NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

Unsupervised Feature Extraction Techniques for Plasma Semiconductor Etch Processes

By

Beibei Ma, MEng

A thesis presented on application for the degree of

Doctor of Philosophy

Department of Electronic Engineering

National University of Ireland, Maynooth

December 2009

Head of the Department: Dr. Seán McLoone

Supervisors: Dr. Seán McLoone, Prof. John Ringwood

Abstract

As feature sizes on semiconductor chips continue to shrink plasma etching is becoming a more and more critical process in achieving low cost high-volume manufacturing. Due to the highly complex physics of plasma and chemical reactions between plasma species, control of plasma etch processes is one of the most difficult challenges facing the integrated circuit industry. This is largely due to the difficulty with monitoring plasmas.

Optical Emission Spectroscopy (OES) technology can be used to produce rich plasma chemical information in real time and is increasingly being considered in semiconductor manufacturing for process monitoring and control of plasma etch processes. However, OES data is complex and inherently highly redundant, necessitating the development of advanced algorithms for effective feature extraction.

In this thesis, three new unsupervised feature extraction algorithms have been proposed for OES data analysis and the algorithm properties have been explored with the aid of both artificial and industrial benchmark data sets. The first algorithm, AWSPCA (Adaptive Weighting Sparse Principal Component Analysis), is developed for dimension reduction with respect to variations in the analysed variables. The algorithm generates sparse principle components while retaining orthogonality and grouping correlated variables together. The second algorithm, MSC (Max Separation Clustering), is developed for clustering variables with distinctive patterns and providing effective pattern representation by a small number of representative variables. The third algorithm, SLHC (Single Linkage Hierarchical Clustering), is developed to achieve a complete and detailed visualisation of the correlation between variables and across clusters in an OES data set.

The developed algorithms open up opportunities for using OES data for accurate process control applications. For example, MSC enables the selection of relevant OES variables for better modeling and control of plasma etching processes. SLHC makes it possible to understand and interpret patterns in OES spectra and how they relate to the plasma chemistry. This in turns can help engineers to achieve an in-depth understanding of underlying plasma processes.

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Seán McLoone for his *tremendous* assistance, patience and encouragement and *excellent* guidance throughout my PhD.

I would also like to thank my co-supervisor Prof. John Ringwood for his valuable advice, support and patience with me.

The assistance and cooperation from my industrial mentor Niall Macgearailt are truly appreciated.

The financial support provided by Intel (Ireland) Ltd. and Enterprise Ireland is gratefully acknowledged.

Thanks to all my colleagues and friends in the Dynamics and Control Group (Shane Lynn, Dr. Emanuele Ragnoli, Georgio Bacelli, Shane Butler and Francesco Fusco), in the Biomedical Engineering Research Group (Kevin Sweeney, Violeta Mangourova, Damian Kelly, Lorcan Walsh, Claire Dormer) and Dr. Xiaochen Liu (in the Irish Climate Analysis & Research Units) for their help and friendship.

Finally, I would like to thank my parents Yunfei Ma and Jie Guan and my partner William C. Flynn and his parents William P. Flynn and Vera Flynn for their love and encouragement over the years.

Contents

Chapter 1

Introduction

The semiconductor industry has experienced exceptional growth since the invention of integrated circuits in 1960. As predicted by Moore's law [122], the number of transistors on an integrated circuit has doubled roughly every 2 years. Semiconductor devices constitute the foundation of the electronics industry, which is currently one of the largest industries in the world. An important characteristic of the semiconductor industry is the rapid pace of improvement in its products [117], stimulated by the sustained and substantial research and development (R&D) investment. Efficient capital investment in R&D is therefore vital for companies to remain competitive when faced with global economic competition, especially from major electronic firms located in US, Japan, Europe, Korea and Taiwan.

To promote competition and guide research to the areas that need most breakthroughs, a technology roadmap has been provided by the International Technology Roadmap for Semiconductors (ITRS) and updated every two years. According to the 2007 ITRS roadmap [24] for wafer design parameters over an 8-year period, the wafer diameter is required to increase from 300 to 450mm while the critical dimensions are required to decrease from 80 to 22nm, to meet the industry historical 30% cost-per-function reduction and 50% cycle time improvement in manufacturing per decade.

In the past cost reductions were obtained via yield improvement, but when yield limits are reached, further improvements must come from increased capital equipment

utilization [33], that is, maximizing throughput of products with reduced setup and maintenance costs. This objective was achieved in the late 1980's and early 1990's [145] by the wide application of statistical process control (SPC) techniques to monitor process faults. With the aid of SPC charts, values outside of the control limits can be taken as indicators of possible process failure. As the semiconductor industry moved into nano-scale manufacturing, traditional SPC was unable to deal adequately with the resulting tighter operating tolerances and increased process complexities, leading to substantial increases in undetected process errors and false alarms. Undetected process errors in semiconductor can lead to the destruction of an entire batch or batches of wafers with no hope of recovering the product through further processing, causing serious rises in manufacturing cost. In a modern manufacturing plant, the average cost for making a chip is US $40 dollars. If the chip size were 140mm^2, damaging a 300mm wafer with 430 gross dies will cost US$17,200 dollars.

In seeking possible alternatives to SPC, a broad advanced process control (APC) methodology has been adopted. APC is employed to maximise the use of available information about material, processes, diagnostic data and desired targets, select model and control strategies, estimate the feasibility of the desired targets and generate the necessary alarms for process faults [33].

A typical semiconductor manufacturing process often involves several hundred unit operations, among which, plasma etching has been recognised as one of the main unit operations that has a decisive effect on product yield [33]. The main focus of existing modeling and control studies has been on plasma etching, photolithography and deposition [33]. However, due to the high complexities of plasma physics and etching chemistry and plasma sensitivity to subtle process variations, the plasma etching process still poses great difficulty with respect to achieving effective APC.

1.1 Plasma Etching

Plasma is considered as a gas containing an electrically neutral medium of ions and electrons, dissociated from a proportion of the atoms or molecules. Plasma is achieved

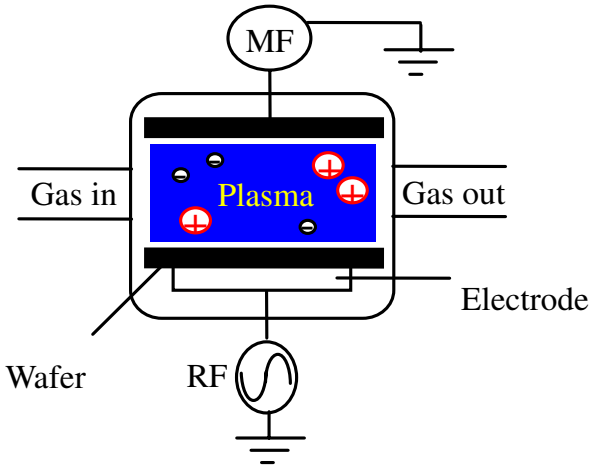


Figure 1.1: A diagram showing the basic features of a plasma etching chamber: MF=Microwave Frequency; RF=Radio Frequency.

by supplying sufficient microwave energy to a gas to allow significant numbers of electrons to break free from their atoms, forming free moving electrons and coexisting with equally charged ions. The diagram of a typical plasma etching chamber is shown in Fig. 1.1. Gas is pumped into the chamber under vacuum and ionised using a high power Microwave Frequency (MF) source to create a plasma. By applying a Radio Frequency (RF) external electrical field to the plasma, the ions can be accelerated towards the wafer surface, where they interact both chemically and physically with the silicon wafer, etching away the exposed surface. Physical etching occurs when the wafer surface is bombarded by the positive ions which travel at high speeds. Material on the target surface is removed due to collisions with the incoming ions. By adjusting the external bias voltage on the wafer, the direction and speed of ions can be controlled, yielding a highly directional etch. However, pure mechanical collisions have very little selectivity. Chemical etching occurs when chemical compounds on the wafer surface are exposed to the chemically reactive species in the plasma. By appropriate selection of the chemical species in the plasma, chemical etching can target specific compounds on the target surface, thus producing a 'selective' etch.

Plasma etching is a form of plasma processing used for integrated circuit (IC) manufacturing. Plasma etching emerged as an alternative to acid bath chemical etching (wet

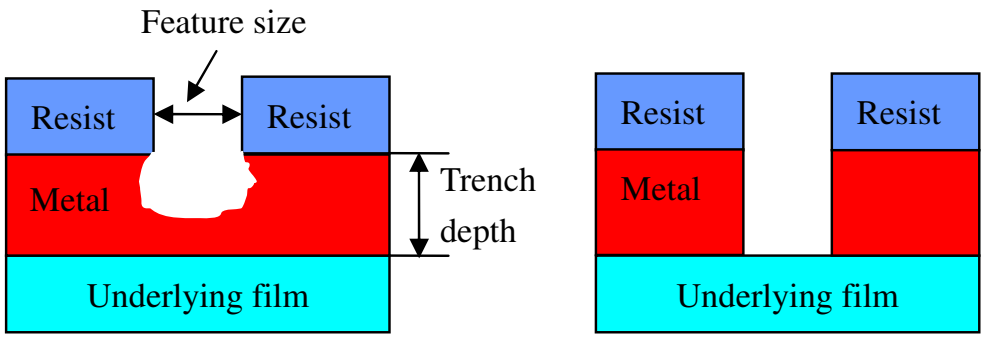


Figure 1.2: Difference between plasma and chemical etching: (a) Chemical etching with undercutting, which is characterized by not clearing out completely the film being etched; (b) Ideal plasma etching.

etching) in the late 1960s, as a result of increasing demand for smaller feature sizes and tighter tolerances. The main advantage of plasma etching is that the direction of etching can be controlled. In contrast, wet etching using acid baths proceeds in all directions with similar speed (isotropic etching) [106]. As feature sizes continue to shrink, it is crucial to have etching with high directionality to guarantee product quality [106]. As an example, when the feature size is less than trench depth, the trench cannot be removed completely using chemical etching while retaining the desired feature size, leading to problems such as short circuits. This is illustrated in Fig. 1.2 (a). In contrast, plasma etching can be directed to remove material at the bottom of a trench while leaving the same material on the sidewalls unaffected, as shown in Fig. 1.2 (b). Generated etch by-products are volatile at room temperature and can be easily cleared by the flowing gases.

1.2 Plasma Monitoring

Control of plasma etch processes is one of the most difficult challenges that faces the IC industry. This is due to the highly complex physics of plasma and chemical reactions between plasma species. In industrial practice, most silicon chip manufacturers rely on the rigorous adherence to a process 'recipe' for the various etch processes, which are often operated empirically with little understanding of the underlying physics and chemistry. With the continuing drive towards smaller feature sizes (nanometer scale

presently), this lack of understanding becomes a significant problem [33]. As such, the development of new and better plasma sensors becomes essential for successful application of APC to nanometer-scale manufacturing.

In industry, the most widely used plasma etching monitors are the Langmuir probe, plasma impedance monitor and optical emission spectrometer.

1.2.1 Langmuir Probe

A Langmuir probe is a small device that can be used to determine the electron temperature, electron density and electric potential of a plasma. It works by inserting one or more electrodes into a plasma. The electric potential between the various electrodes is varied, leading to changes in the ion or electron currents that flow to a plasma [19]. The relationship between the resulting current and voltage are recorded in the so-called I-V characteristic curve, which can be used to determine the physical properties of a plasma.

A Langmuir probe is able to provide direct measurements of plasma properties [19]. However, the placement of the Langmuir probe is intrusive to the production environment and since it interacts directly with the plasma, it can significantly impact on the operating conditions of the chamber and the uniformity of interaction of the plasma with the wafer surface. Thus, the measurements obtained by the Langmuir probe are not reliable.

1.2.2 Plasma Impedance Monitor

A Plasma Impedance Monitor (PIM) is a non-intrusive plasma diagnostic sensor, used to measure the currents, voltages and phases of the RF power supply delivering power to the plasma chamber.

A typical plasma etching chamber with a PIM is illustrated in Fig. 1.3. The plasma behaves as a variable impedance in the RF circuit. This impedance is a complex and nonlinear function of chemistry and energy of the plasma and as such reflects the underlying properties of the plasma etch. A match box is used to adjust the input

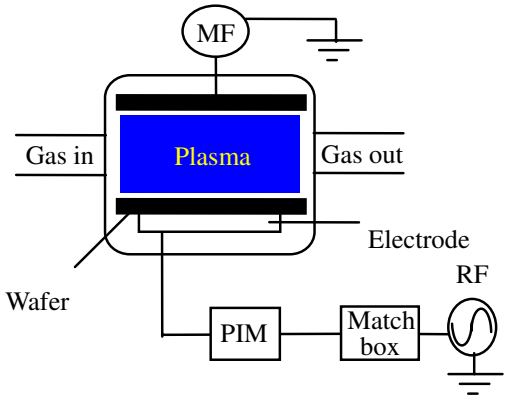


Figure 1.3: The diagram of a typical plasma etching chamber with the PIM; MF = Microwave Frequency, RF = Radio Frequency.

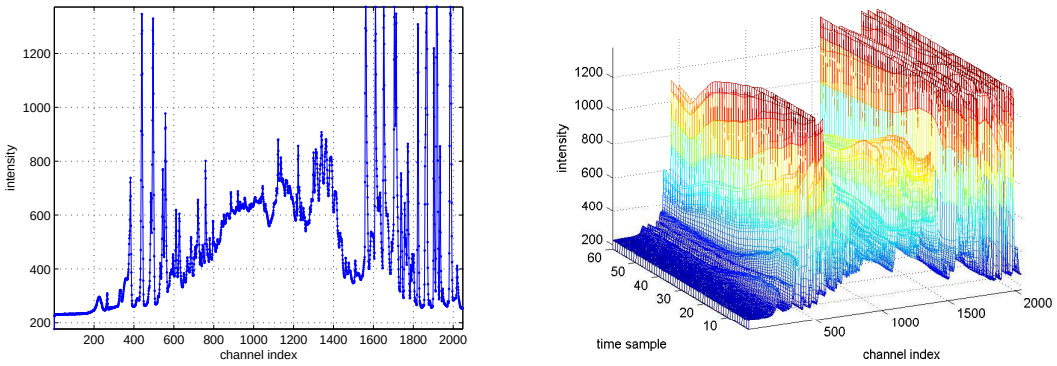
impedance to be equivalent to the impedance produced by the plasma to maximise the power transferred from the RF source to the plasma chamber. Under these settings, a PIM is placed between the match box and the electrode.

