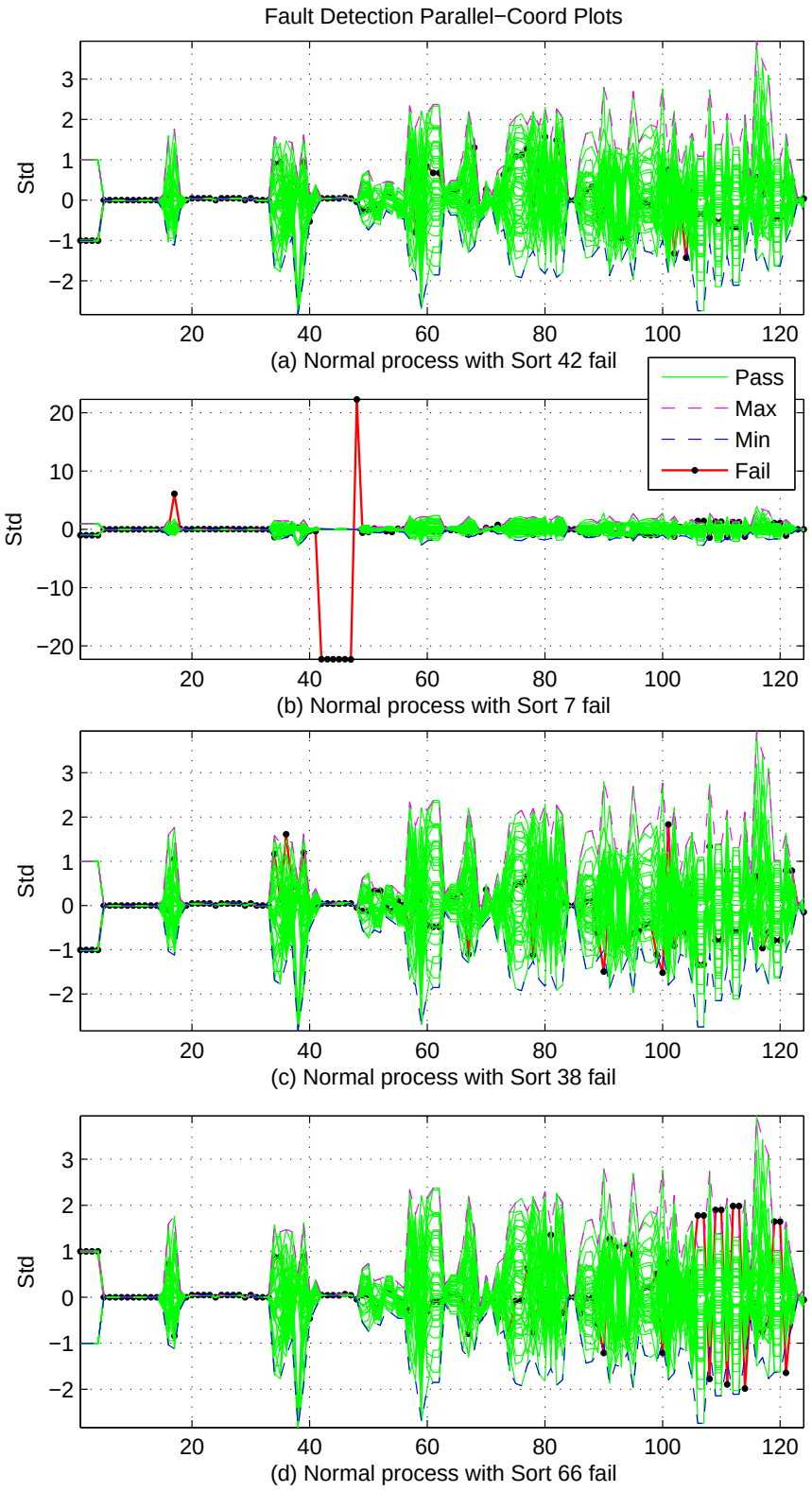


Figure 6.6. Fault detection parallel-coord plots. *Variable number (x-axis) against standardised magnitude (y-axis). 4 fault conditions are shown.*

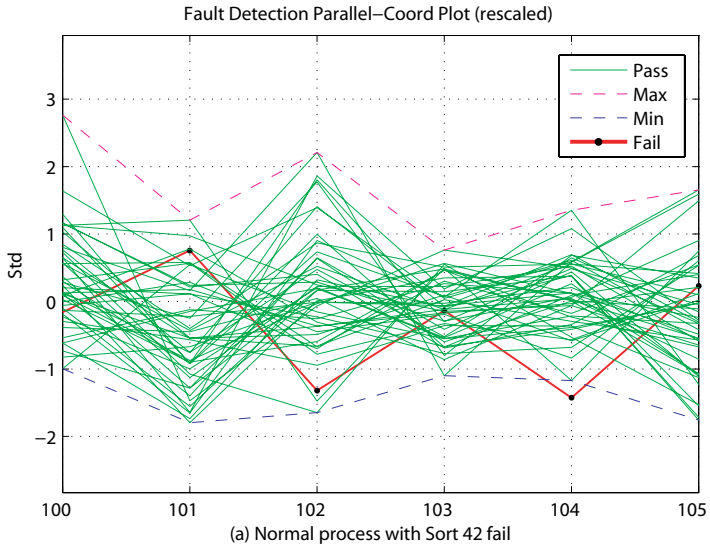


Figure 6.7. Fault detection parallel-coord plot (rescaled). *Variable number 104 exceeds lower limit (Min) level for normal process operation. This rescaled plot is consistent with Figure 6.6 (a).*

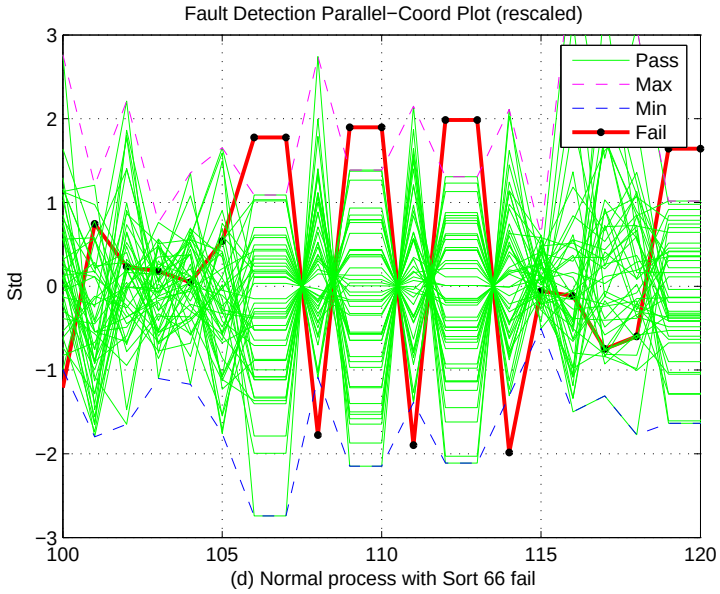


Figure 6.8. Fault detection parallel-coord plot (rescaled). *Variables exceed both upper (Max) and lower (Min) limits for normal process operation. This rescaled plot is consistent with Figure 6.6 (d).*

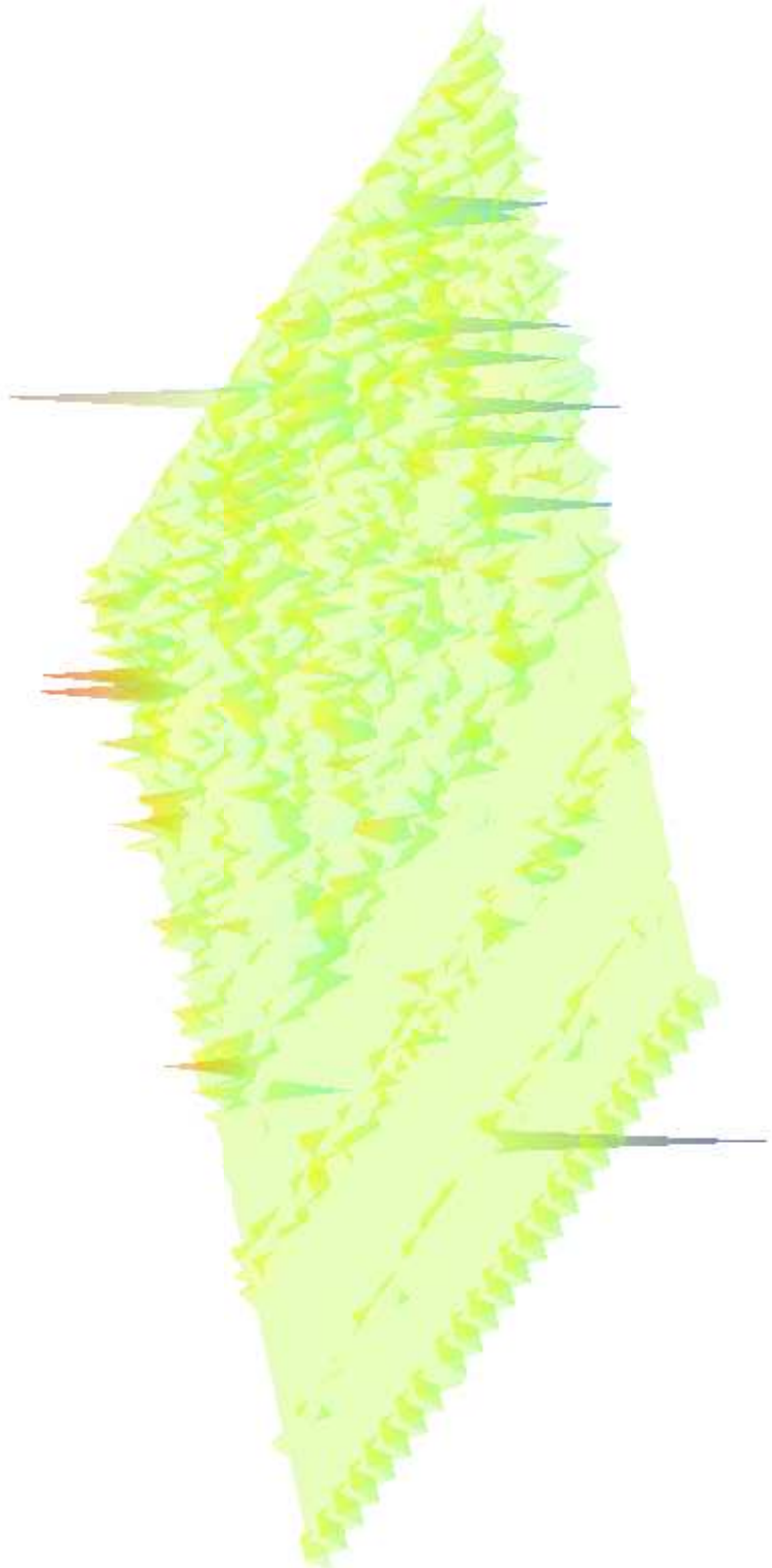


Figure 6.9. Response surface plot with mixed data. *The sparse areas represent data stationarity while the undulations, peaks and troughs represent data variation (magnitude ∝ surface severity).*

In order to classify a process state, or indeed, adequately describe a process using fewer dimensions, it is necessary to build a suitable model which can be used as a reference set for any subsequent data that may be generated. It is imperative to isolate normal process data from a mixture of normal and abnormal historical data, He et al. (2004). Referring to Table 5.1, the data are prepared through pre-processing and partitioning, and a Normal Operating Condition (NOC) model is developed (Section 5.2.3). For the purpose of the discussion, and in keeping with the parallel-coord plots, two process states are identified. Figure 6.13 shows the first three PCs as time-series plots with their respective sample numbers. The `PCA Model Summary` shows each eigenvalue contribution to the model. The decision to retain 8 PCs in the model was motivated by the fact that a reduction in feature space from $\mathbb{R}^{124} \rightarrow \mathbb{R}^{8}$ occurs. These PCs represent 73.14% of the variance in the original data. If the data has been autoscaled to $\mathbf{N} \sim (0, 1)$, the eigenvalues will sum to the number of original variables, and the eigenvalues can be interpreted as the number of original variables *each* PC is worth. In this PCA model, the first PC captures as much variance as 24 original variables, while the second PC captures as much variance as 15 original variables. This suggests a high degree of correlation in the data set, as two PCs capture as much variance as 39 original process variables.

---PCA Model Summary---

Principal Component Number	Eigenvalue of Cov Matrix	% Variance Captured per PC	% Variance Captured Total
1	2.30e+001	23.96	23.96
2	1.45e+001	15.09	39.06
3	1.03e+001	10.76	49.82
4	7.36e+000	7.66	57.49
5	4.97e+000	5.17	62.66
6	3.71e+000	3.87	66.53
7	3.48e+000	3.63	70.16
8	2.86e+000	2.98	73.14

PCA produces linear combinations of the original variables, or *latent variables*,

through a transform equation of

$$Score\ on\ PC1 = w_1x_1 + w_2x_2 + ... + w_dx_d$$

where x is the original data and w is the loading or *weight* of the variables on the score. Figure 6.13 shows three the first three score vectors and respective limits. The scores contain information on how the samples relate to each other and the three component model contains just under 50% of the variance in the original data. The loadings give information about the variables and outline the dominant process variables for each score vector. Space precludes listing the loadings for the PCA model here but the first three PCs are given in Appendix A.

In Figure 6.11 , variables $7-16$ and $19-32$ are not loaded by any components. The reason for this is similar in concept to Figure 6.5 where the original variables have little or no input to the process, and thus are excluded in the model as they contain no information. Appendix B shows a complete breakdown of how the variance is captured by each PC. Figure 6.12 shows both score and loading plots. These can be used to reveal any correlation in the data and also the variables which influence the scores the most. Furthermore, a *bi-plot* may be generated by superimposing the score plot and the loading plots into one Figure. This shows which samples scores are similar and which are not. In Figure 6.12, two clusters of loadings are identified. These clusters suggest that some loadings are similar to one another and hence may influence the scores in the same way. As the loadings represent the variables, one would expect these to be similar also. The clusters are pseudo labelled ($X_1 \rightarrow X_d$) and described in `LOADING Cluster Assignments`.

```
LOADING Cluster Assignments
CLUSTER 1  X108  Gainslope PCSB   CLUSTER 2  X106  OUTPUT 10 PCSB
           X111  Gainslope PCSA              X107  OUTPUT MIN PCSB
           X114  Gainslope CELLB             X109  OUTPUT 10 PCSA
           X121  Gainslope CELLA             X110  OUTPUT MIN PCSA
                                              X112  OUTPUT 10 CELLB
                                              X113  OUTPUT MIN CELLB
                                              X119  OUTPUT 10 CELLA
                                              X120  OUTPUT MIN CELLA
```

When the cluster assignment is cross checked with the process variables, it is confirmed that they are indeed similar measurements. Variables which load most significantly in the first PC (PC1) are those which are located furthest away from the centroid (*x-axis* and *y-axis* intersection) of the data. These two clusters were chosen to illustrate this. Another point to note is they load with opposite signs, which indicates they are anti-correlated. Scores in the lower left quadrant, *-PC1* and *-PC2*, would tend to be dominated by the cluster 1 loadings and likewise scores in the upper right quadrant *+PC1* and *+PC2* would tend to be dominated by cluster 2. This information is not immediately obvious when analysing univariate score and loading plots, and thus reinforces the importance of bivariate and multivariate plotting. Another important contribution to the score and loading plots are the x and y-axis lines indicating zero. Depending on the plot type, these indicate different things. For instance in a PCA model, a zero loading means that the variable does not contribute to the displayed component and a zero score means that the sample shown is close to the mean (centroid) of the data.

Similarly, model visualisation can be extended to $\mathbb{R}^3$ space, with a PC1,PC2 and PC3 score and loading plot. The transition from Figure 6.13 to Figure 6.14 is achieved by incorporating the third PC axis, PC3. This is not fully representative of the PCA model however, as there are an additional five orthogonal axes but it serves as a useful indicator. As this model is a NOC model, one would expect the scores to be quite closely clustered about the centroid (*X-Y-Z intersection*) and this is visible in Figure 6.14. Figure 6.15 shows a slightly more advanced visualisation method where the PC scores and a confidence interval are plotted together. A departure from the NOC centroid indicates that a sample is not described exclusively by the model and therefore is an outlier. An outlier is an observation that deviates so much from the other observations as to arouse suspicion that it was generated from a different mechanism. An outlier can be the onset of abnormal behaviour and it may carry important diagnostic information as to the root cause of the anomaly but before abnormal observations can be singled out, it is necessary to characterise normal process observations accurately.

In order to test the PCA model and display new fault conditions, unseen process data is introduced. This data is standardised with respect to the mean and standard deviation of the NOC model. It is then possible to show the model

result through a NOC ellipsoid plot. This is shown in Figure 6.16. In this figure, 2σ and 3σ confidence ellipses have been calculated from normal process data, *i.e.* data devoid of anomalies or failures. The NOC scores dictating the ellipsoids are shown by the '●' and the new batch scores are shown by '▼'. On inspection, there are a number of '▼' points disassociated with the NOC ellipsoid, suggesting they may have been generated by a different mechanism and therefore are not sampled from the same distribution. The resolve is a method which seeks to classify *new, unseen* data in order to pass judgement on it.

It is however, important to look at the remaining PCs, the so called lower order components, to see if they represent any patterns or trends within the data. The PCA model used to generate the NOC ellipsoid contains eight components, which give $\approx 73\%$ of the variance in the original data. The model gives a reduction in feature space from $\mathbb{R}^{124} \rightarrow \mathbb{R}^{8}$, which in any monitoring environment is a useful result. It would be naive to suggest that the feature space reduction represents entirely, and that discarding $\approx 25\%$ of the variance does not loose any information but one of the strengths of this analysis is decorrelation of original data into linear, orthogonal axes. A supervisory system monitoring the NOC model and subsequent test scores can then be used, with a certain level of confidence, to classify new process data. The model is not predictive in nature, *i.e.* it cannot forecast a result prior to a test but heuristic information can assign a certain expectation level based on historical data.

The multivariate indices of $\mathbf{T^2}$ and $\mathbf{Q}$ are used to monitor the entire PC subspace of the model. Figure 6.17 shows both indices and sample contributions. Sample number 26 arouses suspicion due to the fact that it is so dominant in both indices. Other samples exceed the 95% confidence level but the plot has to be rescaled accordingly to show them. Figure 6.18 shows the rescaled indices which indicate model performance on the test data. Samples in both control charts exceed the confidence intervals indicating a departure from the normal. As the latent variable space of the PCA model has been calculated with *fail* free process data, comparing new data with the normal or *common-cause* variation captured by the model signals any abnormal situations. The $\mathbf{T}^2$ control chart only detects variation in the plane of the first k PCs that is greater than what can be explained by the *common-cause* variation and the $\mathbf{Q}$ represents fluctuations that cannot be accounted for by the PCA model. When an unusual event occurs that results in

a change of the process mean or covariance structure, it will be detected as a high value of this statistic. Section 3.2.4 offers a more detailed review on multivariate indices.

A fail matrix indicating the *sample* number, bin-sort code (in square brackets) and multivariate signalling index ($\mathbf{Q}$ *or* $\mathbf{T}^2$) summarises Figure 6.18.

$$Fails\,(Figure\ 6.18) = \begin{pmatrix} 2 & \rightarrow & [7] & \rightarrow & \mathbf{T^2} \\ 10 & \rightarrow & [7] & \rightarrow & \mathbf{T^2} \\ 12 & \rightarrow & [11] & \rightarrow & \mathbf{T^2}\&\mathbf{Q} \\ 14 & \rightarrow & [11] & \rightarrow & \mathbf{T^2} \\ 18 & \rightarrow & [55] & \rightarrow & \mathbf{T^2}\&\mathbf{Q} \\ 21 & \rightarrow & [50] & \rightarrow & \mathbf{Q} \\ 24 & \rightarrow & [42] & \rightarrow & \mathbf{T^2}\&\mathbf{Q} \\ 25 & \rightarrow & [\texttt{FN}] & \rightarrow & \mathbf{T^2} \\ 26 & \rightarrow & [42] & \rightarrow & \mathbf{Q} \\ 43 & \rightarrow & [\texttt{FN}] & \rightarrow & \mathbf{T^2} \end{pmatrix}$$

Two entries in the fail matrix are `False Negatives (FN)`, *i.e. when the outcome is incorrectly predicted by the model to be a fail*, giving an indication that the model is sensitive to the loadings that generate those particular scores. There are no `False Positives (FP)` in this model, *i.e. when the outcome is incorrectly predicted by the model as a pass*, which of the two is more important to prevent. A `FN` can be considered a false alarm where a `FP` is a failed alarm or *miss*. In Figure 6.18 , both of the `FN` entries are signalled by the $\mathbf{T^2}$ index. In practice, violations of the $\mathbf{T^2}$ and $\mathbf{Q}$ limits occur for different reasons. A $\mathbf{T^2}$ fault indicates that the process has gone outside the usual range of operation but in a direction of variation that is common to the process, *i.e.* there may be too much or too little of a particular variable normally present in the model. A $\mathbf{Q}$ fault indicates that the process has gone in an entirely different direction and something new, not included in the model, has happened. Traditional MSPC philosophies calculate and observe both indices as they compliment each other.

Contribution plots are diagnostic tools used to complete the loop between fault detection and fault identification. They are individual cases in which the loadings of a sample score can be isolated and inspected. This diagnostic information is extracted from the underlying PCA model at the point where the

abnormal event or *fail* has been detected. Figure 6.19 and Figure 6.20 outline $\mathbf{T^2}$ and $\mathbf{Q}$ contribution plots for different samples from Figure 6.18. In Figure 6.19, two contribution plots are shown. Figure 6.19 (a) shows a $\mathbf{T^2}$ contribution plot for a normal process sample score and in comparison, Figure 6.19 (b) shows a $\mathbf{T^2}$ contribution plot for an abnormal process sample score. Similarly, $\mathbf{Q}$ contribution plots for two process states are shown in Figure 6.20 (a) and (b). The green and red bar charts represent *pass* and *fail* for the respective indices. The dominant variables are apparent in both Figures, and these give an indication of the process variables that most influence the result on that score. This fault finding methodology is normally applied to each score that breaches either of the indices.