Along with the voltage, current and phase measured at the fundamental frequency, modern PIM sensors can record many harmonics of these signals. The SmartPIM, developed by the Scientific Systems Incorporation, for example, measures the fundamental frequency and 52 Harmonics. These high frequency harmonic signals have been found to be very sensitive to the subtle changes in a production environment and hence can be used for monitoring plasma etching processes. The experiments conducted by Dewan *et al* [30], for example, showed that it is feasible to use PIM signals to detect etch end point in a plasma etch process where SF_6 is used to etch a SiO_2/Si layer.

1.2.3 Optical Emission Spectrometer

An important characteristic of plasmas is that they emit light. This optical emission occurs because excited electrons are continually falling from higher to lower energy levels, releasing energy in the form of a photon emitted at a particular wavelength which is a function of the change in energy levels [154]. The measured optical emissions, often referred to as the fingerprints of species, can be used to indicate the changes of a particular species in an etching process.

Optical Emission Spectroscopy (OES) is an non-intrusive plasma diagnostic technique and has been widely employed in industry for measuring the chemical changes in a plasma [151]. OES data contains rich chemical information and has the potential to be used to track the root causes of process variations and realise an in-dept understanding of underlying plasma physics and chemistry, the foundations of APC. OES measures the optical emission intensities as a function of wavelength, time and location. As an example, Fig. 1.4 (a) shows the OES footprint of a plasma at a particular sample instant in a wafer etch step. The time evolution of the footprint over the full etch step is illustrated in Fig. 1.4 (b). This is a 3-D plot with emission intensity on the z-axis and time and channel index (wavelength) on the x and y axes, respectively.



(a) (b)

Figure 1.4: Plasma etch OES data for a single wafer: (a) Recorded at a single instant; (b) Recorded over a complete etch step.

The existing applications of OES mainly rely on engineers having a detailed knowledge of the underlying process chemistry and dynamics so that the most relevant wavelengths can be identified and used to achieve etch process control. This approach is problematic. The learning process is time consuming, the selection of the key wavelengths based on engineers' personal experience is subjective and the effectiveness of the approach is limited to a particular process (given changes in the process recipes or etching products, the effectiveness of the selected wavelengths can be destroyed).

As feature sizes continue to shrink, OES sensors have been designed to be more and more sensitive to subtle changes in the process. The generated OES data sets consist of thousands of variables, measured over tens or hundreds of time samples. OES data is inherently highly redundant with the result that it is difficult to recognise useful features and key wavelengths by direct visualisation. It thus, becomes necessary to employ automated feature extraction algorithms.

1.3 Feature Extraction

Feature extraction, as a technical term, originated in the fields of pattern recognition and image processing. As stated in [102], feature extraction should focus on extracting from raw data the information which is crucial for classification purposes. With the continuing expansion of the application of feature extraction techniques towards more diversified fields, nowadays, it has become impossible to provide an unified and accurate definition of feature extraction [155], or an effective categorization of the theories and algorithms for tackling these feature extraction issues [103]. As Selfridge and Neisser [138] pointed out, feature extraction algorithms have to be designed individually to effectively tackle an unknown issue.

In the context of our research, we define the features as the extractable patterns contained in the sensor measurements that can be used to indicate plasma etch process characteristics, such as etch rate, process variations, faults, etch endpoint, etch change point, *etc.* Feature extraction is defined as applying effective methods to extract useful features while excluding any uncorrelated/corrupted features from the data.

Feature extraction algorithms can be divided into three categories: supervised feature extraction, semi-supervised and unsupervised feature extraction. In supervised feature extraction, examples of the target outputs are available to guide the selection of the effective algorithm. If the features selected cannot match the target features, new algorithms need to be explored. The selection process is conducted repeatedly until the algorithm can identify the features as targeted. In this way, the available information can be transferred to the feature extraction algorithm through iterative selection. In

unsupervised feature extraction, there is no available information to supervise the selection process. The actual features extracted cannot be compared with any existing target features and hence, the selection is free of guidance. In semi-supervised feature extraction, both supervised and unsupervised feature extraction are employed, typically a large amount of unsupervised feature extraction used in conjunction with a small amount of supervised feature extraction [192]. When the acquisition of the supervised information is expensive, semi-supervised learning can be of great advantage with respect to cost and algorithm effectiveness.

In our research, the acquisition of *in situ* measurements is unreliable and the offline methodology is costly. In a practical production environment, the *in situ* measurements of the process parameters such as trench depth, etch rate and wafer surface thickness must be obtained by directly inserting the diagnostic devices into an etching chamber, which can disturb the plasma, making the measurements unreliable. To obtain the offline metrology data, the testing has to be conducted after the actual process is finished, so it is time consuming and expensive. Considering these existing problems in practice, there is a need for the development of new feature extraction algorithms that can operate effectively on OES data in an unsupervised manner.

1.4 Aims and Scope of Thesis

This thesis will focus on the development of unsupervised feature extraction algorithms applied to extract different patterns and identify representative variables in complex OES data, in order to provide effective variable selection for further modeling and control. The developed algorithms should be able to achieve effective low-dimensional reconstruction and summarisation of the high-dimensional data, be able to achieve feature classification based on feature differences and feature representation by a small number of variables. Moreover, the developed algorithms should be able to capture and show the different levels of similarity between variables with similar features.

1.5 Contributions of This Thesis

The major contributions of this thesis are as follows.

1. Two data summarisation methods based on the use of PCA (Principal Component Analysis) are proposed that provide more efficient computation when dealing with the high-volume and high-dimensional OES data sets. One is implemented as an improvement on conventional data unfolding approaches and the other is realised by monitoring changes in the directions of the PC loading vectors. The two proposed methods can provide effective identification of plasma etching process variations across wafers and across lots.

2. The recently proposed Sparse Principal Component Analysis (SPCA) algorithm [196] has been applied to OES data analysis for the first time. The properties of the algorithm have been fully investigated, with the aid of artificial data sets and OES data and the strengths and weaknesses of the approach highlighted.

3. A new adaptive weighting SPCA (AWSPCA) algorithm is proposed and the algorithm numerical solutions developed. As an improvement on the recently proposed SPCA and adaptive LASSO algorithms, AWSPCA can provide more flexible control of component sparsity. In addition, the grouping effect and loading orthogonality properties that are possessed by some existing algorithms are also encouraged in AWSPCA.

4. A new clustering algorithm, Max Separation Clustering (MSC), is developed. As compared to many existing non-hierarchial clustering algorithms, MSC does not require *a priori* specification of the number of clusters and is not subject to inter-run variability.

5. A customised single linkage hierarchical clustering (SLHC) algorithm is developed for application to OES data and a new method for estimating the appropriate number of clusters, B-index, is proposed. The effectiveness of B-index is highlighted as compared to a number of existing best performers, with the aid of simulated data sets. Experiments on OES data show that the joint use of the

SLHC and B-index methods can help to recognise the similarity between intra-cluster objects and across clusters.

Other contributions in the thesis include:

1. A review of the methodology for feature extraction for plasma etching process control.
2. A new and systematic method for selecting noise filter bandwidth for OES signal filtering.
3. A new method for estimating OES sensor resolution.
4. An improved method for quantifying the estimation accuracy of SPCA.
5. A derivation of the relationship between the similarity threshold in MSC and OES data signal-to-noise ratio.
6. An in-depth discussion of the characteristics and properties of K-means, SOM and QT and the challenges when using these methods for OES data feature extraction.

1.6 Thesis Structure

Chapter 2 provides a technical description of plasma and plasma etching and a review of the methodology for feature extraction for plasma etching process control. This chapter also introduces experimental benchmark data sets that are used throughout the thesis to estimate the performance of the various algorithms developed.

In Chapter 3, the application of PCA to the analysis of OES data from plasma etch processes is explored. Conventional methods of plotting the PC scores are applied to the OES data with experimental results presented. A novel low cost method for monitoring changes in PC loadings is proposed. Experimental results show that the proposed methods are effective in identifying and capturing the process variations across wafers and across lots.

In Chapter 4, a thorough description of the theoretical frameworks of SPCA is provided. With the aid of simulated and OES data sets, the feasibility of using SPCA as a variable selection tool for identifying key variables from a large data set is examined. Experimental results demonstrate SPCA lacks flexibility in controlling model sparsity. Addressing this issue motivates the development of adaptive weighting SPCA (AWSPCA).

In Chapter 5, a novel AWSPCA algorithm and algorithm numerical solutions are proposed and developed in detail. Experimental results of applying AWSPCA to simulated and OES data sets show that AWSPCA combines many desirable properties possessed by existing PCA, SPCA and adaptive LASSO algorithms.

In Chapter 6, a survey of non-hierarchical clustering methods is provided, followed by a detailed discussion of the properties of K-means, SOM and QT, three of the most powerful and widely used non-hierarchical clustering algorithms. The insufficiency of these algorithms for the analysis of OES data is highlighted. As a solution, a novel Max Separation Clustering (MSC) algorithm is proposed and described in detail. Experimental results for the application of MSC to clustering of OES data sets are used to confirm that MSC can extract and classify the different patterns in different clusters and the newly proposed maxoid in MSC is effective for representing the patterns contained in each cluster.

In Chapter 7, a review of hierarchical clustering is provided, followed by the description of a custom single linkage hierarchical clustering (SLHC) implementation for the analysis of OES data sets. A new cluster number selection method, B-index is also proposed. Experimental results on the OES data sets show the consistency of the clustering results obtained by SLHC/B-index and MSC and as such, provides a form of validation for both methods.

Chapter 8 provides a concluding summary on the advantages and disadvantages of the proposed methods, as well as possibilities for future research.

Chapter 2

Background

This chapter provides an introduction to plasma techniques and feature extraction applications in plasma etching. To begin with, technical details on plasma generation, plasma etcher and process diagnostic devices that haven't been covered in the general introduction in Chapter One are provided. Then existing work on supervised and unsupervised feature extraction techniques used for plasma etching diagnostics is reviewed. This provides the methodology background for the thesis. The final section of the chapter is devoted to the introduction of the experimental benchmark data sets that are used throughout the thesis to illustrate the properties and estimate the effectiveness of proposed algorithms.

2.1 Plasma Etching

This section provides an overview of the mechanism of plasma generation, the Electron Cyclotron Resonance plasma etcher used for generating plasma and achieving plasma etching, and the Optical Emission Spectrometer used for plasma diagnostics.

2.1.1 Plasma Generation

Plasma is an electrically neutral gas mixed with atoms, molecules, free moving electrons and equally charged ions. The generation of the electrons and ions results from a series of collisions in a plasma, which are referred to as electron impact ionization, excitation, relaxation and recombination [17].

Ionization: When an incoming ion or electron with enough energy collides with an atom, the outermost electron of this atom can absorb energy to break the electric potential barrier that originally bound it to the atom, resulting in a free moving electron and equally charged ion. Defining A as the atom, the ionization of A can be expressed as:

$$e^- + A \rightarrow 2e^- + A^+. \tag{2.1}$$

Excitation: Excitation refers to the process of a plasma atom being activated to a higher energy level when colliding with a free moving electron, but where the absorbed energy is not enough, to break the electric potential barrier to form a free moving electron. The process can be summarised as

$$e^- + A \rightarrow A^* + e^-. \tag{2.2}$$

where A^* represents the activated atom.

Relaxation: Relaxation refers to the process of the electron in an electronically excited atom transiting from a higher energy level to a lower energy level with excess energy released in the form of a photon.

$$A^* \rightarrow A + E \text{ (photon)}. \tag{2.3}$$

The wavelength of the emitted light corresponds to exactly the energy difference between the two energy levels with

$$E = hc/\lambda, \tag{2.4}$$

where λ denotes the wavelength of the photon, c denotes the speed of light and h is Planck's constant. Thus, an atom emits light at only certain discrete wavelengths (Fig. 2.1). This phenomenon leads to the characteristic light emission of a plasma which is used in Optical Emission Spectrometry (OES) (discussed later in Section 1.3.3) to indicate the existence of the gaseous species in a plasma.

Recombination: Recombination refers to the process of an electron being combined with an ion to form a neutral atom. However, a third body is required to take part in the process to allow the recombination to satisfy the conservation of energy and momentum requirements [17]. The recombination process can be expressed as

$$e^- + A^+ + A \rightarrow A + A \tag{2.5}$$

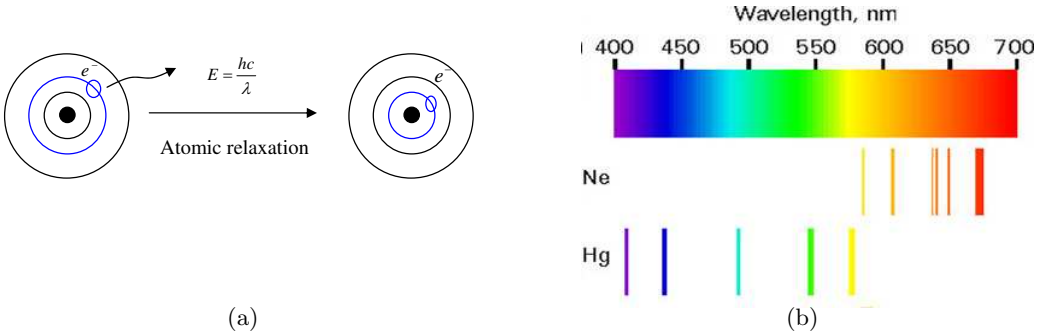


Figure 2.1: (a) Mechanism of plasma light emission. Excess energy, E, is released as the electron decays from a higher energy level to a lower one; (b) Light emission spectra of Ne and Hg.

2.1.2 Electron Cyclotron Resonance Plasma Etcher

In physics, Electron Cyclotron Resonance (ECR) refers to a phenomenon in which electrons in a static and uniform magnetic field rotate around the magnetic lines of force. The ECR plasma etcher makes use of microwave energy and a strong magnetic field to produce a low pressure and high density plasma and provides the necessities for achieving plasma etching.

The main components of an ECR etcher include a gas supply system, a magnetic field generation system, a microwave oscillator, a RF (radio frequency) generator, and an etch chamber. The diagram of an ECR etching chamber is shown in Fig. 2.2, where the OES monitor is included for ease of explanation of the OES measurements in the later text.