When a *fail* is diagnosed in this manner, the root cause can be attributed to either a single variable or a number of variables. Translating this information into a successful control strategy is achieved either through actual process interaction with setpoint adjustment or by discrete recommendations made to process operatives with the root cause of the *fail*.

6.1.5 Decision Tree Classification

Decision tree classification of process data is performed differently. Partitioned data, *i.e.* subsets of *pass* and *fail*, are important in classification routines that seek to compare one sample with another. This is the philosophy of unsupervised learning techniques. However, using the same partitioned data in supervised classification techniques will result in reduced predictive ability. The reason for this is due to the fact the a supervised model relies on good training data to make an accurate prediction.

A generic method of decision tree induction is to build a model based on a set of training data known to consist of both normal and abnormal process operation conditions. This ensures that the model has an accurate description of the process running normally and more importantly, a description of any *fails* that occur. Section 5.4.2 describes the evaluation of decision trees that have been trained on mixed signal data.

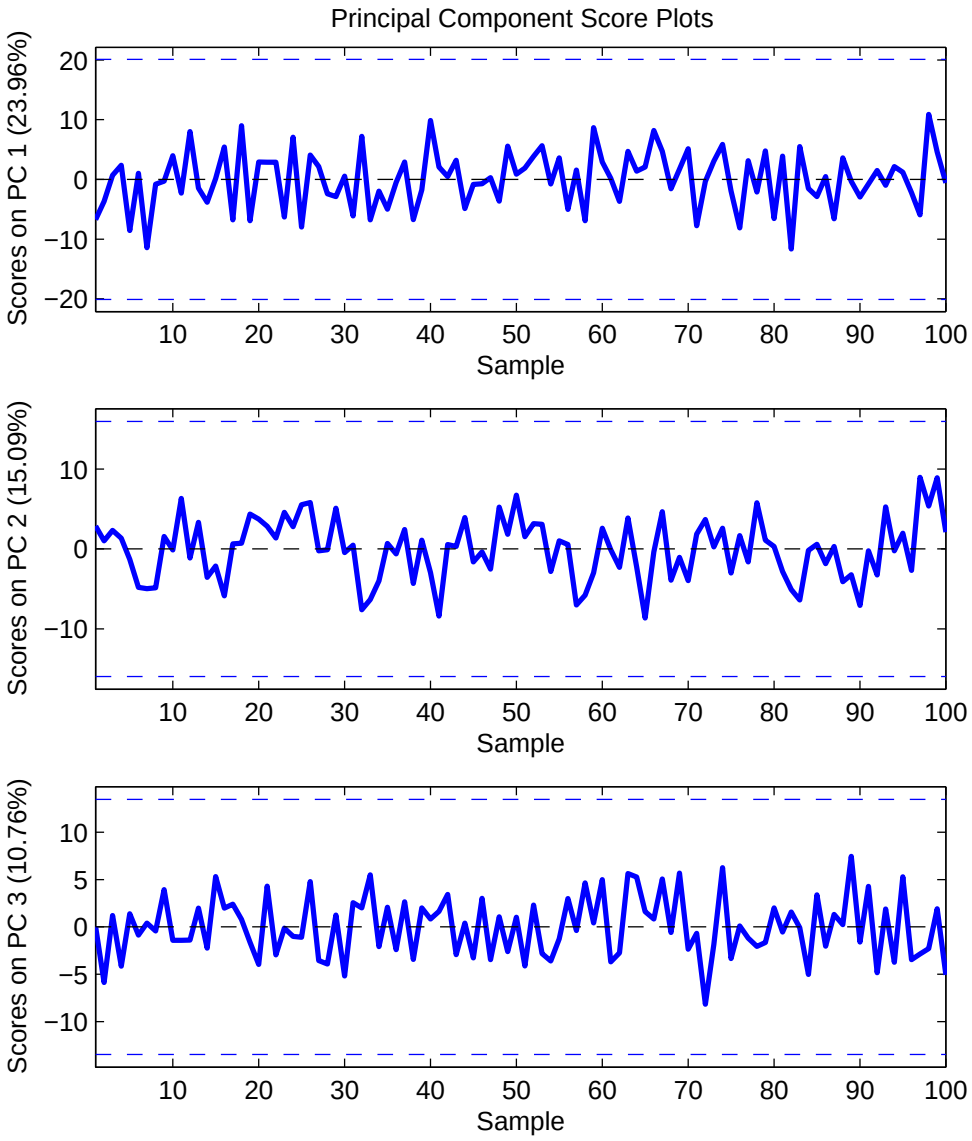


Figure 6.10. PC score plot. *The first 3 PC scores plotted individually with respective 95% control limits. The PC1-PC2-PC3 space describes 49.82% variation in the data.*

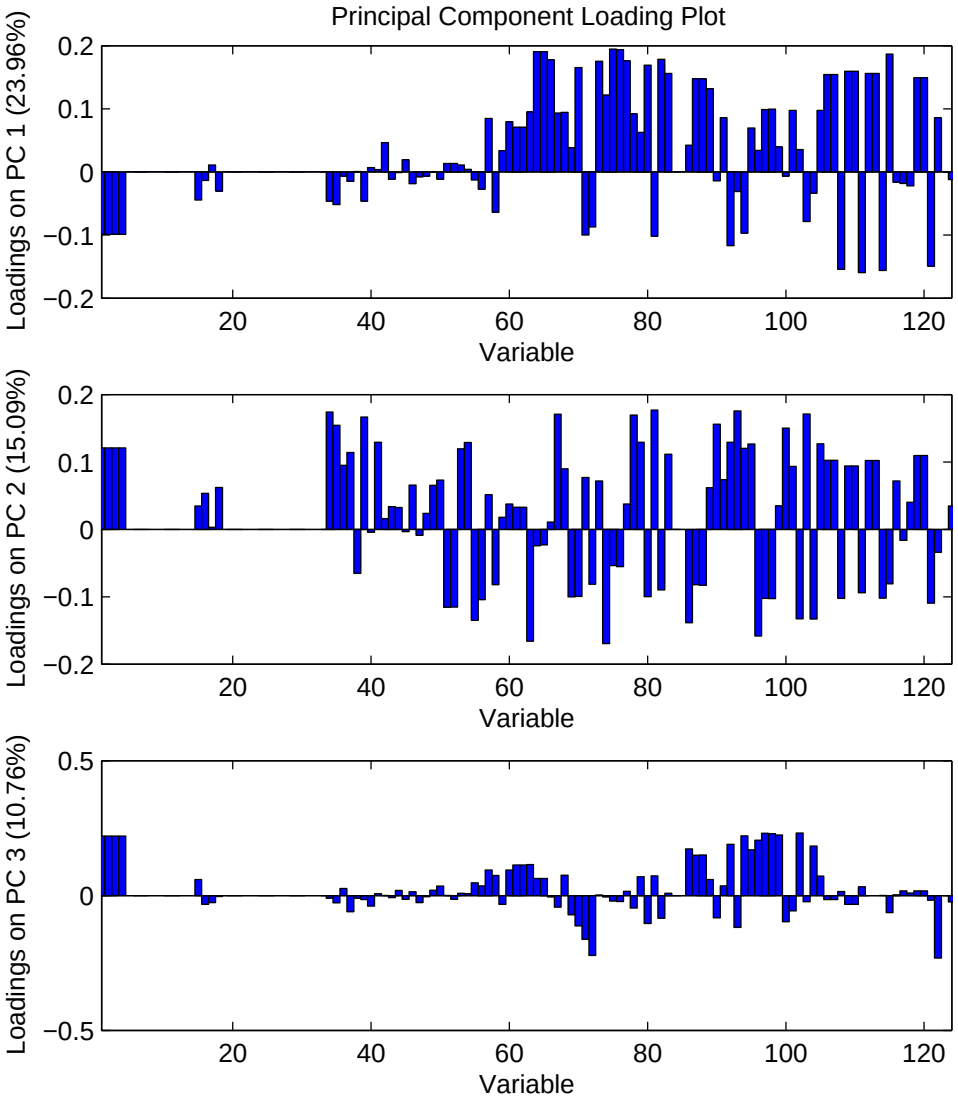
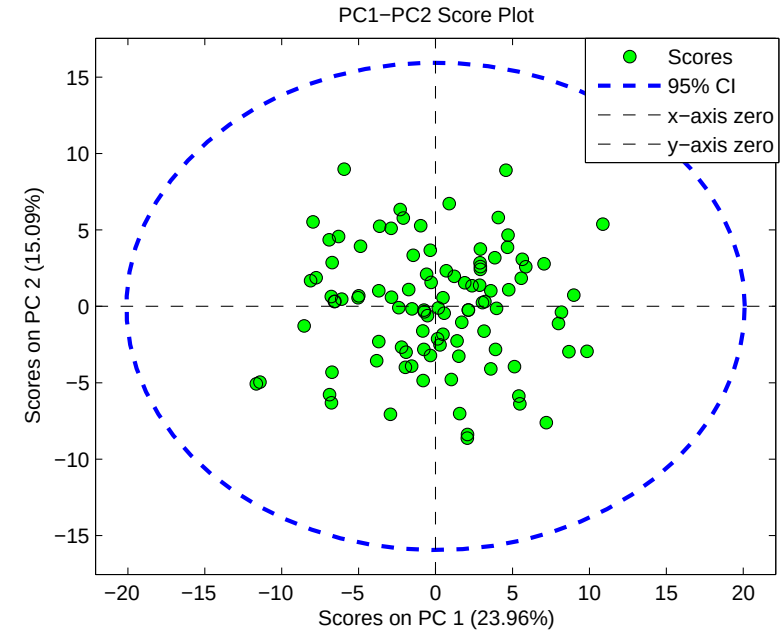
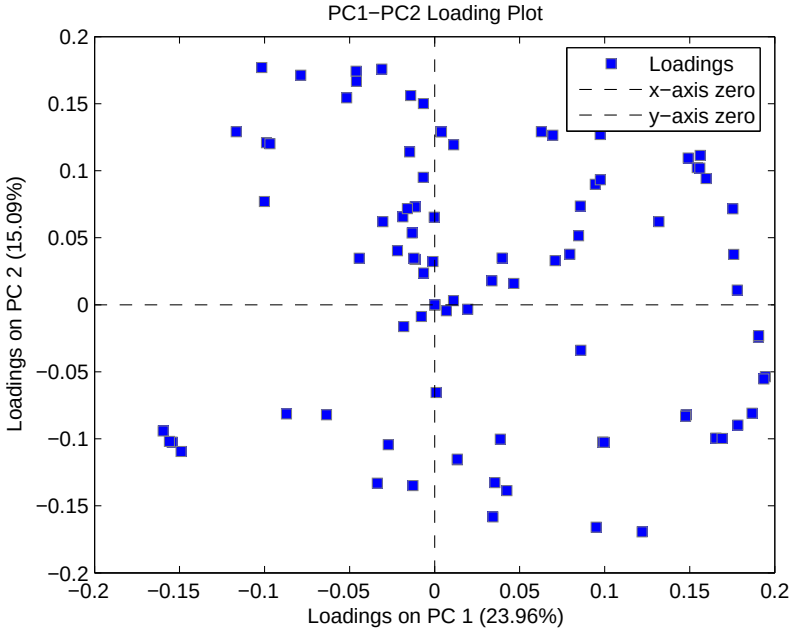


Figure 6.11. PC loading plot. *Contributions of the process variables to PCA model. The dominant loadings show the process variables that are responsible for the PC scores.*



(a)



(b)

Figure 6.12. PC1-PC2 score & loading plots. (a) *PC1-PC2 score plot with 95% confidence interval.* (b) *PC1-PC2 loading plot .*

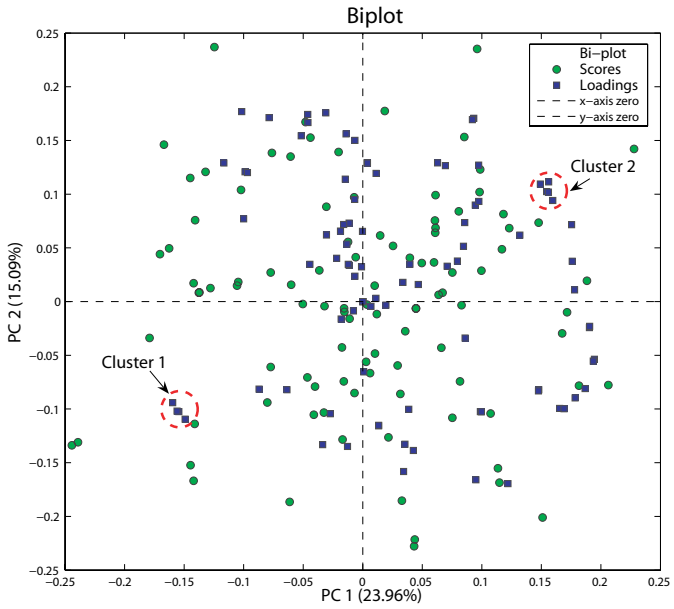


Figure 6.13. Biplot of PC scores & loadings. *Scores and loadings plotted simultaneously showing the loadings which influence the scores. 2 loading clusters are identified and their constituent variables described.*

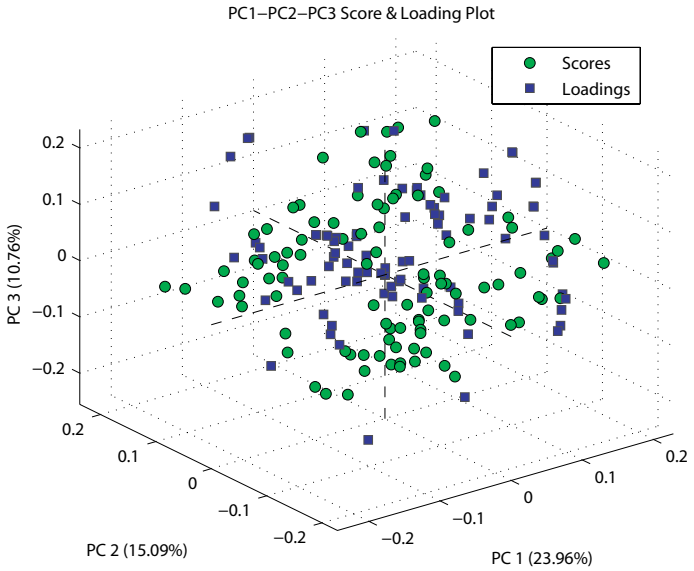


Figure 6.14. PC1-PC2-PC3 score & loadings plot. *Scores and loadings plotted simultaneously for the first 3 PCs. Again, loadings which influence the scores are clearly visible.*

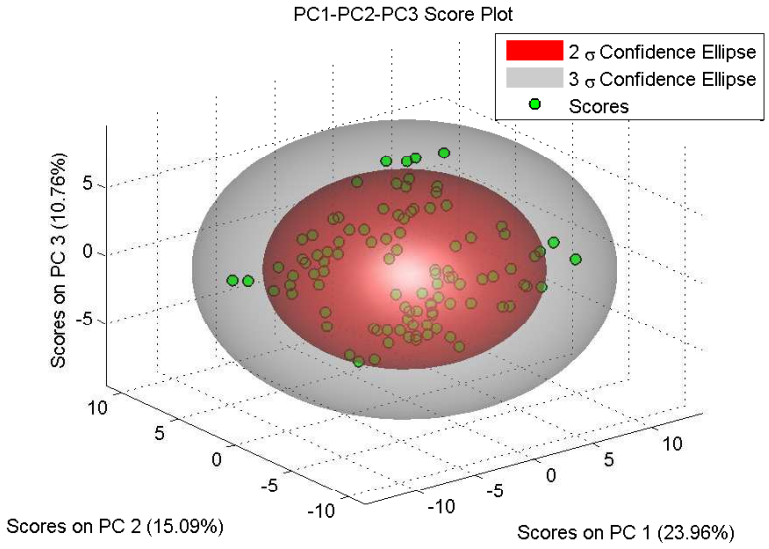


Figure 6.15. PC1-PC2-PC3 NOC ellipsoid. *NOC scores with 2σ (95%) and 3σ (99%) confidence ellipses. As NOC model is derived from good process data, all scores fall with in ellipses.*

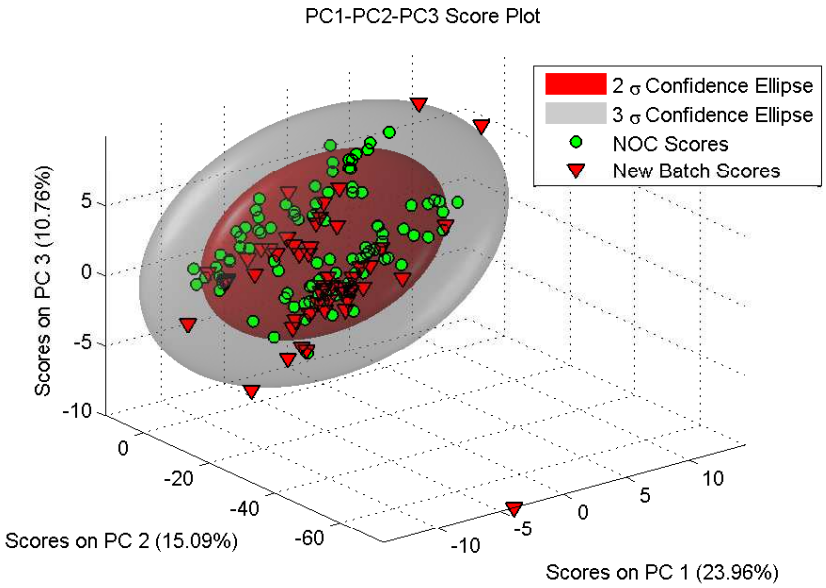


Figure 6.16. PC1-PC2-PC3 NOC ellipsoid with new batch scores. *A certain number of the new scores, '▼', are located away from the 3σ (99%) confidence ellipse. This suggests they are significantly different from the NOC model and are not sampled from the same distribution. This method of representation can be used to detect an anomaly.*

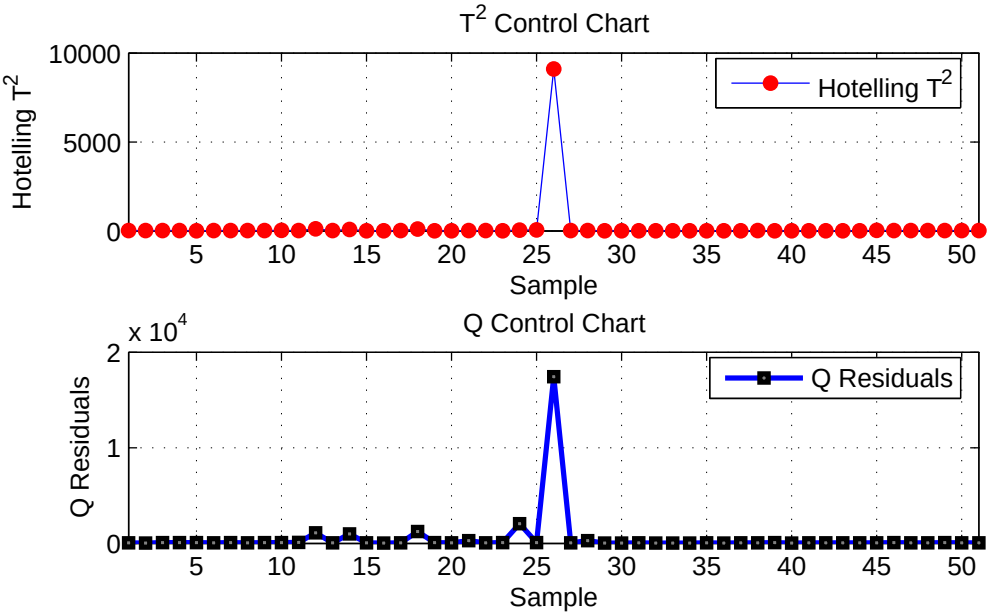


Figure 6.17. Multivariate detection indices. $\mathbf{T^2}$ *and* $\mathbf{Q}$ *chart for the PCA model. Sample number 26 has a significant impact on both indices, and is a suspected abnormal result or failure.*