The 2.45GHz microwave power is oscillated by a magnetron, transmitted along a waveguide and injected into a quartz bell jar. The microwaves produce a dynamic electric field, which is perpendicular to the static magnetic field, which is generated as a DC current flowing through the solenoid coils. The interaction of these two fields generates Lorentz Force, which causes the electrons to spiral in a helical motion. In this way, the microwaves transfer the energy to free electrons which in turn accelerate and collide

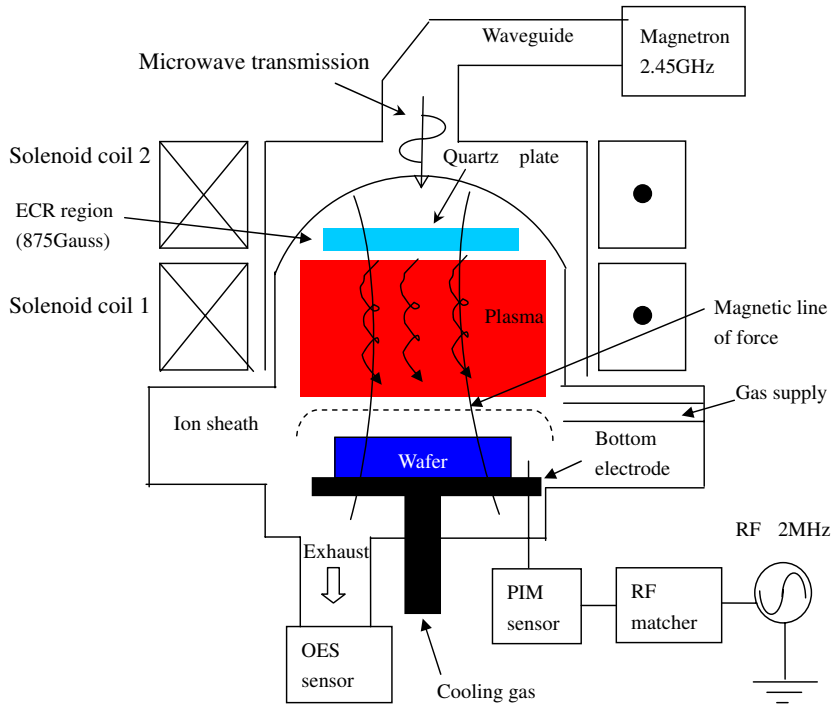


Figure 2.2: A diagram of ECR plasma etcher with the OES monitor included.

with the atoms or molecules in the gas and produce ionization. The low gas pressure, which helps to reduce electron impact recombinations, is achieved by controlling the flow rates of the gases supplied to the chamber. A separate RF bias is applied to the wafer electrode to independently control ion energy at the wafer surface. A 2MHz generator is connected to the powered electrode to create a negative DC bias on the ground electrode. This makes the plasma more positive with respect to the powered electrode, leading to an increased ion bombardment on the wafer surface, thereby achieving faster etching. Helium is pumped to the backside of a wafer to cool the wafer temperature, which is an important factor influencing the uniformity of etch across the wafer surface. An important feature of the ECR etcher is that ion energies can be controlled separately by the RF supply, allowing much greater control of etch rate due to bombardment.

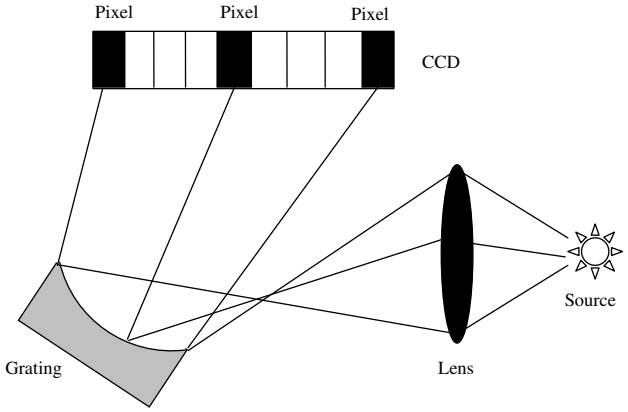


Figure 2.3: The diagram of an optical emission spectrometer.

2.1.3 Optical Emission Spectrometer

The optical emission spectrum of each individual chemical atom or molecule is unique. As such, analysis of plasma emission spectra can be used to estimate the instantaneous composition of a plasma and track the density changes of the chemical species over-time. An Optical Emission Spectrometer is an optical device used to detect the optical emissions of plasma species, providing direct information on plasma chemistry.

In Optical Emission Spectroscopy (OES), visible light is collected by a lens and focused onto a grating. The grating then redirects the light onto a Charged Coupled Device (CCD) detector with different wavelengths dispersed to different CCD pixels as shown in Fig. 2.3. The key component of a typical Optical Emission Spectrometer is the CCD detector. CCDs are a type of quantum detectors, which are used to measure the flux of photons. In contrast to thermal detectors, which are used to measure the optical power, quantum detectors have a faster response time and are more sensitive to small photon fluxes and therefore, have been widely employed in modern optical detection devices.

In CCD detectors, the photons are detected by a photoactive detection area and converted to an electrical signal by a photoelectric device. Applying the added external voltage supply, the produced electrical signal or electron is then moved to a capacitor. Accumulation of charge proceeds as more electrons are stored in the capacitor until the

capacitor is discharged for the readout. The readout analog signals are converted to digital signals through an A/D (analog-to-digital) convertor and recorded. The main advantages of CCD detectors are that they are small and have a high photon-to-electron ratio. Our OES data is collected using an Ocean Optics USB2000 spectrometer with a CCD detector consisting of 2048 pixels (corresponding to 2048 wavelengths). As shown in Fig. 2.2, the OES monitor is normally connected to the exhaust plasma leaving the chamber as this allows etch byproducts in the plasma to be detected.

2.2 Survey of Existing Work

While the relevant literature is reviewed in each chapter, a brief survey of existing work on the application of feature extraction (supervised and unsupervised) to the analysis of complex plasma etching diagnostic data sets is documented in this section, so that the contributions of the thesis can be placed in context.

2.2.1 Unsupervised Feature Extraction Algorithms

The objectives of unsupervised feature extraction methods are to extract distinctive and representative information from the data. Due to the lack of a teacher's knowledge, the usefulness of the obtained information cannot be estimated. Hence, unsupervised feature extraction techniques are mainly drawn from the statistical analysis domain. In this section, a survey of unsupervised feature extraction algorithms used for analysing plasma etching diagnostic data sets is provided.

Principal Component Analysis

Principal component analysis (PCA), as a linear multivariate data projection technique, has been widely employed for data compression and visualisation. [69]. PCA provides low dimensional representations of high dimension data sets while retaining the most information in the data in terms of variance explained [109].

The main application of PCA in etch process analysis has been in the detection of etch endpoint. Etch endpoint refers to the transition of etching between two layers of material on a wafer signifying that etching should be stopped. Yue *et al.* [185]

proposed that the sharp changes in the amplitude of the second and third PCA scores can be used to indicate the etch endpoint. PCA can also be used for key wavelength selection. As an example, Yue *et al.* [186] proposed a 'sphere criterion' method, which aims to select wavelengths based on the joint use of a few principal components (PCs). Suppose that the first l PCs are employed; According to the 'sphere criterion,' the i^{th} wavelength can be selected if it satisfies

$$\sum_{j=1}^{l} p_{ij}^2 \geq r^2 \tag{2.6}$$

where p_{ij} denotes the amplitude of the i^{th} wavelength in the j^{th} loading and r is a threshold or the so-called radius of the sphere. One advantage of the 'sphere criterion' method is that the wavelength selection is based on its performance *w.r.t.* all the PCs that are used to represent the data and hence, can more truly reflect the importance of selected wavelengths. In addition, the 'sphere criterion' gives users some control over the selection of the number of key wavelengths through the use of different values of r.

Direct visualisation of the 2-D plots of PC scores was shown to be useful for spotting changes in process states, such as power [170], different percentage of gas composition [170] and changes across lots [174]. In [165], PCA is used as a signal filtering method to have better use of OES data for end point detection. Koh *et al* [89] reported that if PCA is applied to PIM data, changes in the 2-D plots of PC loadings can also be used to indicate etch endpoint.

To determine the number of principal components in a PCA model for best reconstruction, a typical method is to set up a threshold of variance explained. Han *et. al.* [52] proposed that if one PC can explain over 10% of the total variance, then that PC should be considered as a significant PC. Joint application of this PC number selection method and the 'sphere criterion' shows that the selected wavelengths can work efficiently in finding the etch endpoint for wafers with etch open area of 10% or less.

Toprac *et al.* [161] and White *et al.* [171] proposed a simpler way to select the number of PCs. For plasma etch processes, they found that the first four PCs were sufficient for representing the patterns that exist in OES data. As another example, Qin and

Dunia [125] proposed the so-called variance of reconstruction error (VRE) method, which guarantees that the reconstruction has a minimum error over the number of PCs. When applying PCA to analyse complex data sets, one important issue is how to rearrange the data sets for further analysis. For OES data sets, methods of unfolding the 3-D measurements into 2-D along different dimensions have been widely discussed in [178, 44, 144, 174, 175].

Independent Component Analysis

As another multivariate statistical process method, Independent Component Analysis (ICA) is used to separate mixed signal sources into a few factors that are mutually independent. Although ICA has found considerable application in areas such as blind source separation [14], the application to semiconductor etch processes has been limited [100]. A recent application of ICA [100] has been reported to outperform PCA in fault detection and diagnosis in conjunction with Hotelling's T^2 and Sum of Predicted Error (SPE) methods. He *et al.* [57] proposed a new method which jointly uses independent component analysis (ICA) and multi-way PCA. The method has been shown to have significant benefits when the time-series variables are not subject to Gaussian distributions.

Segmental Semi-Markov Model

In a stable system, the continuity of measured signals can often be interrupted by change events, leading to segments in the measured time sequences. The corresponding segmental points are defined as change points. As an example, a change point is shown in Fig. 2.4. Detecting change points from Interferometry sensor measurements, Ge and Symth [42, 43] proposed a segmental semi-Markov model, which is an extension of the standard hidden Markov model. Segmental semi-Markov models allow the variables to have different distribution models in different time segments. Sudden change in the variable distribution is regarded as occurrence of the change point. Experimental results on both simulated and real semiconductor manufacturing data show the accuracy of the proposed framework in change point detection [42, 43].

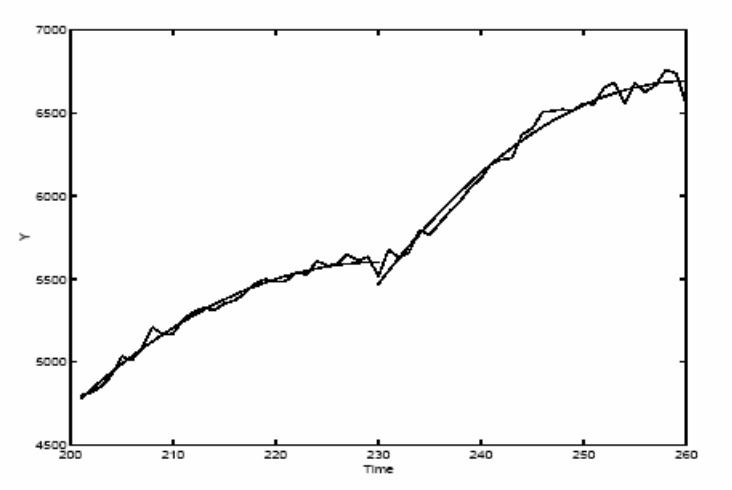


Figure 2.4: An illustrative example of a change point problem [43].

The drawback of the semi-Markov model is that the method focuses on the change-point detection for a single variable. For multivariate and multiple change-point detection problems where the time-series distribution of the variables is unknown, Eruhimov *et al.* [35] and Li *et al.* [104] proposed a supervised learning method, which can be expressed as

$$t = g(\mathbf{x}_1, \cdots, \mathbf{x}_p), \tag{2.7}$$

where $g(\cdot)$ is the supervised learning function, $\mathbf{x}_1, \cdots, \mathbf{x}_p$ are the p process variables and t is the model output, which represents the estimate of the time of the change point.

Fourier Series Decomposition and Discrete Wavelet Transform

Fourier analysis is a typical signal analysis method. Rietman *et al.* [133] showed that Fourier series decomposition of etch process variables (RF power, gas flow rate, *etc.*) is effective for etch endpoint detection. The authors reported that plotting the Fourier components under different coordinates can give different curve shapes, which can help in the recognition of the shape of the endpoint signature. Kim and Choi [83] proposed using Discrete Wavelet Transform (DWT) to analyze the plasma impedance match data. The results show that DWT can effectively detect the signal variations and helps to recognize process abnormalities.

Other Statistical Techniques

More recently, many advanced statistical techniques drawn from the data mining domain have been applied to advanced process control in semiconductor manufacturing. Cheery and Qin [21] proposed that fisher discriminant analysis can be used to differentiate sensor data from different tools and chambers and is shown to be useful for identifying process faults.

Non-Negative Matrix Factorisation (NMF) was employed by Ragnoli *et al* [127] on a case study involving optical emission spectroscopy data from a plasma etch process. By comparison analysis with PCA, the properties of NMF have been highlighted. Forward selection component analysis has been proposed in [126] for the analysis of OES data and found to be more effective for feature selection due to the selection of fewer OES lines to summarise key variations in the process data.

2.2.2 Supervised Feature Extraction Approaches

Supervised feature extraction refers to the implementation of feature extraction algorithms which use a *priori* information about how the features related to desired targets to enhance the selection process. This *priori* information can be for example which data corresponds to normal and abnormal operations, target responses that the features to be selected can be used to predict, or simply knowledge of the underlying systems and characteristics of the features that are of interest, *e.g.* a well defined change point. In this section, methods drawn from the supervised statistics and artificial neural networks are reviewed. These methods are employed either to directly implement feature extraction or to build up models of key features which are subsequently used for process control.

Experience-based Single Variable Selection

The experience-based single variable selection is the simplest method of achieving supervised feature extraction and hence, has been widely employed in the early application of plasma diagnostic data for etching process control. Selection of the single variable, which contains the process signature features, such as etch end point and process varia-

tion, is determined by the engineers having a detailed knowledge of underlying process chemistry and dynamics. In this subsection, the discussion of single variable selection is categorised based on the different sensor techniques used for collecting the data.