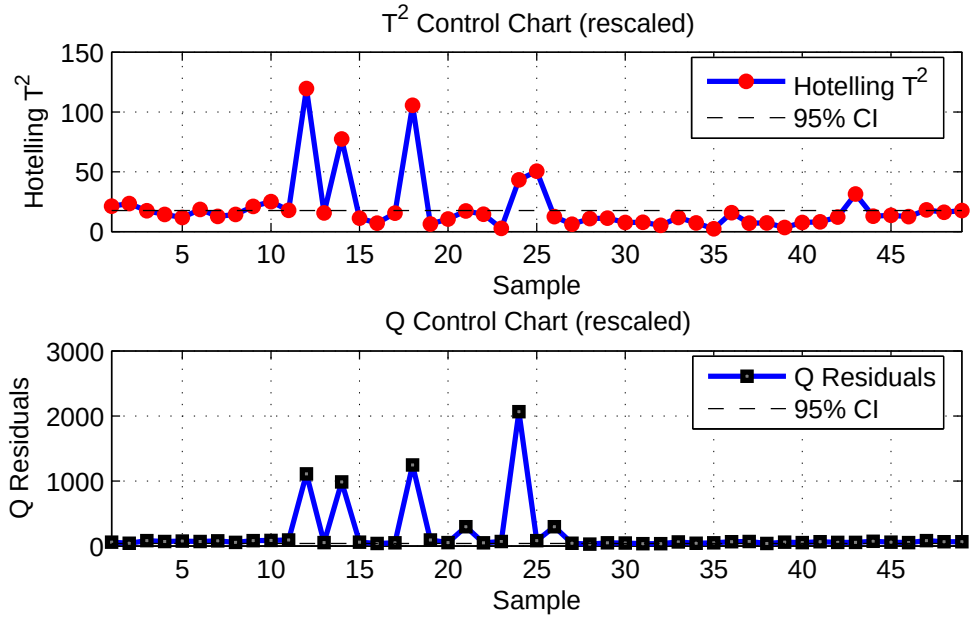
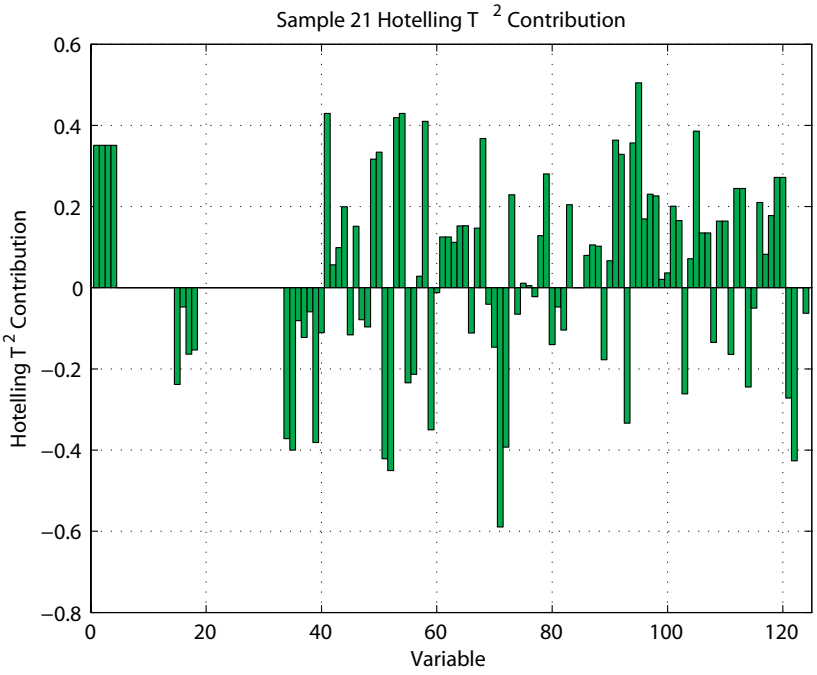
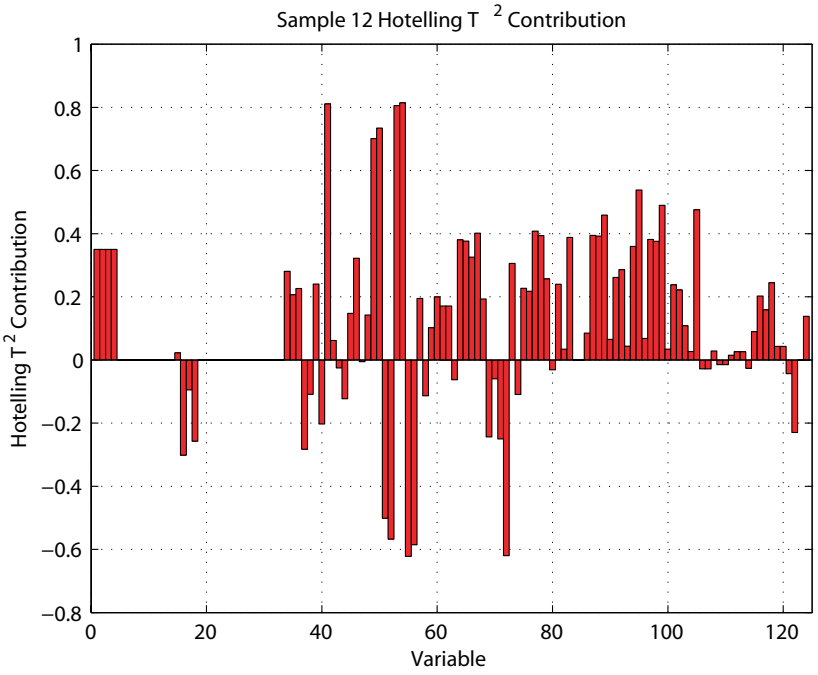


Figure 6.18. Multivariate detection indices (rescaled). *Rescaled indices to show model performance on test data. Samples on both the* $\mathbf{T^2}$ *and* $\mathbf{Q}$ *chart are seen to exceed the 95% CI.*

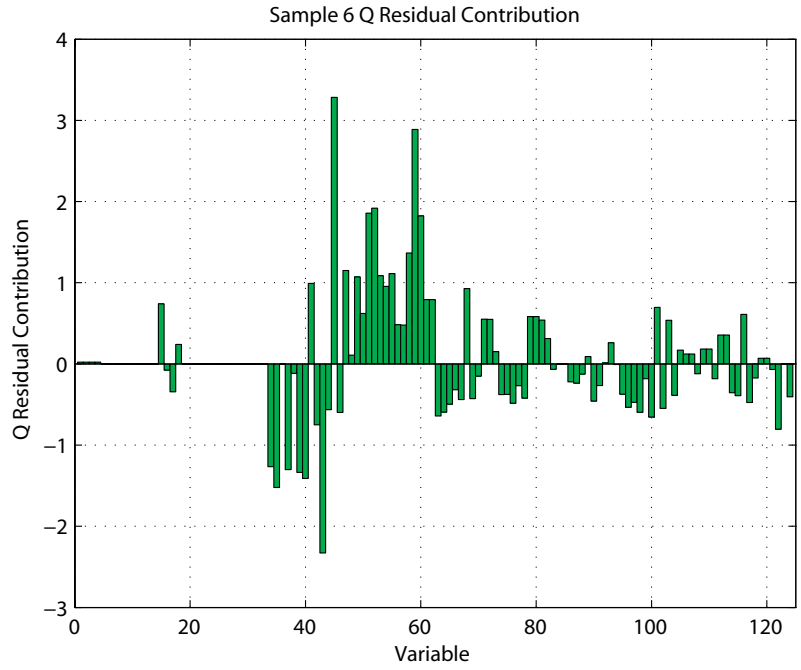


(a)

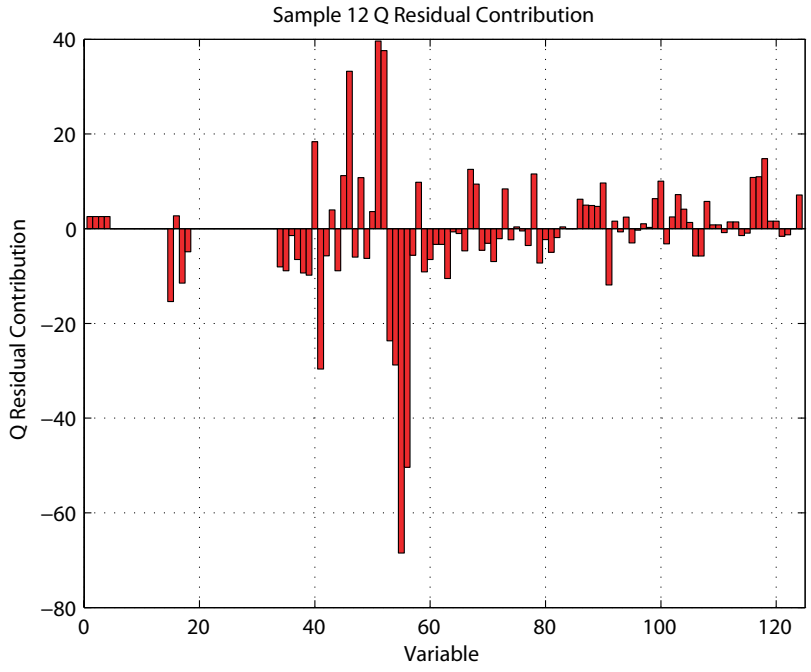


(b)

Figure 6.19. $\mathbf{T^2}$ contribution plots for samples 21 & 12. (a) $\mathbf{T^2}$ *contribution to sample 21, a pass NOC score* and (b) $\mathbf{T^2}$ *contribution to sample 12, a fail batch score. The green and red bar charts represent pass and fail. The dominant variables are apparent.*



(a)



(b)

Figure 6.20. **Q** contribution plots for sample 6 & 12. (a) **Q** *contribution to sample 6, a pass NOC score* and (b) **Q** *contribution to sample 12, a fail batch score. The green and red bar charts represent pass and fail. The dominant variables are apparent.*

6.2 Chapter summary

In this chapter, it has been shown that the application of exploratory data analysis methods in semiconductor batch test data can effectively differentiate between normal process operation and abnormal process operation. Fault detection with parallel coordinate monitoring plots is achieved and the identification of the process variables responsible is also shown. Parallel-coords gives an indication of the process variables at fault and corresponding magnitude. The method of PCA gives a reduced subspace in which the process is described. New batch data can be classified by the model based on score location and variable contribution plots. A score loading bi-plot details the exact process variables that influence a sample score. Variables clustered together are seen to influence the scores in the same manner. It is seen from Figure 6.16 that departure from the centroid of the NOC scores is an indication of an abnormal process condition or *fail*.

Independent univariate monitoring of the original process variables will not sufficiently capture and diagnose abnormal operating conditions. This is due to the fact that most of the time variables are not independent of one another, and in fact, the quality index is a multivariate property that can not be explained by one factor alone.

Multivariate monitoring indices provide information on PCA model fit and residual contributions. These indices provide a summary of the process condition in a single control chart which allows easy monitoring and fault detection and classification.

Chapter 7

Conclusions

Contents

"If all economists were laid end to end, they would not reach a conclusion"

(George Bernard Shaw (1856-1950)

7.1 Main Conclusions

This work investigates the application of multivariate statistical methods for fault detection and classification in a semiconductor device testing process. Current approaches to process monitoring can be grouped into three categories

1. Analytical model based methods
2. Knowledge based and expert systems
3. Historical data driven methods

Analytical methods are theoretically elegant but are limited in a real world senario. Knowledge based and expert systems for fault classification are relatively straight forward to develop but have the disadvantage of being time consuming and sensitive to training information. Historical data driven methods are the

most widely applied to industrial processes. This thesis explores different multivariate methods based on historical data and is separated into unsupervised (Chapter 3) and supervised (Chapter 4) analytical techniques.

The purpose of implementing a quality initiative is to improve the quality of a *product* by reducing the variability in a *process*. In the context of this thesis, only the process is controllable so the emphasis is on a reduction in *variability* where the product is actually used to test the process. This work demonstrates the difficulty in separating product and process variation. As presented in Chapter 6, one method is to look at process variable contributions in order to see any underlying trends. It is not sufficient to monitor a high level quality index such as yield in total isolation from a process. In one particular example in this thesis, 125 process variables are tested for every semiconductor device. Although the dichotomous yield index summarises the outcome in the form of *pass/fail*, much more information is available from the process variables themselves.

The nature of the process data in this thesis can best be described as *mixed-mode*, *i.e.* composed of both analogue and digital test variables. This thesis presents a methodology of pre-processing mixed-mode data as an input to a control strategy.

Exploratory methods are applied to the data in Chapter 3. The inherent variability in the test process can be seen from the Parallel Coordinate Analysis (parallel-coord) plots. The development of a 3-dimensional visualisation technique based on parallel-coords is shown as a process monitoring tool which portrays the process through the variables.

The use of Principal Component Analysis (PCA) results in a reduced subspace or internal structure that is used to describe the process in fewer dimensions. PCA is successfully applied to high dimension semiconductor batch test data which results in a reduction of monitored variables and the ability to diagnose abnormal situations or *fails*. Multivariate monitoring indices such as the $\mathbf{Q}$ and Hotelling's $\mathbf{T^2}$ statistics provide fault detection capability. Variable contribution plots provide diagnostic information as to the root cause of the initial problem.