Optical Emission Spectroscopy

Single wavelength OES measurement has seen great success in detecting the etch endpoint and faults for various plasma etch processes. The authors in [25], for example, reported that the optical emission of a plasma at wavelengths of 405nm, 520nm or 706nm can be used to spot the etch endpoint for SF_6 etching of Nitride in an Oxide film stack. They showed the effectiveness of these variables for detecting end point for different plasma etchers. In an another experiment, Manos and Flamm [113] reported that 297.7nm, 483.5nm, 519.5nm (CO), 308.9nm (OH), 615.5nm (O) and 656.6nm (H) can be used to monitor the Oxygen etching of photoresist and that 279nm (CCl) is effective for monitoring Chlorine etching of photoresist.

A summary of individual wavelengths that have been identified for endpoint detection for different etch chemicals is given in Table 2.1. The data is compiled from information provided in [48], [139] and [142]. This summary attempts to provide a collection of the single OES wavelengths used for detecting endpoint. However, one has to be aware that in practice, the effectiveness of selected single variables should always be verified, because any tiny variations in the etch conditions (*e.g.* pressure, temperature, power supply, gas flow rate *etc.*) could greatly change the plasma performance and eventually change the optical emissions. As such, it is strongly suggested that Table 2.1 is used a reference rather than the final solution for single variable selection.

Successful application of selected wavelengths for endpoint detection has been reported in [159], [129], [6], [173] and [182], with simple algebraic operations such as addition, subtraction, multiplication and division used to improve the signal strength of selected single variables. However, according to Yue *et al.* [185], single wavelength endpoint detection fails for etch open areas under 0.5%.

Etched Film	Etchant Species	Species Monitored	Wavelength(nm)
Polysilicon	C_xF_y, SF_6, NF_3	F	685.4, 703.7, 712.8
		SiF	777
		Si	288.2
		S	469.5
Polysilicon	Cl_2, HBr	Cl	725.6, 741.4
		H	486.1, 656.5
		Br	827.5
		SiCl	287.1
		$SiH_xBr_yCl_z$	300-350
		Si	288.2
Silicon Dioxide	C_xF_y/O_2	CO	292.5, 302.8, 313.8 325.3, 483.5, 519.8
		O	777.2, 844.7
		CF_2	251.9
		SiF	777
Silicon Nitride	O_2, C_xF_y	F	703.7
		CN	387.1
		N	674
		N_2	315.9, 337.1
		O	777.2, 844.7
Aluminum/Cu	CCl_4	AlCl	261.4
		Al	308.2, 309.3, 396.1
		Cu	325
Aluminum	$SiCl_4/Cl_2/BCl_3$	AlCl	261band, 522band
Si	F_2	F	704
photoresist	O_2	CO	297.7, 483.5, 519.5
		OH	308.9
		O	615.5
		H	656.6
	Cl_2	CCl	279

Table 2.1: Monitored single OES wavelengths for endpoint detection for various substrate film/etchant species combinations.

Plasma Impedance Monitor

The feasibility of using PIM for endpoint detection was demonstrated in [123] for SF_6 reactive ion etching of polysilicon and Si_3N_4. According to Malone *et al.* [112], PIM can provide more robust measurements than OES, especially in cases where the etch open area is under 0.3%. When the number of measured PIM signals is small, say 15 as an example, the simplest method of achieving signal selection, as proposed in [3], is to visualise the signal patterns for each individual signal. Yang *et al.* [181] reported that the plasma impedance can have more obvious changes around the etch endpoint than direct PIM measurements and hence, is more effective for endpoint detection. Dewan *et al.* [30, 29] proposed the plasma impedance can be determined as a function of RF power, chamber pressure and gas flow rate, key parameters for generating a plasma.

Process State Monitor

Process State Monitor (PSM) is used to monitor the changes of process states that engineers or researchers are concerned about, so PSM variables can be diversified for different experiments. When Chang *et al.* [16] investigated the performance of PSM measurements, they found that the direct current (DC) signal had superior performance to univariate OES measurements, with an order of magnitude improvement in the signal-to-noise ratio. Single PSM variables has been used to capture features for endpoint and process variations. Fortunato [39], as an example, reported that reflected power supply signal contained a feature that can be used for detecting etch endpoint, given a polysilicon and silicon nitride stack etch. Roland *et al* [136] proposed using the measurement of chamber pressure to detect etch variations, which can also be achieved via the measurements of wafer pad temperature [168] and cross resistor voltage [66].

Supervised Statistical Techniques

Supervised statistical techniques, such as Partial Least Squares (PLS) and principal component regression (PCR) are also widely employed for dimensionality reduction, with the attempt to extract multi-dimensional relationship between inputs and outputs. The feasibility of applying PCA, PCR and PLS to a semiconductor etch process is discussed in [169]. The overview of the effectiveness of PCA, PCR and PLS for fault

detection and diagnosis is presented in [92].

Applying Hotelling's T^2 for high-dimensional data analysis has received increased attention in multivariate statistical process control [93]. Data with normal and abnormal process operations is required as *priori* information for calculating the control threshold. White *et al.* [171], for example, proposed using the Hotelling's T^2 in conjunction with PCA and the Q-statistic to improve the sensitivity of endpoint detection at an extremely low etch open area (1%). In this algorithm, the Q-statistic is used to investigate the effectiveness of a PCA model (by analyzing the residual data). As a result, the first three PCs are justified for reconstructing the original OES data set and the corresponding Hotelling's T^2 values for each of the PC scores. The usefulness of Hotelling's T^2 for detecting outliers in different operations for batch processes is presented in [115]. Methods for improving the robustness of Hotelling's T^2 for fault detection are proposed in [150]. Joint use of Hotelling's T^2 and Q-statistics is proposed by Yue and Qin [184] as the so-called combined index method, as a means of extracting useful information from historical fault data and helping to improve the prediction accuracy for fault detection.

Recently, Forward Selection Regression (FSR) [126] has been proposed as a competitor to PCA and PLS for identifying key features in OES data. Whereas PCA and PLS employ linear combinations of all input variables, FSR attempts to use only a few variables to capture the observed variation. Hence, FSR is more effective than PCA/PLS for parsimonious feature selection. Experimental results in [126] show that with a comparable number of components/variables involved, FSR can give better performance than PCR/PLS when predicting etch rate using OES data.

The joint use of PCA and other available *priori* information has led to the development of a series of different algorithms for achieving statistical process control. Weighted PCA (WPCA) is proposed by Yue *et al.* [188] to improve the long-term validity of a PCA model. Two different forms of WPCA have been proposed: sample-wise and variable-wise WPCA. Sample-wise WPCA is used to address issues with model updating by adapting models with process changes and variable-wise WPCA is used to

incorporate engineers' experience and knowledge about processes and sensors. PCA models built in these ways require less maintenance and result in better fault detection and classification performance. In [183], the authors reported that WPCA was also effective for selecting the key wavelengths.

Modified Principal component analysis (PCA) was proposed by Han *et al.* [53] for real-time endpoint detection of small open area SiO_2 plasma etching. As a contrast to the regular way of using PCA, the model loadings are obtained from training data sets, while the model scores are computed based on the real-time data set. The modified PCA method is shown to be effective in detecting endpoint under 0.4%-0.8% open area.

Multiblock PCA [22] has been proposed as a method for identifying the subset of variables in etch process fault detection applications. The original variables are separated into a set of subset blocks for further analysis, but the algorithm effectiveness greatly relies on the availability of the *prior* knowledge about variable separation and hence is not suitable for cases where such information is not available.

Recursive PCA (RPCA) [105] has been recently proposed to tackle the time-varying behavior of semiconductor manufacturing processes due to equipment aging, sensor and process drifts, preventive maintenance, and cleaning. The behavior, which is considered part of normal process operation, is often reported as process faults when using a static PCA model. RPCA is proposed to compensate for normal changes and has been shown to be more effective for detecting process faults.

Artificial Neural Networks

ANNs have great flexibility in synthesizing nonlinear relationships from process data. Even relatively simple ANNs, such as feed forward preceptron neural networks with one hidden layer can approximate any continuous function [41].

Applications of nonlinear ANNs have been seen in time-series feature extraction and multivariate data projection. The authors in [114] gave a discussion of the effectiveness

of different nonlinear ANNs for feature extraction from data sets with various features. In plasma etch applications, ANNs have been widely used in the prediction of etch rate, uniformity, fault detection and classification, end point detection, *etc.*

The work of Kim [86, 87, 82, 84] has demonstrated that etch rate can be determined from manipulated inputs, such as gas flows, power, pressure and bias, using ANNs. Building up the models using off-line measurements and subsequently using them for real-time predictions, 5-7% prediction errors are reported [86, 87, 82]. Himmel and May [59] reported that ANN model using the PSM variables (RF power, pressure, electrode gap, CCl_4, He flow rate and O_2 flow rate) as model inputs can provide more accurate prediction of etch rate than manual operations. However, Lee and Spanos [101] reported that no improvement on etch rate prediction can be distinguished when comparing ANNs to a variety of statistical techniques such as PCA, PLS and least squares regression.

Uniformity is referred to as a measure of the spatial variation in etch across the wafer. Kim *et al.* [86] proposed using RF power, pressure and CF_3 flow rate as ANN inputs to predict the uniformity of oxide via etching in a CHF_3/CF_4 plasma and found that RF power is the key input. In addition, faster etch is found to lead to less etch uniformity. Kim and Kim [85] found that the addition of DC bias as an input only served to reduce the accuracy and increase the model complexity. Kim *et al* [84] proposed using an ANN model to predict the discrepancy in the sidewall bottom etch rate with respect to the center etch rate, namely DSE. A neural network model was successfully constructed to model the etching characteristics of DSE and the experiments show that a uniform surface etching can be achieved, using the proposed ANN model. In addition, the research also found that a large DC bias can produce a smaller DSE.

Fault detection and classification for plasma etch is a significant application area. Time series neural network (TSNN) models were employed by Hong and May [62] to achieve malfunction diagnosis of reactive ion etching. Employing two types of *in situ* measurements: OES and residual gas analysis, the TSNN models have been shown to be effective in achieving malfunction diagnosis with only a single missed alarm and a single

false alarm occurring over 21 test runs. Similarly, the TSNN models are used to predict the manipulated inputs (RF power, pressure and two gas flows) based on the variations in seven OES lines, six atomic mass signals from a residual gas analyser and the sample time index [61]. The sample time index is used to measure the aging of the chamber and hence, an indicator of process drifts due to chamber residue build up. The system demonstrated a sensitivity to performance deviations down to 10%.

ANNs have also been shown to be effective in addressing the issues of endpoint detection [118]. Rietman *et al.* [132] showed that a back propagation neural network with a structure of 24 inputs, 5 neurons and 1 output can provide robust detection of etch rate of oxide etching in the polysilicon etch chamber. The ability to predict final film thickness measurements from manipulated variables has been presented with more precision than standard processing techniques [131]. Moreover, the DC bias was found to be an important input variable affecting the resulting prediction accuracy. When we assume that more input variables can help to improve model effectiveness, Allen *et al.* [2] pointed out that the number of inputs are not a key factor in deciding the model prediction effectiveness. By employing four OES variables, the proposed ANN model gave an effective prediction of the etch endpoint.

When using full spectral OES data for ANN modeling, PCA is often performed as data preprocessing step to reduce the data dimensions. Hong *et al.* [64] compared PCA and ANNs for feature extraction from OES data and reported that a 226 wavelength (chosen from the 2048) based ANN returned 7 features, while PCA returned 5, with both giving prediction errors as low as 0.2%. Methods for selecting key variables as ANN inputs vary with the choice of sensor data. Maynard *et al.* [118], for example, proposed using a mean-variance-ratio based method for selecting the key variables from PSM data, where the ratio is defined as:

$$\frac{\overline{\triangle x_i}}{\sigma_i} \tag{2.8}$$

where i is the index of the signals. $\overline{\triangle x_i}$ and σ_i^2 can be expressed as:

$$\overline{\triangle x_i} = \frac{1}{N}\sum_{j=1}^{N}(x_{ij}^{ep} - x_{ij}^{t}) \tag{2.9}$$

and

$$\sigma_i^2 = \frac{1}{N-1}\sum_{j=1}^{N}(x_{ij}^{ep} - x_{ij}^{t} - \overline{\triangle\ x_i})^2 \tag{2.10}$$

for N wafers and j is the index of the wafer number, x_{ij}^{ep} denotes the signal at endpoint and x_{ij}^{t} denote the signal before endpoint. As a result, signals with large mean-variance-ratio can be taken as sensitive signals *w.r.t.* endpoint and therefore, useful as model inputs. Lin *et al* [107] showed the effectiveness of using stepwise regression to select the key variables from a variety of sensor variables. Stepwise regression is also reported useful for selecting the variables as inputs for multi-layer perceptron and radial basis ANN models for chemical vapor deposition processes [38].

Closely-related to ANNs, support vector machine (SVM) has recently become a popular tool in time series forecasting. In plasma etching, SVM has seen successfully used to detect endpoint with OES data, with PCA as a feature extraction method for reducing the number of SVM inputs. Han *et al.* [51] reported that a PCA based SVM method is effective for endpoint detection for BCl_2/Cl_2 etching of Al-Cu alloy stack. In [13], a variety of methods such as PCA, kernel PCA and ICA have been compared for key feature extraction. Results demonstrated that kernel PCA and ICA are better than PCA and the joint use of feature extraction with SVM shows better prediction than directly using SVM on the tested data sets. Sarmiento *et al* [137] proposed using one-class SVMs to detect faults in a reactive ion etching system using optical emission spectroscopy data. Results demonstrated that using normal operation data to train the the one-class SVM, the model can provide a 100% detection of the process faults occurring in their experiments.

While the supervised methods discussed above are promising, a key requirement is the availability of metrology data, *i.e.* measurements of the targets that are of interest. When such metrology data is not available, as is often the case due to cost and time involved, the only option is unsupervised feature selection, which is the focus of this thesis.

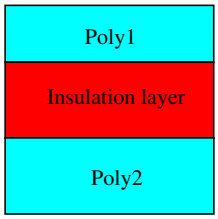


Figure 2.5: Distribution of the polysilicon layers in the etching stack (Poly1 = The first Polysilicon layer; Poly2 = The second Polysilicon layer).