The use of decision trees with the high volume, mixed-mode semiconductor batch test data is discussed in Chapter 4. The results show supervised learning methods are useful in dealing with this data type for the purpose of classification. Typical performance on unseen data is presented in the form of a confusion

matrix. Non-traditional exploratory methods such as decision tree induction are very sensitive to the training data used and success is a function of data type, attribute selection, quantity of data, induction method and class label distinction (*i.e.* good and bad process description).

This thesis provides a methodology that can be extended to any data rich environment and it demonstrates the importance of multivariate methods in fault detection and classification. The techniques have been described in the context of semiconductor device batch testing but they are widely applicable and the solution approaches suggested are very adaptable. This thesis suggests a data handling strategy that is sufficient for capturing the required information but allows for a reduction in the number of monitored variables. The original hypothesis of the thesis was to gain a greater understanding of the process through analytical methods described in the preceding Chapters.

7.2 Future Work

This thesis outlines a strategy which indicates when the process is performing abnormally and through various multivariate techniques, allows for diagnosis and classification of the problem. The semiconductor test process does not allow for set point adjustment in any of the process variables. Therefore, the application of statistical methods to the process is at a supervisory level instead of an interaction level. Run to run control (R2R), allows actual tester interaction where self diagnostic models duly supervise and alter any process variables at fault for process degradation.

An important factor in the success of a monitoring scheme is to have consistency in the process. In the context of semiconductor batch testing, this requires similar batches be tested on similar platforms with minimal set up and transition. Although batch scheduling is beyond the scope of this thesis, as it is dictated by economics and global market demand, removing any unnecessary variation would benefit the process.

This thesis presents a selection of multivariate methods that have been applied to a specific problem domain. These approaches are widely applicable to other areas and it is of interest to the Author to apply these methods to stock market trading data for the purpose of fault detection and outlier detection.

Bibliography

Abbott, E. A. (1884). *Flatland, A Romance of Many Dimensions*. Second edition.

Alpaydin, E. (2004). *Introduction to Machine Learning*. The MIT Press.

Anderson, T. W. (1963). Asymptotic theory for principal component analysis. *The Annals of Mathematical Statistics*, 34.

Arunajadai, S. G., Uder, S. J., Stone, R. B., & Tumer, I. Y. (2004). Failure mode identification through clustering analysis. *Quality and Reliability Engineering International*, 20, 511–526.

Asgharian, H. & Hansson, B. (2003). The explanatory role of factor portfolios for industries exposed to foreign competition: evidence from the swedish stock market. *Journal of International Financial Markets, Institutions and Money*, 13, 325–353.

Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton Universtiy Press.

Bhatia, S. K. & Deogun, J. S. (1998). Conceptual clustering in information retrieval. *IEEE Transactions on Systems Manufacturing and Cybernetics - Part B*, 28(3), 427–436.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.

Buzzell, R. & Gale, B. (1987). *The PIMS Principles: Linking Strategy To Performance*. The Free Press.

Carpineto, C. & Romano, G. (1996). A lattice conceptual clustering system and its application to browsing retrieval. *Journal of Machine Learning*, 24(2), 95–122.

Caulcutt, R. (1995). *Achieving Quality Improvement*. Chapman & Hall, 1st edition.

Chen, Q., Kruger, U., Meronk, M., & leung, A. Y. T. (2004). Synthesis of $\mathbf{T^2}$ and $\mathbf{Q}$ statistics for process monitoring. *Control Engineering Practice*, 12(8), 745–755.

Chernoff, H. (1973). Using faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68, 361–368.

Chernoff, H. & Rizvi, M. H. (1975). Effect on classification error or random permutations of features in representing multivariate data by faces. *Journal of the American Statistical Association*, 70, 757–765.

Chou, S.-Y., Lin, S.-W., & Yeh, C.-S. (1999). Cluster identification with parallel coordinates. *Pattern Recognition Letters*, 20, 565–572.

Crosby, P. B. (1979). *Quality Is Free. The art of making quality certain*. Mentor.

Deming, W. E. (1993). *The New Economics: for Industry, Government, Education*. MIT Press, 2nd edition.

Dooley, K. J., Anderson, J. C., & Liu, X. (2000). Process quality knowledge bases. *Journal of Quality Management*, 4(2), 207–224.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery. *Advances in Knowledge Discovery and Data Mining*, (pp. 37–54).

Feigenbaum, A. V. (1991). *Total Quality Control*. McGraw-Hill, 3rd edition.

Fielding, A. (1999). *Ecological Applications of Machine Learning Methods*. Kluwer Academic.

Frank, E., Wang, Y., Inglis, S., Homes, G., & Witten, I. H. (1997). Using model trees for classification. *Machine Learning*, 32(1), 63–76.

Fuchs, C. & Kenett, R. S. (1998). *Multivariate Quality Control*. Marcel Dekker, Inc.

Gallagher, N., Wise, B., Butler, S., White, D., & Barna, G. (1997a). Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process: Impact of measurement selection and data treatment on sensitivity.

Gallagher, N., Wise, B., Butler, S., White, D., & Barna, G. (1997b). Development and benchmarking of multivariate statistical process control tools for a semiconductor etch process: Improving robustness through model updating.

Garvin, D. A. (1987). Competing on the eight dimensions of quality. *Harvard Business Review*, 65(6), 101–109.

Goodlin, B. E., Boning, D. S., Sawin, H. H., & Wise, B. M. (2002). Simultaneous fault detection and classification for semiconductor manufacturing tools. In *International Symposium on Plasma Processing XIV* Philadelphia, PA: 201st Meeting of the Electrochemical Society.

Hartigan, J. (1984). *Clustering Algorithms*. Wiley.

He, Q. P., Wang, J., & Qin, S. J. (2004). *A New Fault Diagnosis Method Using Fault Directions in Fisher Discriminant Analysis*. Technical report TWMCC-2004-05, Texas-Wisconsin Modelling and Control Consortium.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11, 63–91.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(24), 417–441.

Hunt, E. B., Marlin, J., & Stone, P. J. (1966). *Experiments in Induction*. Academic Press.

Hunter, J. S. (1986). Exponentially weighted moving average. *The Journal of Quality Technology*, 18, 203–210.

Inselberg, A. (1981). *N-Dimensional Graphics, Part I - Lines and Hyperplanes*. Technical report, IBM LA Scientific Centre.

Inselberg, A. (1997). Multidimensional detective. In *INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)* (pp. 100). Washington, DC, USA: IEEE Computer Society.

Jackson, J. E. (1991). *A User's Guide to Principal Components*. Wiley.

Jackson, J. E. & Mudholkar, G. S. (1979). Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3), 341–349.

Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer-Verlag.

Judd, D., McKinley, P. K., & Jain, A. K. (1998). Large-scale parallel data clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), 871–876.

Juran, J. M. (1988). *Juran on planning for Quality*. The Free Press.

Kain, A. J., Murty, M., & Flynn, P. J. (1999). Data clustering - a review. *ACM Computing Surveys*, 31(3), 264–323.

Kourti, T. (2002). Process analysis and abnormal situation detection. from theory to practice. *IEEE Control Systems Magazine*, (pp. 10–21).

Kourti, T., Lee, J., & MacGregor, J. F. (1996). Experiences with industrial applications of projection methods for multivariate statistical process control. *Computers and Chemical Engineering*, 20, 745–750.

Kourti, T. & MacGregor, J. F. (1995). Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28, 3–21.

Kourti, T. & MacGregor, J. F. (1996). Multivariate spc methods for process and product monitoring. *Journal of Quality Technology*, 28(4), 409–427.

Krzanowski, W. J. (2000). *Principles of Multivariate Analysis: a User's Perspective*. Oxford University Press.

Krzanowski, W. J. (2002). Orthogonal components for grouped data: Review and applications. *Statistics in Transition*, 5(5), 759–777.

Ku, W., Storer, R. H., & Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal components analysis. *Chemometrics and Intelligent Laboratory Systems*, 30, 179–196.

Lane, S., Martin, E. B., Kooijmans, R., & Morris, A. J. (2001). Performance monitoring of a multi-product semi-batch process. *Journal of Process Control*, 11, 1–11.

Lane, S., Martin, E. B., Morris, A. J., & Gower, P. (2003). Application of exponentially weighted principal component analysis for the monitoring of a polymer film manufacturing process. *Transactions of the Institute of Measurement and Control*, 1, 17–35.

Leung, D. (2002). An integration mechanism for multivariate knowledge-based fault diagnosis. *Journal of Process Control*, 12, 15–26.

Loh, W.-Y. & Shin, Y.-S. (1997). Split selection methods for classification trees. *Statistica Sinica*, 7, 815–840.

MacGregor, J. F. & Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3), 403–414.

MacGregor, J. F., Yu, H., Muñoz, S. G., & Flores-Cerrillo, J. (2005). Data-based latent variable methods for process analysis, monitoring and control. *Computers and Chemical Engineering*, 29, 1217–1223.

Maesschalck, R. D., Jouan-Rimbaud, D., & Massart, D. L. (2000). The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50, 1–18.

Mantaras, R. L. D. (1991). A distance based attribute selection measure for decision tree induction. *Machine Learning*, 6, 81–92.

Marlin, T. E. (2000). *Process Control -Designing Processes and Control Systems for Dynamic Performance*. McGraw hill, 2nd edition edition.

Martens, H. & Martens, M. (2001). *Multivariate Analysis of Quality. An Introduction*. Wiley.

Martin, E. B., Kumar, S., & Morris, A. J. (2002). Detection of process model changes in pca based performance monitoring. In *Proceedings of the American Control Conference* (pp. 2719–2724).

Martin, E. B. & Morris, A. J. (2002). Enhanced bio-manufacturing through advanced multivariate statistical technologies. *Journal of Biotechnology*, 99, 223–235.

Martin, E. B., Morris, A. J., & Kiparissides, C. (1999). Manufacturing performance enhancement through multivariate statistical process control. *Annual Review in Control*, 23(1), 35–44.

Mason, R. L., Chou, Y.-M., & Young, J. C. (2001). Applying hotelling's $\mathbf{T^2}$ statistic to batch processes. *Journal of Quality Technology*, 33(4), 466–478.

Misterek, S. A., Anderson, J. C., & Dooley, K. J. (1990). The strategic nature of process quality. *Decision Sciences Journal*.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Montgomery, D. C. (2001). *Introduction To Statistical Quality Control*. Wiley, 4th edition.

Montgomery, D. C. & Runger, G. C. (2003). *Applied Statistics and Probability for Engineers*. Wiley, 3rd edition.

Murthy, S. K. (1998). Automatic construction of decision trees from data: a multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2, 345–389.

Myles, A., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modelling. *Journal of Chemometrics*, 18, 275–285.

Nevill-Manning, C., Holmes, G., & Witten, I. (1995). Development of holte's 1r classifier.

Noorossana, R., Farrokhi, M., & Saghaei, A. (2003). Using neural networks to classify out-of-control signals in autocorrelated processes. *Quality and Reliability Engineering International*, 19, 493–504.