2.3 Experimental Benchmark Data Sets

To estimate the effectiveness of the various unsupervised feature extraction algorithms investigated in this thesis, a number of benchmark data sets are used, namely, three OES data sets collected from plasma etch processes at a semiconductor manufacturing factory and one simulated data set. These are referred to as IDS1, IDS2, IDS3 and SDS1, respectively. Filtered and pre-processed versions of IDS1 and IDS2 are also used as benchmark data sets and these are denoted as IDS1Filt and IDS2Filt, respectively.

2.3.1 Industrial Data Sets

IDS1 and IDS2

Silicon layer etch data were collected for a Hitachi ECR etch system used to etch the poly-silicon layer on a 300mm wafer. As shown in Fig. 2.5, the etching stack contains two polysilicon layers. Optical emission measurements for Poly1 and Poly 2 were collected using an Ocean Optics USB2000 spectrometer connected to the process exhaust. The generated data sets are referred to as IDS1 and IDS2, respectively.

The primary etchants used to etch the first polysilicon layer (Poly1) are HBr, Cl_2 and O_2. The etching step takes 70 seconds, generating an OES data set having dimensions of 91×2048, namely IDS1. To etch the second polysilicon layer (Poly2), HBr and O_2 are used. The resulting etch step takes 90 seconds, generating the OES data set having dimensions of 101×2048, namely IDS2. IDS1 is known to contain an end-point signature while IDS2 does not.

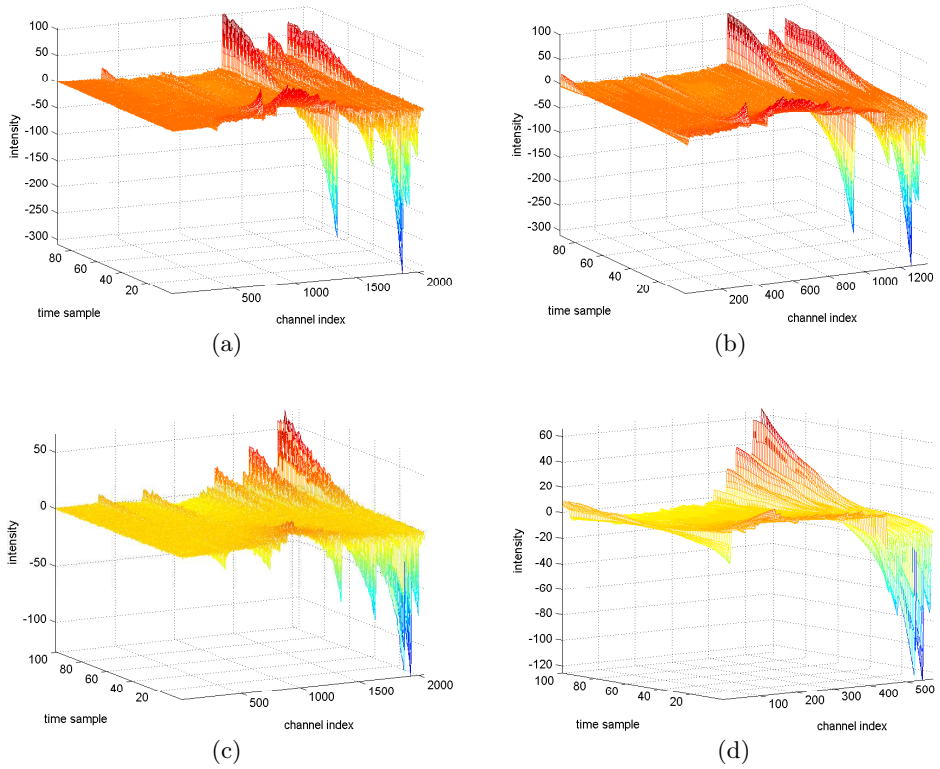


Figure 2.6: Data set visualisation (data mean-centered): (a) IDS1; (b) IDS1Filt; (c) IDS2; (d) IDS2Filt.

Methods for data preprocessing, filtering and noise removal are discussed in detail in Section 3.7.8 and Section 3.7.10, resulting in the generation of cleaned-up data IDS1 and IDS2, namely IDS1Filt and IDS2Filt, respectively. IDS1Filt has dimensions of 91×1354 and IDS2Filt has dimensions of 101×572. Visualization of the four benchmark data sets, IDS1, IDS1Filt, IDS2 and IDS2Filt is provided in Fig. 2.6.

IDS3

Metal layer etch data were collected for a Hitachi ECR etch system used to etch an Aluminium alloy on a 300mm wafer. Transistors are the most basic units in a semiconductor chip and metals are layered to connect these transistors to achieve the desired functionality. The first metal layer (M1), the Aluminium-Copper alloy stack, is used to connect the chip and the external circuit. Proper aligning M1 to the substrate layers is important to avoid faulty functioning of a chip. A typical M1 stack consists of 5 layers,

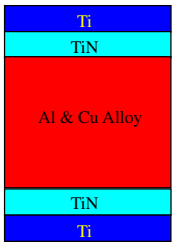


Figure 2.7: Components in a M1 stack.

Titanium (Ti), Titanium Nitride (TiN), Aluminium-Copper (Al-Cu) alloy, TiN and Ti layers as shown in Fig. 2.7.

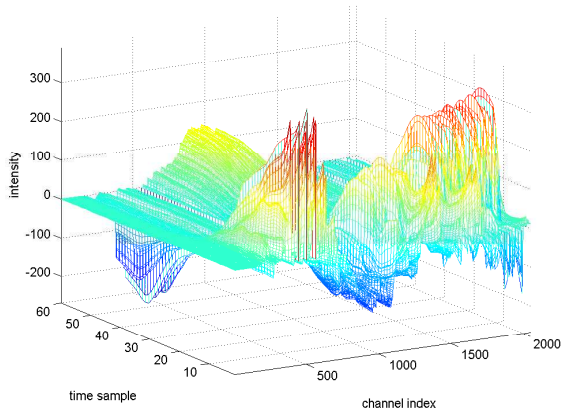
The primary etch gases used in the M1 etching were Boron Chlorine (BCl_3) and Chlorine (Cl_2). To protect the side walls of the M1 stack from being etched, Oxygen (O_2) is used. Chlorine is Aluminium active and reacts spontaneously on contact with Aluminium, generating etch by-product $AlCl_3$ as

$$2Al + 3Cl_2 -> 2AlCl_3. \tag{2.11}$$

$AlCl_3$ is the volatile product. The reaction between Aluminum and Oxygen occurs, generating a surface Oxide (Al_2O_3) as

$$4Al + 3O_2 -> 2Al_2O_3. \tag{2.12}$$

Under the protection of a surface oxide (Al_2O_3), Chlorine cannot etch the Aluminium. Hence, it is necessary to add BCl_3. BCl_3 is a known scavenger of O_2 and reacts actively with any Oxide, leaving a Boron Oxide film on the newly formed side walls of the M1 stack, protecting them from the mechanical etching and thereby improving the isotropy of the etch. The released Aluminium can be etched away by Chlorine. The addition of BCl_3 also generates large Boron ions which enhances the mechanical etching. Helium (He), the cooling gas, is pumped to the backside of the wafer to control the wafer temperature. Effective heat transfer across the wafer is critical to guaranteeing etching uniformity across the wafer surface.



(a)

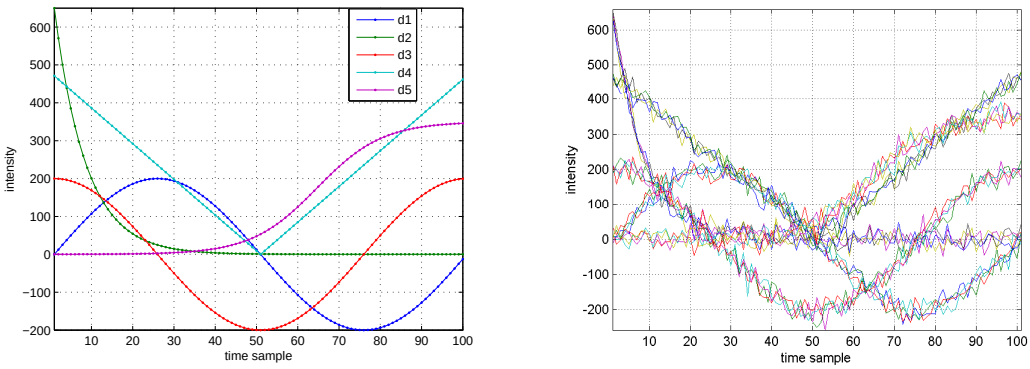
Figure 2.8: A plasma etch OES data set for a single wafer for IDS3.

Optical emission measurements were collected using an Ocean Optics USB2000 spectrometer connected to the process exhaust. These measurements consisted of 2048 spectral channels over a wavelength range from 175 to 875 nm with a sampling interval of 0.76s. Using this setup OES data was collected for 17 lots of 24 wafers, with each wafer undergoing a two step etch process lasting 45s. The resulting data set is referred to as IDS3. A sample data set for a single wafer is shown in Fig. 2.8.

2.3.2 Simulated Data Set

The simulated data set is generated to imitate OES data (having patterns in the time domain) while using a small number of variables. The simulated data set is used in the thesis to illustrate the properties of the algorithms investigated. The simulated data is constructed from 5 signals defined as follows:

$$\begin{aligned}
\mathbf{d}_1 &= 200 \times \sin(\mathbf{x}) \\
\mathbf{d}_2 &= 650 \times (\tfrac{1}{8})^{\mathbf{x}} \\
\mathbf{d}_3 &= 200 \times \cos(\mathbf{x}) \\
\mathbf{d}_4 &= 150 \times |\mathbf{x} - \pi| \\
\mathbf{d}_5 &= 200 \times (1 - \tfrac{1}{1+e^{(-8+2\mathbf{x})}})
\end{aligned} \tag{2.13}$$



(a) (b)

Figure 2.9: (a) Plot of the 5 noise free signals, $\mathbf{d}_1$ to $\mathbf{d}_5$, used to generate SDS1; (b) Plot of the complete set of noisy SDS1 features.

where the variates, $\mathbf{x}(\in \mathbb{R}^{1\times 100})$, are ranged from 0 to 2π with a sampling interval of $\pi/50$. Within the given range of $\mathbf{x}$, the features defined by $\mathbf{d}_3$ and $\mathbf{d}_4$ are similar, so in effect there are only 4 distinctive features contained in the data (Fig. 6.5 (a)).

To generate 20 signals, each of the 5 signals is repeated 4 times with different noise realisations. The data set is generated as

$$\begin{aligned}
\mathbf{z}_i &= \alpha \times \mathbf{d}_1 + (1-\alpha) \times \mathbf{d}_2 + 20 \times \mathbf{e}_i^1, \ i = 1,2,3,4, \\
\mathbf{z}_i &= (1-\alpha) \times \mathbf{d}_1 + \alpha \times \mathbf{d}_2 + 20 \times \mathbf{e}_i^2, \ i = 5,5,7,8, \\
\mathbf{z}_i &= \mathbf{d}_3 + 20 \times \mathbf{e}_i^3, \ i = 9,10,11,12, \\
\mathbf{z}_i &= \mathbf{d}_4 + 20 \times \mathbf{e}_i^4, \ i = 13,14,15,16, \\
\mathbf{z}_i &= \mathbf{d}_5 + 20 \times \mathbf{e}_i^5, \ i = 17,18,19,20,
\end{aligned} \tag{2.14}$$

where $\mathbf{e}_i^j, (j = 1,2,3,4,5)$ are independent identically distributed noise sequences, drawn from a normal distribution function $N(m,v)$ with mean, $m = 0$, and variance, $v = 1$. α is a tuning parameter used for adjusting the similarity between the first eight objects. When $\alpha = 0.5$, without counting the effect of noise, all the first eight objects are identical. The data set, $\mathbf{Z} \in \mathbb{R}^{20\times 100}$ with $\alpha = 0$ is referred to as SDS1. The features contained in SDS1 are displayed in Fig. 6.5 (b).

2.4 Conclusions

In this chapter, an introduction to plasma and plasma etching techniques and process diagnostic devices has been given to provide an in-depth understanding of the technical background of this research. Extensive literature on the feature extraction techniques applied to plasma etching has been reviewed, which highlights the fact that due to the lack of process metrology data, only unsupervised feature extraction techniques are selected as the research focus of this thesis. Experimental benchmark data sets have been introduced to estimate the effectiveness of the various unsupervised feature extraction algorithms investigated in this thesis.

Chapter 3

Principal Component Analysis

3.1 Introduction

The technique of principal component analysis (PCA) was first proposed by Pearson in 1901 [124] and Hotelling in 1933 [65]. Pearson's and Hotelling's papers adopted two different approaches. Pearson's approach focuses on seeking the best-fit straight line or plane to represent the points in a p-dimensional space. In this framework, PCA is equivalent to a geometric optimization problem [78]. Hotelling's approach was concerned with finding a smaller 'fundamental set of independent variables' that can be used to determine the values of the original p variables. The resulting variables are referred to as 'components'. Hotelling chose his 'components' to maximise the total variance of the original variables explained by the components, leading to a singular value decomposition problem.

However, both these approaches require considerable computing power. Before the era of the computer, it was not feasible to do PCA for more than four variables [78]. Nowadays, PCA is widely used in areas such as agriculture, biology, chemistry, climatology, demography, ecology, economics, food research, genetics, geology, meteorology, oceanography, psychology, quality control, *etc* [78] for multivariate data analysis, compression and visulisation [67].

OES has been widely used to monitor the chemistry of plasma to achieve different objectives of process control, *e.g.* plasma modeling [20, 63, 128] and etch point detection

[187, 172]. With the OES footprint of each wafer having over 2000 dimensions, direct visualisation and monitoring of variations in the plasma chemistry across wafers and across lots is impractical. Fortunately, optical emission spectra are inherently highly redundant with the result that PCA based methods prove to be effective at achieving substantial data compression without losing valuable information on plasma changes.

Considering the ability of using PCA in summarising the OES data, Toprac *et al* [161] and White *et al* [171] proposed that it is adequate to use the first four PCs, while Han *et. al* [52] proposed that any PC that can explain over 10% of the whole variance should be considered. Yue *et al* [185] proposed that the second and third PC scores are effective for spotting etch end points and the corresponding PC loadings can be used to select the key variables that cause the changes. Other application like Hotelling's T-square calculated based on PC scores and Q-statistics have been widely employed in the control of process variations [94, 95, 110, 171]. Direct visualisation of the 2-D plots of PC scores is also proposed useful in spotting the changes in process states, such as power, different percentage of gas composition [170] and the changes across lots [174].

In this chapter, the application of PCA to the analysis of OES data from plasma etch process is explored. The basic theory of PCA is first introduced, followed by the discussion of numerical solutions. Conventional methods of plotting the PC scores are applied to the OES data with experimental results presented. A novel low cost method for monitoring changes in PC loadings is then proposed. The effectiveness of using the proposed method for exploring the process variations contained in the high-volume OES data is demonstrated. Finally, the noise level in OES data is investigated and an approach developed to estimate the appropriate noise filter bandwidth.

3.2 Basic PCA Theory

Among the numerous books and articles written about PCA, the book by Jolliffe [78] has achieved the most popularity. This book provides the first comprehensive description of the history, existing and potential applications of PCA, theoretically and practically. Another useful work is the toolbox developed by Wise and Gallagher [176],

which has been widely employed as the computer realization of the PCA technique.

3.2.1 Definition

The objective of PCA is to provide a low-dimensional representation of a high dimension data set, while retaining the maximum amount of the variance observed in the original data. Given a data set $\mathbf{X} \in \mathbb{R}^{m \times n}$ (m observations of the n variables) and unit length direction vector $\mathbf{p}_1 \in \mathbb{R}^{n \times 1}$, the projection of $\mathbf{X}$ onto $\mathbf{p}_1$ is given by

$$\mathbf{t}_1 = \mathbf{X}\mathbf{p}_1, \tag{3.1}$$

where $\mathbf{t}_1 \in \mathbb{R}^{m \times 1}$ are the coordinates of each data point on the $\mathbf{p}_1$ axis. If $\mathbf{X}$ is mean centered, then the variance of the data in the direction $\mathbf{p}_1$ can be expressed by

$$var(\mathbf{t}_1) = \frac{\mathbf{t}_1^{\mathrm{T}}\mathbf{t}_1}{m-1} = \frac{\mathbf{p}_1^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})\mathbf{p}_1}{m-1}. \tag{3.2}$$

In PCA the direction $\mathbf{p}_1$ is chosen to maximise $var(\mathbf{t}_1)$, *i.e.*

$$\begin{aligned} \widehat{\mathbf{p}}_1 &= \arg\max_{\mathbf{p}_1} \frac{\mathbf{p}_1^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})\mathbf{p}_1}{m-1}, \text{ s.t. } ||\mathbf{p}_1||_2 = 1 \\ &= \arg\max_{\mathbf{p}_1} \mathbf{p}_1^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})\mathbf{p}_1, \text{ s.t. } ||\mathbf{p}_1||_2 = 1 \\ &= \arg\max_{\mathbf{p}_1} \frac{\mathbf{p}_1^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})\mathbf{p}_1}{\mathbf{p}_1^{\mathrm{T}}\mathbf{p}_1}, \text{ s.t. } ||\mathbf{p}_1||_2 = 1 \end{aligned} \tag{3.3}$$

Note that the expression $\frac{\mathbf{p}_1^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})\mathbf{p}_1}{\mathbf{p}_1^{\mathrm{T}}\mathbf{p}_1}$ is the well known Rayleigh Quotient which is maximised when $\mathbf{p}_1$ is the eigenvector of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ associated with the largest eigenvalue of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. Thus the direction of largest variation in data $\mathbf{X}$ is given by the largest eigenvector of its covariance matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. To calculate the direction with the next largest data variation, the contribution to $\mathbf{X}$ in the direction $\mathbf{p}_1$ needs to be removed, leading to the generation of the residual data matrix $\mathbf{X}_1$,

$$\mathbf{X}_1 = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1^{\mathrm{T}}. \tag{3.4}$$

Correspondingly, the maximum variance of $\mathbf{t}_2$,

$$var(\mathbf{t}_2) = \frac{\mathbf{p}_2^{\mathrm{T}}(\mathbf{X}_1^{\mathrm{T}}\mathbf{X}_1)\mathbf{p}_2}{m-1} \tag{3.5}$$

occurs when $\mathbf{p}_2$ is the eigenvector of $\mathbf{X}_1^{\mathrm{T}}\mathbf{X}_1$ associated with its largest eigenvalue. By repeating the same procedure, additional $\mathbf{p}_i$ can be obtained until the residual matrix

becomes zero. Since the largest eigenvector of $\mathbf{X}_1^{\mathrm{T}}\mathbf{X}_1$ is the same as the second largest eigenvector of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ and so on, it follows that all the eigenvectors of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ can be computed simultaneously and sorted in descending eigenvalue order to give $\mathbf{p}_i$ ranked in order of significance.

Various names are used in the literature to refer to $\mathbf{p}_i$ and $\mathbf{t}_i$ [78]. In this thesis $\mathbf{p}_i$ and $\mathbf{t}_i$ are referred to as principal component loadings and scores, respectively and each pair of $\mathbf{p}_i$ and $\mathbf{t}_i$ is referred to as one principal component (PC).

Given $\mathbf{X}$ with rank r, the PCA decomposition of $\mathbf{X}$ can be expressed as a sum of r matrices with rank 1 [69]:

$$\mathbf{X} = \sum_{i=1}^{r} \mathbf{t}_i \mathbf{p}_i^{\mathrm{T}} = \mathbf{T}\mathbf{P}^{\mathrm{T}}, \tag{3.6}$$

where $\mathbf{P}(\in \mathbb{R}^{n\times r})$ is an orthogonal matrix with columns $\mathbf{p}_i$ defined as:

$$\begin{cases} \mathbf{p}_i^{\mathrm{T}}\mathbf{p}_j = 0, \forall\ i \neq j \\ \mathbf{p}_i^{\mathrm{T}}\mathbf{p}_j = 1, i = j \end{cases} \tag{3.7}$$

and $\mathbf{T}(\in \mathbb{R}^{m\times r}) = [\mathbf{t}_1, \cdots, \mathbf{t}_r]$. It follows that

$$\mathbf{t}_i = \mathbf{X}\mathbf{p}_i \text{ and } \mathbf{T} = \mathbf{X}\mathbf{P}. \tag{3.8}$$

PCA can be viewed as projecting data points from the original X-space to a new space spanned by the principal components. The projection between the spaces is defined by $\mathbf{p}_i$. As illustrated in Fig. 3.1 for a 2-D example, the elements of $\mathbf{p}_1$ are the projections of a unit vector along $\mathbf{p}_1$ on the axes of the original X-space, *i.e.* the cosine and sine of angle θ. The perpendicular projection of the data onto the PC direction given by $\mathbf{p}_1$ is expressed in $\mathbf{t}_1$. Thus each element of $\mathbf{t}_1$ corresponds to the new coordinate of the individual point along $\mathbf{p}_1$.

3.2.2 Singular Vector Decomposition

Singular vector decomposition (SVD) is another popular multivariate analysis method, which underpins PCA. When PCA is calculated using the data covariance matrix, SVD provides a computationally efficient and numerically robust method of finding PCs [55].

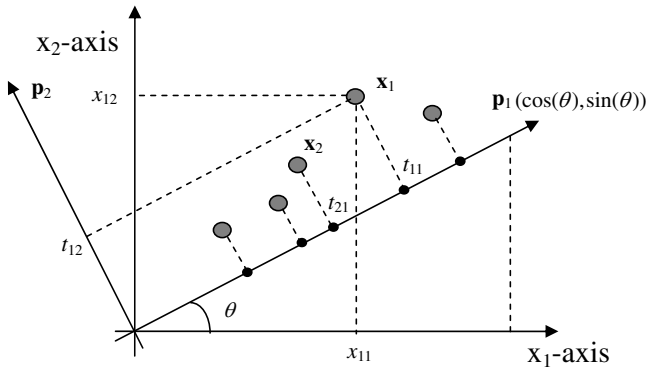


Figure 3.1: Illustrating PCA in a 2-D example

Given a $m \times n$ matrix $\mathbf{X}$, the SVD of $\mathbf{X}$ can be expressed as [121]:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}} \tag{3.9}$$

where $\mathbf{U} \in \mathbb{R}^{m \times r} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_r]$ and $\mathbf{V} \in \mathbb{R}^{n \times r} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_r]$ are the left singular matrix and right singular matrix, respectively, $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$. Σ is a $r \times r$ diagonal matrix with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$ known as the singular values.

Using SVD to compute PCA, the matrix ill-conditioning problem is avoided. The relationship between SVD and PCA is investigated as follows. Consider the covariance matrix $\mathbf{X}^{\mathrm{T}}\mathbf{X}$,

$$\mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{V}\Sigma\mathbf{U}^{\mathrm{T}}\mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}} = \mathbf{V}\Sigma^2\mathbf{V}^{\mathrm{T}}. \tag{3.10}$$

Since $\mathbf{V}$ is orthogonal, Eq. (3.10) can be rewritten as

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{v}_i = \sigma_i^2 \mathbf{v}_i, \tag{3.11}$$

where $\mathbf{v}_i$ is the i^{th} column vector of $\mathbf{V}$ and σ_i is the i^{th} element in the diagonal of Σ. This shows that $\mathbf{v}_i$ is simply an eigenvector of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ and σ_i^2 is the corresponding eigenvalue. Thus the square root of the eigenvalues of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ are the singular values of $\mathbf{X}$ and the column eigenvectors are the right singular vectors of $\mathbf{X}$. Equivalently it can be shown that $\mathbf{u}_i$ can be calculated as the eigenvector of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$, *i.e.*

$$\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{u}_i = \sigma_i^2 \mathbf{u}_i, \tag{3.12}$$

where $\mathbf{u}_i$ is the i^{th} column vector of $\mathbf{U}$. Thus comparing the definition of PCA, $\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathsf{T}}$ with that of the SVD of $\mathbf{X}$, $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{\mathsf{T}}$, it follows that

$$\mathbf{P} = \mathbf{V} \text{ and } \mathbf{T} = \mathbf{U}\Sigma, \tag{3.13}$$

where $\mathbf{P}^{\mathsf{T}}\mathbf{P} = \mathbf{I}_n$, $\mathbf{P}$ are the eigenvectors of $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ and the right singular vectors of $\mathbf{X}$.

3.2.3 Nonlinear Iterative Partial Least Squares

Another popular method for calculating PCs is the so-called nonlinear iterative partial least squares (NIPALS) algorithm [45]. SVD can be used to calculate all r PCs in one step, while NIPALS can be used to calculate them one at a time in order of significance. Thus, when only a few PCs are needed, NIPALS is computationally much more efficient and requires less memory than SVD. The algorithm details can be described as follows [45]:

Step 1: Initialisation. $i = 1$ and $\mathbf{X}$ is mean centred or standardised, depending on the problem.

Step 2: Randomly take a column vector $\mathbf{x}$ ($\in \mathbb{R}^{m\times 1}$) from $\mathbf{X}$ and assign

$$\mathbf{t}_i = \mathbf{x}. \tag{3.14}$$

Step 3: Calculate $\mathbf{p}_i$ ($\in \mathbb{R}^{n\times 1}$) :

$$\mathbf{p}_i = \frac{\mathbf{X}^{\mathsf{T}}\mathbf{t}_i}{\mathbf{t}_i^{\mathsf{T}}\mathbf{t}_i} \tag{3.15}$$

Step 4: Normalize $\mathbf{p}_i$:

$$\mathbf{p}_i = \frac{\mathbf{p}_i}{\|\mathbf{p}_i\|_2} \tag{3.16}$$

Step 5: Calculate $\mathbf{t}_{inew}$:

$$\mathbf{t}_{inew} = \mathbf{X}\mathbf{p}_i \tag{3.17}$$

Step 6: Compare $\mathbf{t}_{inew}$ and $\mathbf{t}_i$.

(a) If they are the same, then store $\mathbf{t}_i$ and $\mathbf{p}_i$ in $\mathbf{T}$ and $\mathbf{P}$, respectively and go to Step 7.

(b) Otherwise, $\mathbf{t}_i = \mathbf{t}_{inew}$ and go back to Step 3.

Step 7: Stop condition. Check if the required number of PCs has been obtained.

(a) If yes, exit the algorithm and return $\mathbf{P}$ and $\mathbf{T}$ as the result.

(b) Otherwise, deflate $\mathbf{X}$ and increasing the PC count,

$$\begin{aligned} \mathbf{X} &= \mathbf{X} - \mathbf{t}_i \mathbf{p}_i^{\mathrm{T}} \\ i &= i + 1. \end{aligned} \tag{3.18}$$

and go back to Step 2.

The operation of NIPALS can be seen by substituting Eq. (3.17) into Eq. (3.15), giving

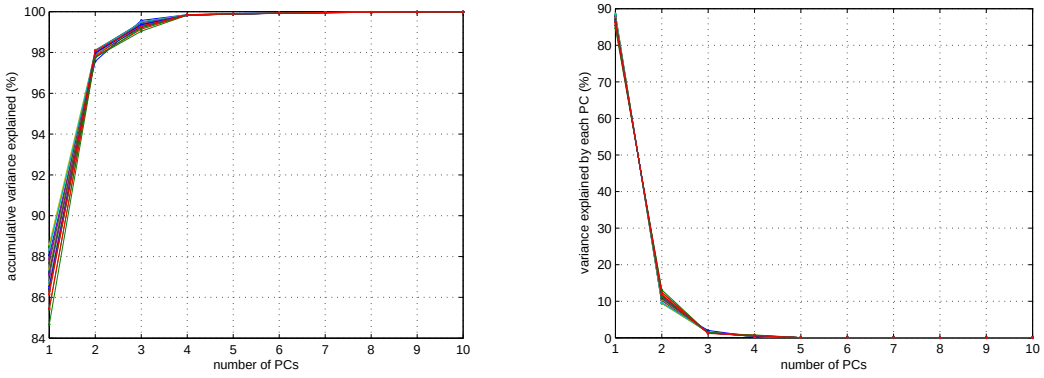
$$\mathbf{p}_i = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{p}_i}{\mathbf{t}_i^{\mathrm{T}}\mathbf{t}_i} = c\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{p}_i, \tag{3.19}$$

where c is a scalar and $\mathbf{p}_i$ are the eigenvector of $\mathbf{X}^{\mathrm{T}}\mathbf{X}$. Equivalently $\mathbf{t}_i$ is the eigenvector of $\mathbf{X}\mathbf{X}^{\mathrm{T}}$. Thus NIPALS and SVD are in essence equivalent in terms of calculating principal components.