Oakland, J. S. (1999). *Statistical Process Control*. Butterworth Heinemann, 4th edition.

Penza, M., Cassano, G., Tortorella, F., & Zaccaria, G. (2001). Classification of food, beverages and perfumes by wo_3 thin-film sensor array and pattern recognition techniques. *Sensors and Actuators B*, 73, 76–87.

Piatetsky-Shapiro, G. (1991). Knowledge discovery in real databases: A report on the ijcai-89 workshop. *Communications of the Association for Computing Machinery*, 11(5), 68–70.

Pravdova, V., Boucon, C., de Jong, S., Walczak, B., & Massart, D. L. (2002). Three-way principal component analysis applied to food analysis: an example. *Analytica Chimica Acta*, 462, 133–148.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.

Quinlan, J. R. (1993). *C4.5 Programs for Machine Learning*. Morgan Kaufmann.

Raich, A. & Çinar, A. (1997). Diagnosis of process disturbances by statistical distance and angle measures. *Computers and Chemical Engineering*, 21(6), 661–673.

Ritter, H., Martinez, T., & Schulten, K. (1992). *Neural Computation and Self-Organizing Maps -An Introduction*. Addison-Wesley.

Roberts, S. W. (1959). Control chart tests based on geometric moving averages. *Technometrics*, 1, 239–250.

Sachs, E., Hu, A., & Ingolfsson, A. (1995). Run by run process control: Combining spc and feedback control. *IEEE Transactions on Semiconductor Manufacturing*, 8(1), 26–43.

Samanta, B. (2001). Multivariate control charts for grade control using principal component analysis and time-series modelling. *Transactions of the Institution of Mining and Metallurgy (Section A)*, 111, 149–157.

Seber, G. A. F. (1984). *Multivariate Observations*. Wiley.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27, 379–423.

Shewhart, W. A. (1986). *Statistical Method from the Viewpoint of Quality Control*. Dover Publications, Inc., 2nd edition.

Simoglou, A., Martin, E., & Morris, A. J. (2000). Multivariate statistical process control of an industrial fluidised-bed reactor. *Control Engineering Practice*, 8, 893–909.

Skinner, K. R., Montgomery, D. C., Runger, G. C., Fowler, J. W., McCarville, D. R., Rhoads, T. R., & Stanley, J. D. (2002). Multivariate statistical methods for modelling and analysis of wafer probe test data. *IEEE Transactions On Semiconductor Manufacturing*, 15(4), 523–530.

Sobrino, M. D. D. C. & Bravo, L. J. B. (1999). Knowledge acquisition from batch semiconductor manufacturing data. *Intelligent Data Analysis*, 3, 399–408.

Stone-Romero, E. F., Stone, D. L., & Grewal, D. (1997). Development of a multidimensional measure of perceived product quality. *Journal of Quality Management*, 2(1), 87–111.

Tryon, R. C. & Bailey, D. E. (1970). *Cluster Analysis*. McGraw-Hill.

Ündey, C. & Çinar, A. (2002). Statistical monitoring of multistage, multiphase batch processes. *IEEE Control Systems Magazine*, (pp. 40–52).

Wang, X. Z. (1999). *Data Mining and Knowledge Discovery for Process Monitoring and Control*. Advances in Industrial Control. Springer.

Ward, M. O., Yang, J., & Rundensteiner, E. A. (2003). Interactive heirarchical displays: a general framework for visualisation and exploration of large multivariate data sets. *Computers and Graphics*, 27, 265–283.

Wegman, E. (1990). Hyper-dimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85, 664–675.

Wise, B. M. & Gallagher, N. B. (1996). The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, 6(6), 329–248.

Wise, B. M., Gallagher, N. B., Butler, S. W., White, D. D., & Barna, G. G. (1999). A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics*, 13(Validate), 379–396.

Witten, I. H. & Frank, E. (2003). *Data Mining: Practical machine learning tools and techniques with java implementations*. Morgan Kaufmann, 1st edition.

Wold, S. & Sjöström, M. (1998). Chemometrics, present and future success. *Chemometrics and Intelligent Laboratory Systems*, 44, 3–14.

Wong, B. K., Bodnovich, T. A., & Selvi, Y. (1997). Neural network applications in business: A review and analysis of the literature (1988-1995). *Decision Support Systems*, 19, 301–320.

Woodward, R. H. & Goldsmith, P. L. (1964). *Cumulative Sum Techniques*. Oliver and Boyd, London.

Zuendorf, G. (2003). Efficient principal component analysis for multivariate 3d voxel-based mapping of brain functional imaging data sets as applied to fdg-pet and normal aging. *Human Brain Mapping*, 18, 13–21.

Appendix A

Principal Component Loadings

This table outlines the first three PCs for each variable from the PCA model described in Section 6.1.4. The model has 8 PCs which describe 73.14% of the variation in the original data. The variable names are taken from the semiconductor device test process and are listed in testing order.

Variable	PC1	PC2	PC3
Calc CellALoss	-0.098911	0.12087	0.22035
Calc CellBLoss	-0.098911	0.12087	0.22035
Calc PCSALoss	-0.098911	0.12087	0.22035
Calc PCSBLoss	-0.098911	0.12087	0.22035
SBST Continuity	1.4664E-19	3.2614E-18	4.2482E-17
SBCK Continuity	-2.2161E-21	-1.2903E-19	1.9715E-17
SBDT Continuity	-1.3663E-21	-1.1618E-19	2.7257E-17
PAON Continuity	7.4139E-22	1.5514E-19	-1.6252E-16
SBST leakage hi	2.9478E-23	1.1056E-20	-1.893E-17
SBCK leakage hi	-1.6935E-24	-1.0695E-21	2.7583E-18
SELECT A leakage hi	2.56E-26	2.6765E-23	-1.0151E-19
PAON leakage hi	-4.8379E-27	-8.4649E-24	4.8005E-20
SBDT leakage hi	2.0186E-28	6.0221E-25	-5.2156E-21
LOCK DET leakage hi	-4.6208E-30	-2.2966E-26	2.947E-22
IDD IN IDLE	-0.044396	0.034747	0.058924
IDD RF IDLE	-0.013285	0.053541	-0.032405

IDD MSM IDLE	0.011103	0.0030656	-0.026379
IDDtotal Idle	-0.030479	0.062005	-0.0037004
DeviceID	-3.9403E-38	-2.253E-33	1.8199E-28
Reg02 Default	7.9243E-40	7.389E-35	-8.6262E-30
Reg03 Default	-1.6228E-41	-2.4663E-36	4.1588E-31
Reg04 Default	4.1992E-44	1.0337E-38	-2.5002E-33
Reg05 Default	-3.2217E-45	-1.2813E-39	4.4302E-34
Reg06 Default	-1.1203E-46	-7.2434E-41	3.6064E-35
Reg07 Default	-1.2098E-48	-1.2698E-42	9.0887E-37
Reg08 Default	-5.6165E-50	-9.5073E-44	9.7106E-38
Reg09 Default	3.1807E-52	8.7783E-46	-1.2954E-39
Reg10 Default	6.9211E-54	3.073E-47	-6.4519E-41
Reg11 Default	-2.2857E-58	-1.0029E-51	8.8693E-46
Reg12 Default	-7.6027E-58	-8.7439E-51	3.7201E-44
Reg10Value CELL	-1.7898E-60	-3.3678E-53	2.0772E-46
Reg8Value CELL	3.0227E-62	9.0284E-55	-7.8078E-48
Reg9Value CELL	9.8914E-64	4.7625E-56	-5.8727E-49
IDD IN Puncture	-0.04619	0.17417	-0.010088
IDD IN Puncture2	-0.051676	0.15433	-0.027143
Idd Delta VCO	-0.006705	0.095103	0.02585
IDD RF Puncture	-0.01469	0.11404	-0.060386
IDD MSM Puncture	0.00082744	-0.065483	-0.0098582
IDDtotal Puncture	-0.046001	0.16675	-0.015106
PDCP UP	0.0069478	-0.0044909	-0.039106
PDCP DN	0.0036439	0.12926	0.0072363
N div ratio	0.046572	0.01595	0.00017212
N div ratio	-0.011488	0.033802	-0.0076266
N div ratio	-0.0010003	0.032312	0.019269
R div ratio	0.019457	-0.0032966	-0.013848
R div ratio	-0.018713	0.065668	0.013877
R div ratio	-0.0078352	-0.0087374	-0.026351
LD Freq	-0.0065331	0.023636	-0.0045178
CPcurr DN lowV1	-0.00025849	0.065347	0.019482
CPcurr DN highV1	-0.011458	0.073051	0.034894

CPcurr UP lowV1	0.013321	-0.11571	0.00011945
CPcurr UP highV1	0.013365	-0.11519	-0.01391
CPcurr DN lowV1	0.011188	0.11937	0.0086351
CPcurr DN highV1	0.0039958	0.12893	0.0071261
CPcurr UP lowV1	-0.01276	-0.1349	0.047482
CPcurr UP highV1	-0.02728	-0.10443	0.036221
Mstic2 DC Current	0.084634	0.051505	0.094439
IQ IMPEDANCE	-0.063631	-0.082033	0.074691
Mstic2 Math result	0.03375	0.017865	-0.032836
Voffset	0.079493	0.037735	0.094592
Vpeak	0.070821	0.0329	0.11305
CDMA InputGain	0.070819	0.032898	0.11305
MaxCDMA Pout CELLA	0.095019	-0.16588	0.11514
Mean Pout CELLA	0.19041	-0.024266	0.063275
Mn ACPR rated CA CP	0.19058	-0.023043	0.063326
Mn ACPR rated CA WC	0.17793	0.010922	-0.0055102
IDD IN CELLA	0.093238	0.17068	-0.043287
IDD RF CELLA	0.094481	0.089905	0.075511
IDD MSM CELLA	0.038665	-0.10024	-0.071253
Mn RXNOISE CDMA CA	0.1654	-0.099476	-0.1121
ACPR rated CELLA WC	-0.10007	0.077033	-0.16265
RXNOISE CDMA CELLA	-0.087083	-0.081493	-0.22241
IDDtotal CELLAPout	0.17538	0.071623	0.0016349
MaxCDMA Pout CELLB	0.12199	-0.16944	-0.0054337
Mn Pout CELLB	0.19452	-0.05364	-0.020356
Mn ACPR rated CB CP	0.19361	-0.055345	-0.022459
Mn ACPR rated CB WC	0.17594	0.0376	0.016034
IDD IN CELLB	0.092068	0.16939	-0.046741
IDD RF CELLB	0.062805	0.1291	0.06947
Mn RXNOISE CDMA CB	0.16902	-0.09978	-0.10381
ACPR rated CELLB WC	-0.10165	0.17693	0.073102
RXNOISE CDMA CELLB	0.17832	-0.089725	-0.084092
IDDtotal CELLBPout	0.15613	0.11148	0.0085752
Reg10Value PCS	0	0	0