3.3 Selecting the Number of PCs

An important issue with PCA is how to choose the number of PCs. No doubt, the more PCs selected, the more information in the data can be retained. However, if the data is corrupted by noise then additional components may have little or no useful information. Meanwhile, the ability of using PCs to summarise data is destroyed.

There are a few typical methods dealing with this issue. One is the so-called cumulative percentage of total variation. Given a preset threshold, say 90% or 95%, the required number of PCs is the smallest value for which the accumulative percentage of variance exceeds the threshold. The scree graph, [15] another commonly used approach, involves looking at a plot of accumulated percentage of variance explained by PCs against the number of PCs and deciding the number of PCs that corresponds to the 'scree' point in the plot. Cross validation [177] can be used to select the minimum number of PCs necessary for adequate prediction. The data points in the original data set are randomly selected and separated into training and test data. The model is built using training data and validated by the test data. Model prediction is measured by the prediction error sum of squares as a function of the number of PCs. Using this



(a) (b)

Figure 3.2: (a) Accumulative variance explained as a function of the number of PCs; (b) Variance explained by each PC.

method, the selected model avoids including noise only PCs.

For OES, the simple method of identifying the scree point of the accumulative percentage of variance plot is normally employed to select the number of PCs. Taking lot 10 of the IDS3 benchmark data set, lot10-IDS3, as an example, the accumulative variance explained as a function of the number of PCs for all 24 wafers is shown in Fig. 3.2 (a) (each line corresponds to one wafer) and the variance explained by each PC is shown in Fig. 3.2 (b). The 'elbow point' in Fig. 3.2 (a) occurs when the number of PCs is equal to 3, corresponding to about 99% variance explained. The other PCs (from the 4^{th} PC onwards) are omitted due to the low level of significance (less than 1% variance explained by each PC as shown in Fig. 3.2 (b)).

3.4 Monitoring PC-Loading Direction

If PCA is performed on the OES data as a whole process, trends can only be observed by monitoring the time evolution of the scores. However, if PCA is applied on a wafer-by-wafer or lot-by-lot basis, very effective monitoring of process variation can be achieved by tracking the changes in the directions of the PC loadings. Changes can be expressed either in terms of the angle difference between vectors or the magnitude of the vector

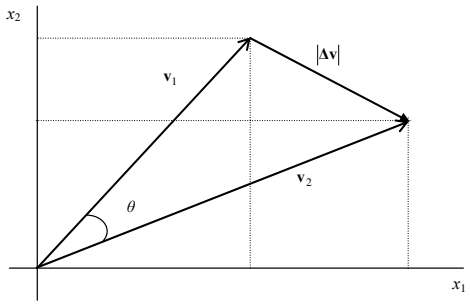


Figure 3.3: Measuring changes in loading vector directions

difference between them [190]. The angle θ (in radians) is given by

$$\theta = arccos(\frac{\mathbf{v}_1\mathbf{v}_2^{\mathrm{T}}}{||\mathbf{v}_1||_2||\mathbf{v}_2||_2}), \tag{3.20}$$

while the magnitude of the vector difference $\triangle\mathbf{v}$ is simply defined as

$$|\triangle\mathbf{v}| = ||\mathbf{v}_1 - \mathbf{v}_2||_1. \tag{3.21}$$

Since, by definition, loading vectors are unit length, it follows that for small θ the two measures are approximately equivalent, *i.e.* $|\triangle\mathbf{v}| \approx \theta$.

As illustrated in Fig. 3.3, $\mathbf{v}_1$ and $\mathbf{v}_2$ denote the two loading vectors, respectively. As a summarization of changes in vector directions, either θ or $|\triangle\mathbf{v}|$ can be used.

3.4.1 Lot-by-lot Analysis

Lot-by-lot analysis refers to the analysis on the variations that takes places across lots. Because we are concerned with tracking process changes over time, the lot data is unfolded along wavelength direction. Details are illustrated in Fig. 3.4.

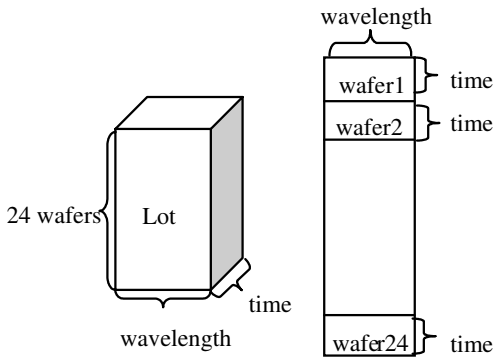


Figure 3.4: Unfolding of the 3-way OES data block along wavelength direction. Each block corresponds to a lot of 24 wafers.

Having unfolded the OES data along wavelength direction, PCA can be performed by treating each lot of 24 wafers as a single data matrix. Mean centering is applied to each lot. We will refer to the resulting PCs as lot-PCs, consisting of lot-PC loadings and lot-PC scores. The variance explained by each lot-PC is plotted as a function of lot number in Fig. 3.5 (a) for the 17 lots in the IDS3 benchmark. Fig. 3.5 (b) shows the accumulated variance explained by the lot-PCs. As can be seen, the first lot-PC captures over 85% of the data variation observed across all 2045 wavelengths and the first three principal components together can explain over 99% of the variance across all lots. Therefore, it is feasible to use the first three lot-PCs to represent the OES data for each lot.

A closer look at Fig. 3.5 (a) shows that a sharp change at lot13-IDS3 occurs for the variance explained by each of the lot-PCs. Analysis of the variation in the direction of lot-PCs across lots, as shown in Fig. 3.6, reveals that the sharp change is linked to the significant change in the orientation of lot-PCs from lot13-IDS3 onwards. Switching the comparison reference from lot1-IDS3 to lot10-IDS3 leads to the results shown in Fig. 3.7, which demonstrate that the sharp change at lot13-IDS3 is not reference-dependent.

Further analysis of the difference in the absolute values of the first PC loadings between lot12-IDS3 and lot13-IDS3 (Fig. 3.8) shows that the channels in the range 200-600 and

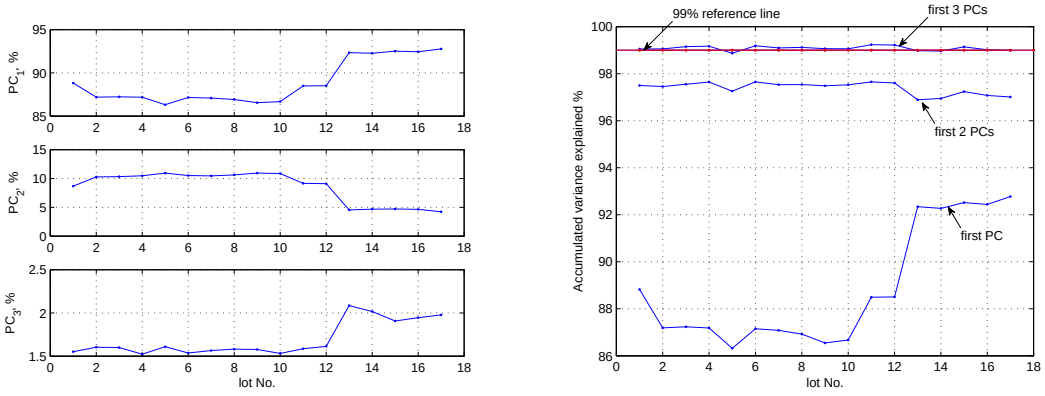
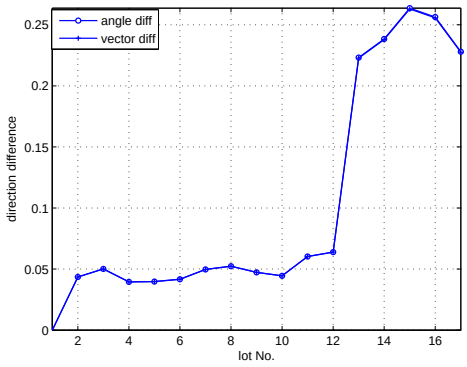
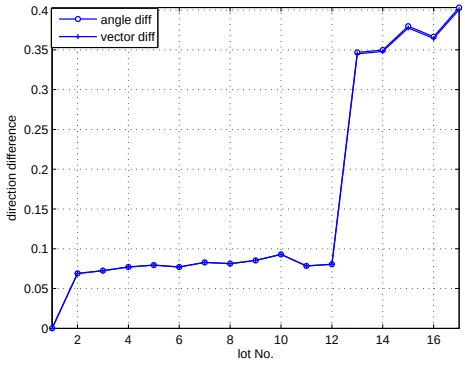


Figure 3.5: (a) Variance explained by each lot-PC; (b) Accumulated variance explained by the lot-PCs.

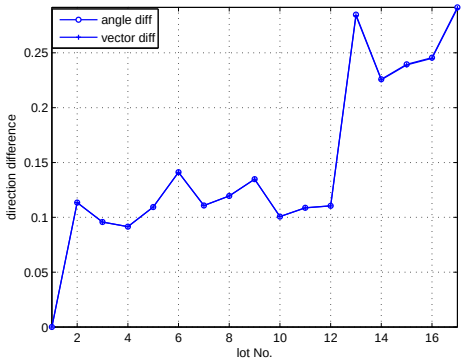
1300-1400 have an increased contribution to the pattern contained in the PC score, while the channels in the range 600-1200 and 1700-2000 have a reduced contribution. Because the number of channels involved is large, it is difficult to establish which channels are the main factors contributing to the difference between lot12-IDS3 and lot13-IDS3, but at least the difference between channels from different areas has been spotted. Following investigation it was determined that the plasma change was as a result of a small drift in the flow rate of a cooling gas applied to the backside of the wafers during etching, a change that was not detected by the existing plasma chamber process monitoring schemes.



(a)

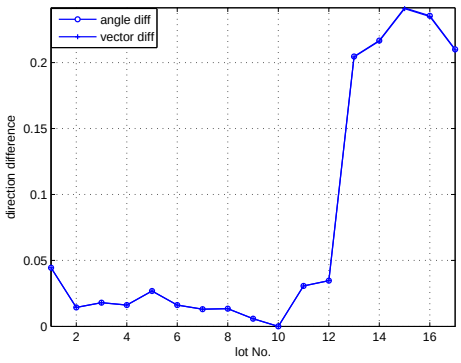


(b)

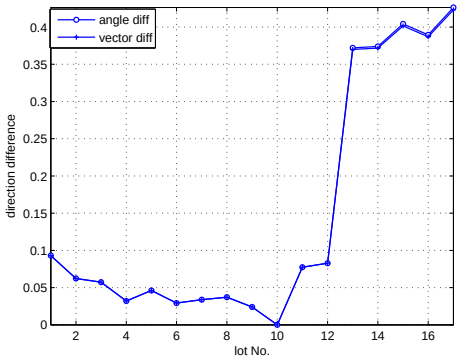


(c)

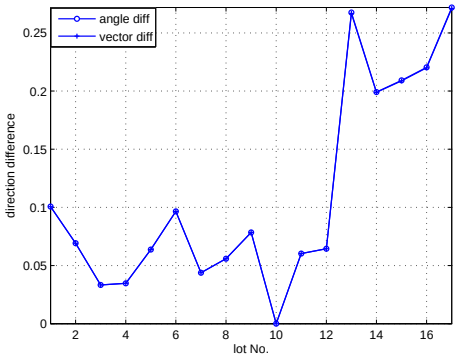
Figure 3.6: Variation in lot-PC (loading) direction across lots (with respect to lot1-IDS3): (a) The first lot-PCs; (b) The second lot-PCs; (c) The third lot-PCs.



(a)

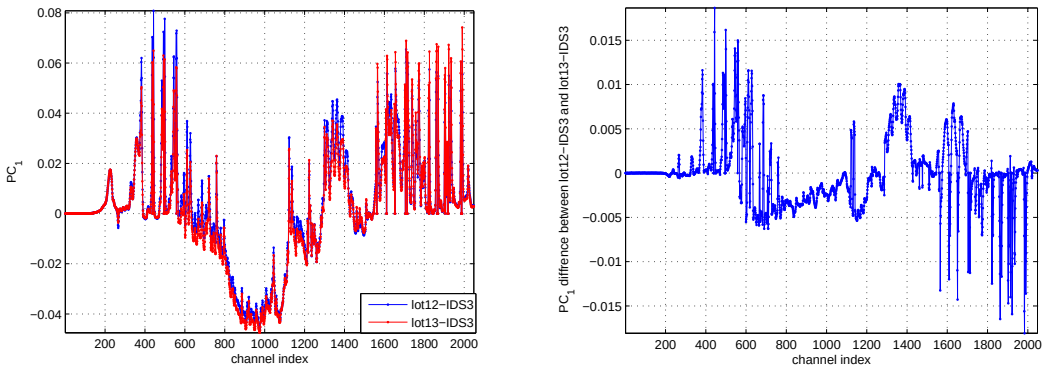


(b)



(c)

Figure 3.7: Variation in lot-PC (loading) direction across lots (with respect to lot10-IDS3): (a) The first lot-PCs; (b) The second lot-PCs; (c) The third lot-PCs.



(a) (b)

Figure 3.8: (a) The first PC loading of lot12-IDS3 and lot13-IDS3; (b) A plot of the difference between the absolute values of these two loadings.

3.4.2 Wafer-by-wafer Analysis

Wafer-by-wafer analysis allows us to explore the variation that takes place across wafers. Here, we simply perform PCA analysis on individual wafer OES data sets and refer to the resulting PCs as wafer-PCs, consisting of wafer-PC loadings and wafer-PC scores.

Comparing the changes in the loading directions, it is necessary to select a reference. Here, the first lot-PC from lot1-IDS3 is chosen. Fig. 3.9 shows the variation in the wafer-PC1 directions over all wafers. The sharp change arises from wafer 267 onwards. A further investigation of the number of wafers contained in each lot (Table 3.1) shows that wafer 267 corresponds to lot13-IDS3. Hence, the plasma change at lot13-IDS3 observed in the lot-PC analysis is clearly present in the wafer-PC analysis as well.

lot index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
No. of wafers	24	24	24	24	7	19	24	24	24	24	24	24	24	24	24	24	18

Table 3.1: Number of wafers contained in each lot.