Reg8Value PCS	0	0	0
MaxCDMA Pout PCSA	0.042467	-0.13866	0.17273
Mean Pout PCSA	0.14786	-0.082112	0.14923
Mn ACPR rated PCSA CP	0.14754	-0.083134	0.15012
Mn ACPR rated PCSA WC	0.13195	0.061921	0.059398
IDD IN PCSA	-0.014063	0.15612	-0.082528
IDD RF PCSA	0.085654	0.073602	0.035866
Mn RXNOISE CDMA PCSA	-0.11666	0.12907	0.19016
ACPR rated PCSA WC	-0.031031	0.17574	-0.11851
RXNOISE CDMA PCSA	-0.096827	0.12014	0.2212
IDDtotal PCSAPout	0.069465	0.12651	0.16985
MaxCDMA Pout PCSB	0.034277	-0.15823	0.20496
Mn Pout PCSB	0.098822	-0.10244	0.23123
Mn ACPR rated PCSB CP	0.099647	-0.10261	0.2298
Mn ACPR rated PCSB WC	0.039676	0.034842	0.22448
IDD IN PCSB	-0.0065487	0.1501	-0.097348
IDD RF PCSB	0.097493	0.093249	-0.056841
Mn RXNOISE CDMA PCSB	0.035425	-0.13274	0.23168
ACPR rated PCSB WC	-0.078699	0.1712	-0.023121
RXNOISE CDMA PCSB	-0.033681	-0.13315	0.1838
IDDtotal PCSBPout	0.097363	0.12704	0.071893
CW OUTPUT 10 PCSB	0.15439	0.1024	-0.015066
CW OUTPUT MIN PCSB	0.15439	0.1024	-0.015066
GainSlope PCSB	-0.15439	-0.1024	0.01507
CW OUTPUT 10 PCSA	0.15964	0.094146	-0.032951
CW OUTPUT MIN PCSA	0.15964	0.094145	-0.03295
GainSlope PCSA	-0.15964	-0.09415	0.032954
CW OUTPUT 10 CELLB	0.15593	0.10194	-0.00004123
CW OUTPUT MIN CELLB	0.15593	0.10195	-0.000041399
GainSlope CELLB	-0.15592	-0.10195	0.000033709
OUTPUT CELLA	0.18669	-0.080949	-0.063318
CarrierSup CellA	-0.016053	0.071739	0.0028572
ImageRej CELLA	-0.018154	-0.01623	0.017195
Image Carrier Comp	-0.021892	0.04032	0.0089787

CW OUTPUT 10 CELLA	0.14914	0.10947	0.017316
CW OUTPUT MIN CELLA	0.14914	0.10947	0.017316
GainSlope CELLA	-0.14914	-0.10947	-0.017311
VCO Osc S11 delta	0.085948	-0.03388	-0.2321
RI internal	0	0	0
Test time-sec	-0.012206	0.03473	-0.024098

Appendix B

Model Variance Captured

The variance captured by each component in the PCA model described in Section 6.1.4 is shown overpage. Figure B.1 summarises the percentage contribution each PC loading vector has on the original variables. The higher order PCs are dominant on variables $105 \rightarrow 115$, indication significant variation. The sparse regions indicate variables that have no input to the model.

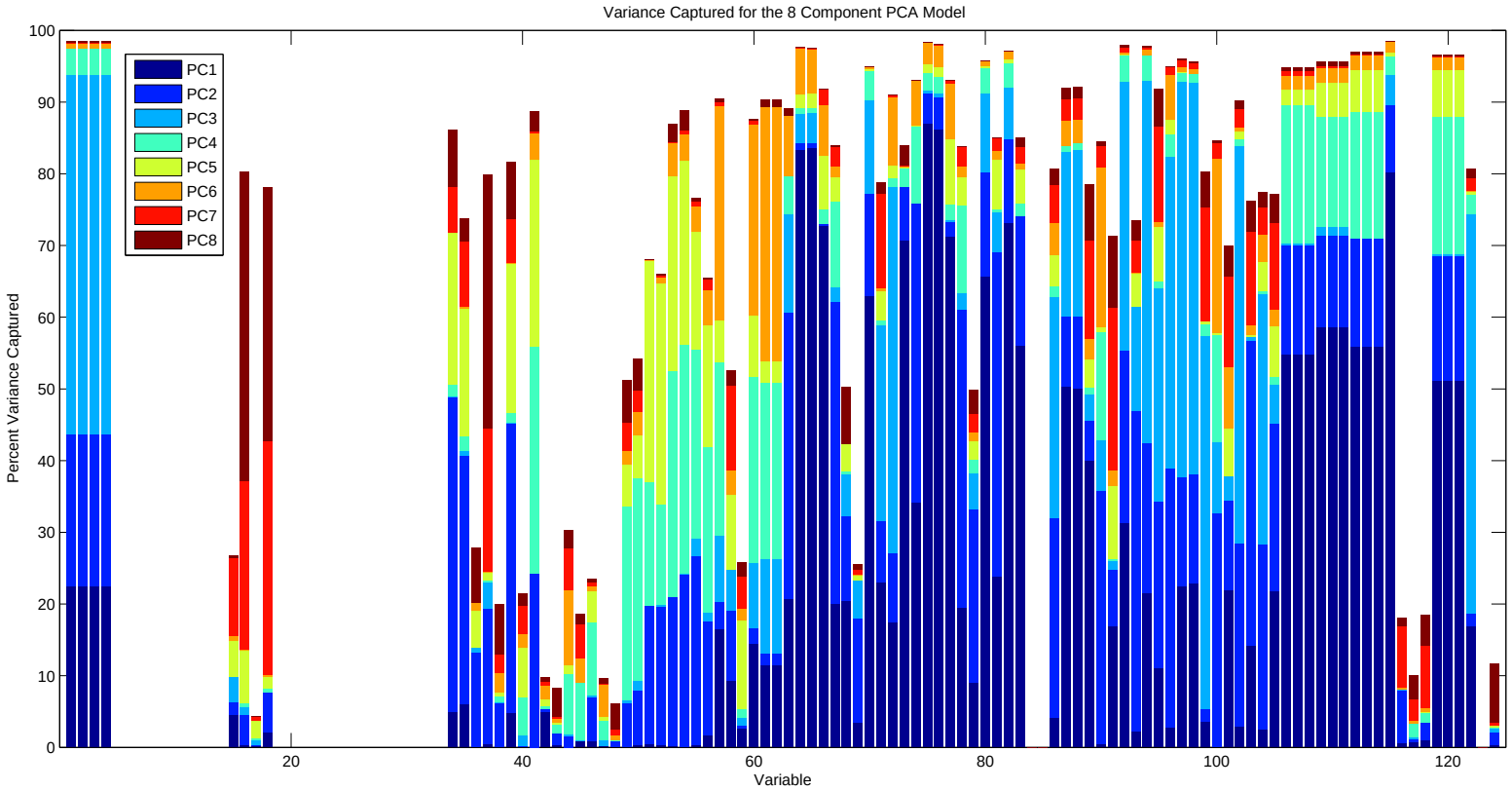


Figure B.1. PCA model variance. *A summary of % contribution to the original variables for each PC. This shows the variance captured by the PCA model in each original variable.*

Appendix C

NOC Model with PC1-PC2-PC3 Subplots

Figure C.1 shows a NOC ellipsoid in 3-D and its constituent 2-D subplots. the purpose for the plot is to identify different regions of PC space as a visualisation method to classify the new scores from the existing NOC model.

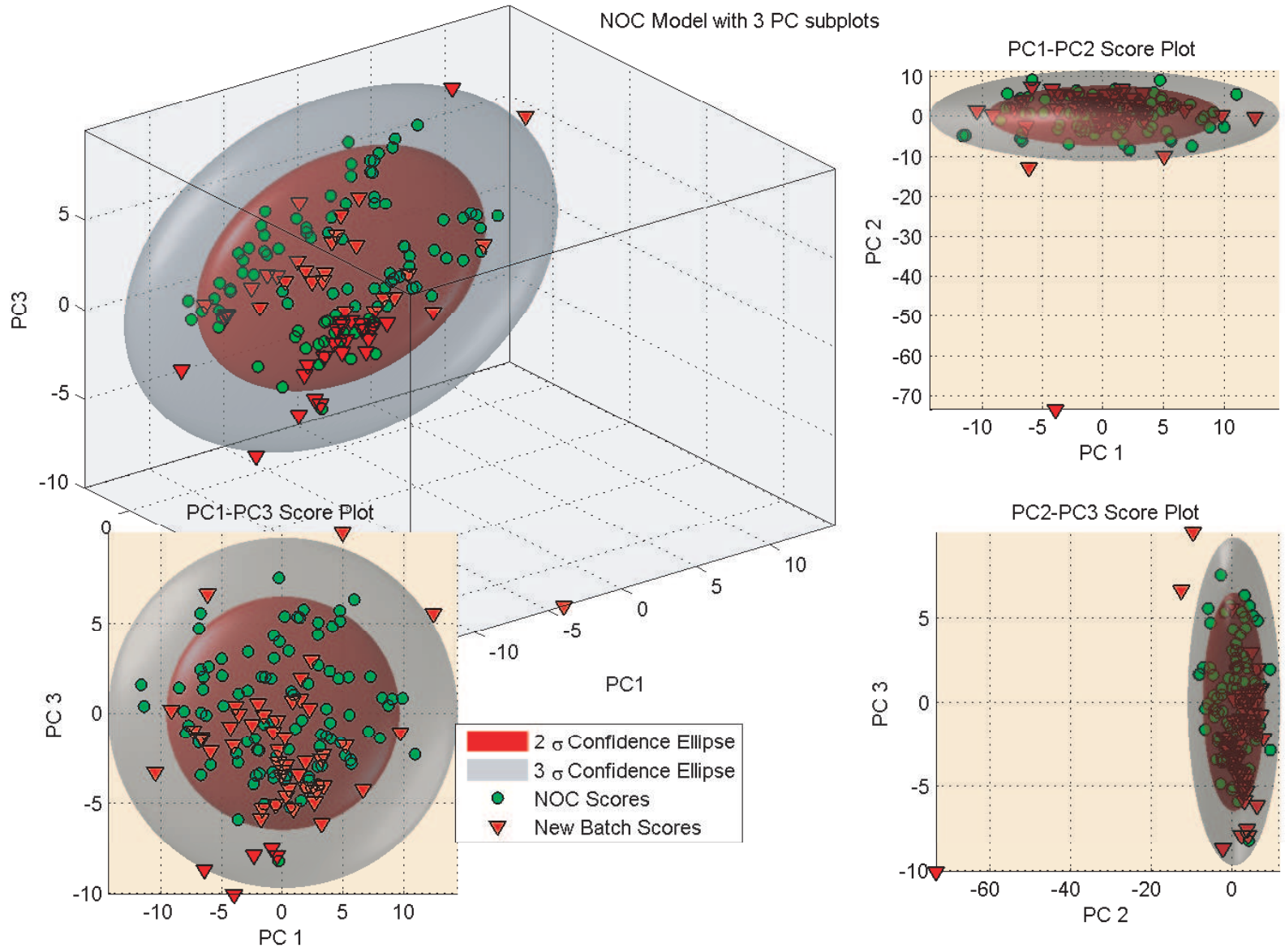


Figure C.1. NOC model with PC1-PC2-PC3 subplots. *This figure details the result of testing new scores with the existing model. The graphical result shows which* new scores *fall within the model and which do not.*