Another feature can be observed is that large spikes are evident throughout Fig. 3.9 (a). These occur at the first wafer in each lot. This is highlighted in Fig. 3.9 (b) which shows variance over a two lot interval. These sharp changes were attributed to changes

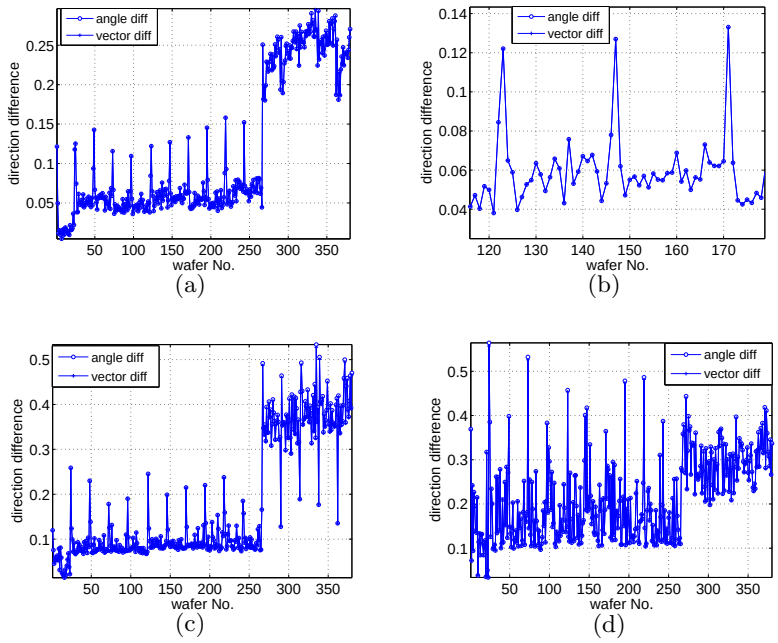


Figure 3.9: Variation in wafer-PC (loading) direction across wafers (with respect to the first lot-PC loading): (a) The first wafer-PCs; (b) Zoomed version of (a) for wafers from 120 to 170; (c) The second wafer-PCs; (d) The third wafer-PCs.

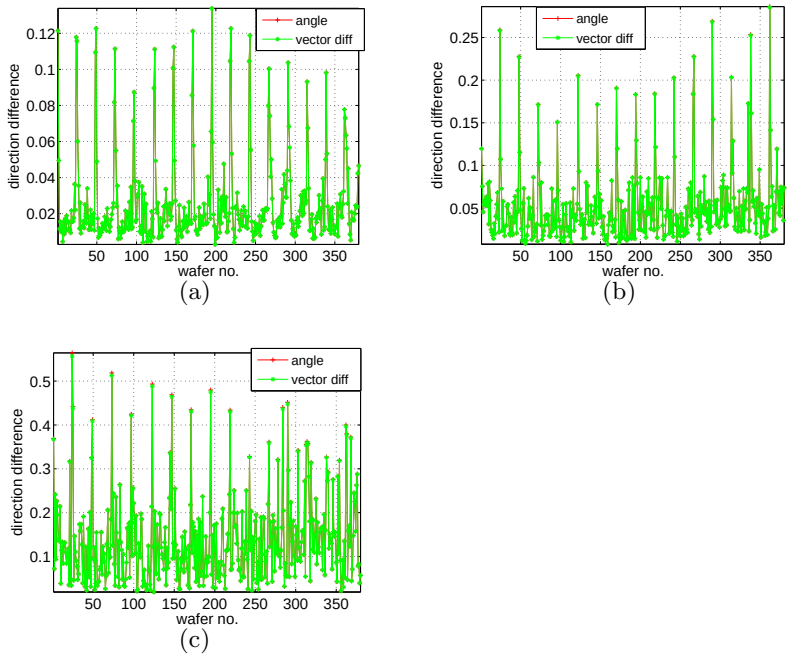


Figure 3.10: Variation in wafer-PC (loading) direction across wafers (with respect to the first lot-PC loading in each lot): (a) The first wafer-PCs; (b) The second wafer-PCs; (c) The third wafer-PCs.

in the absorption characteristics of the plasma chamber wall as a result of a cleaning cycle that is performed between lots. While a dummy etch cycle is performed following each clean cycle to counter this affect, it is clear from Fig. 3.9 (b) that cleaning still has a significant impact on plasma characteristics for the first and to a lesser extent, the second wafer etch of each lot. A closer look at Fig. 3.9 (a) also shows that the spikes occur at the last wafer in lot1-IDS3, lot2-IDS3 and lot10-IDS3. These changes reflect the deterioration in plasma etcher performance due to the accumulation of etch by-products on the chamber wall, which necessitates the use of cleaning cycles in the first instance.

Variation in the second and third wafer-PC directions is shown in Fig. 3.9 (c) and (d), respectively. The across-wafer changes are also evident. To analyse the intra-lot wafer performance, the lot-PCs are selected as the reference for comparing the variation in the wafer-PCs for all intra-lot wafers. As shown in Fig. 3.10, the over-lot changing trends are removed and the large spikes are only evident for the first wafer in each lot.

To investigate which channels mainly contribute to the wafer-to-wafer differences, a comparison analysis of the PC loadings from the outlier and normal wafers is developed. As a comparison benchmark, the average of the first PC loadings for the normal wafers (defined as the fifth to the tenth wafers in each lot) is calculated as:

$$\bar{\mathbf{p}}_1^N = \sum_{i=1}^{14}\sum_{j=5}^{10}(\mathbf{p}_1^{ij})/(14 \times 6), \tag{3.22}$$

where $\bar{\mathbf{p}}_1^N$ denotes the average of the first PC loadings for the normal wafers, $\mathbf{p}_1^{ij}$ denotes the first PC loading for the j^{th} wafer in the i^{th} lot. Note that only the lots having 24 wafers are employed, leading to 14 lots in total. To provide a robust comparison, the average of the first PC loading of the outlier wafers, $\bar{\mathbf{p}}_1^K$, is defined as:

$$\bar{\mathbf{p}}_1^K = \sum_{i=1}^{14}\mathbf{p}_1^K/14, \text{ for } K = 1, 2, 24, \tag{3.23}$$

where $\mathbf{p}_1^K$ denotes the first PC loading of the K^{th} wafer in each lot. This is used as the measure of the outlier wafers, $K = 1, 2$ and 24.

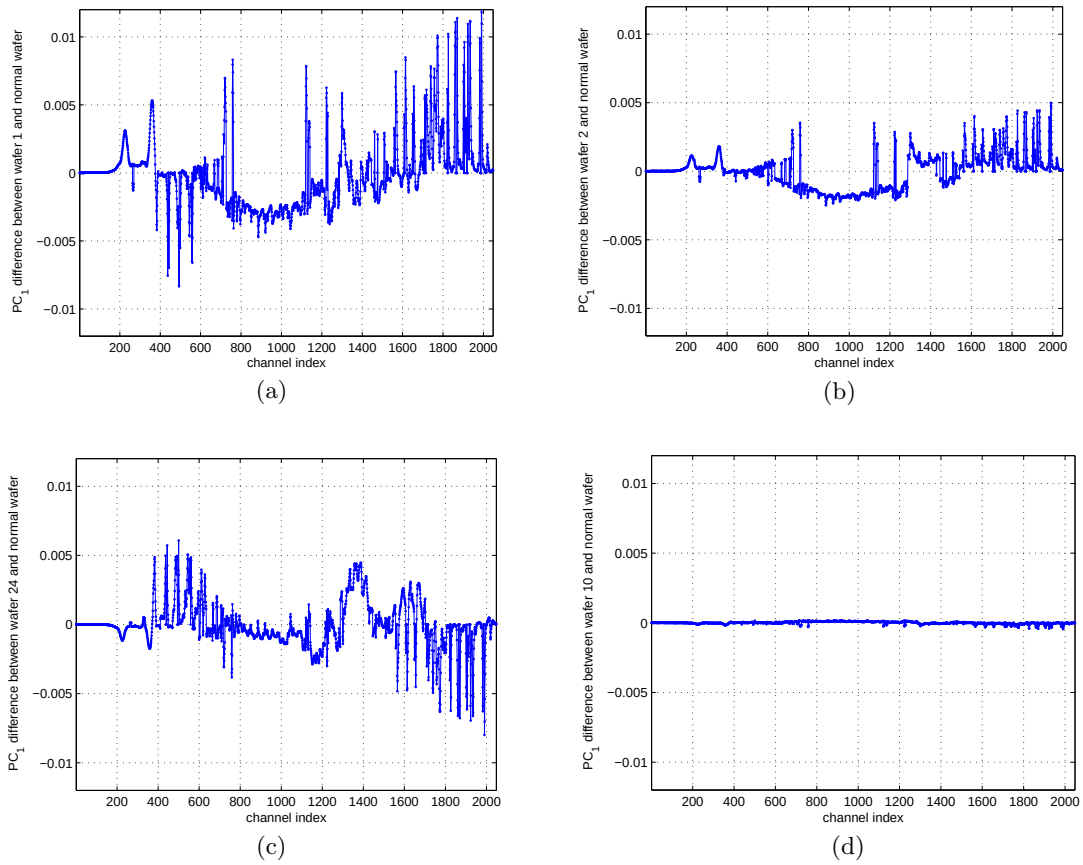


Figure 3.11: Difference in the absolute values of the first PC loading: (a) Between the first and normal wafers; (b) Between the second and normal wafers; (c) Between the last and normal wafers; (d) Between the tenth and normal wafers.

The differences between $\bar{\mathbf{p}}_1^1, \bar{\mathbf{p}}_1^2, \bar{\mathbf{p}}_1^{24}$ and $\bar{\mathbf{p}}_1^N$ are demonstrated in Fig. 3.11 (a), (b) and (c), respectively. A plot of the difference between $\bar{\mathbf{p}}_1^{10}$ and $\bar{\mathbf{p}}_1^N$ is also included in Fig. 3.11 (d) as a control to indicate the normal level of variability between wafers. It is obvious that the channels contributing to the wafer-to-wafer changes are similar for the first two wafers, though the differences are less significant in the second wafer. Comparing Fig. 3.11 (a), (b) and (c) shows that the loading pattern of the last wafer is different from that of either of the first two wafers, indicating that between the start and end of etching of a lot of wafers, significant changes occur in the plasma etch chamber. However, it is difficult to identify the critical variables that cause the differences directly using PCA.

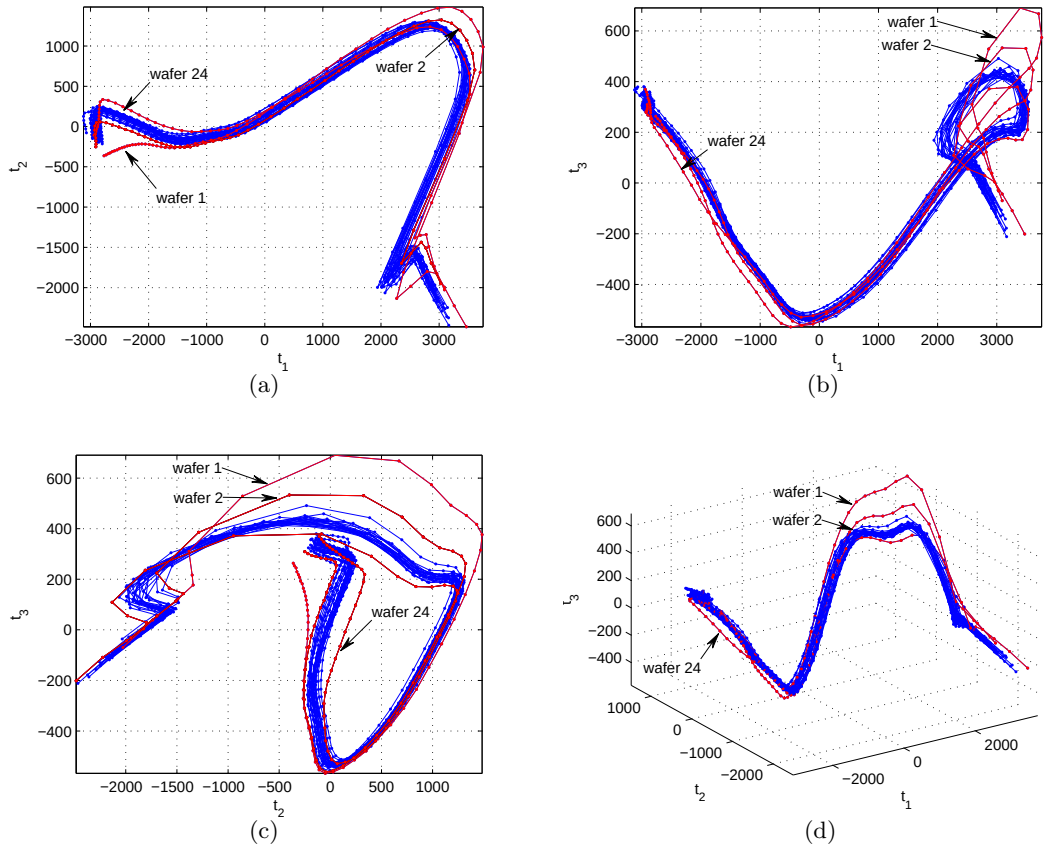


Figure 3.12: The first three PC scores of each wafer in lot10-IDS3: (a) t_1 vs t_2; (b) t_1 vs t_3; (c) t_2 vs t_3; (d) t_1 vs t_2 vs t_3.

3.5 Score Pattern Trends Across Wafers

As an illustration of the data compression and pattern visualisation capabilities of PCA, the score patterns generated by the OES data for all wafers in lot10-IDS3 are investigated. As a reference model, the lot-PC loadings for lot10-IDS3 were used and the PC-scores for each wafer were obtained according to Eq. (3.8). As shown in Fig. 3.12, it is evident that the evolution of the OES data for the first, second and last wafers (highlighted by the red color and labels) is substantially different from the remaining wafers. This trend is detected in each of the 2-D score plots, but the 3-D plot provides the best visualisation of the trends.

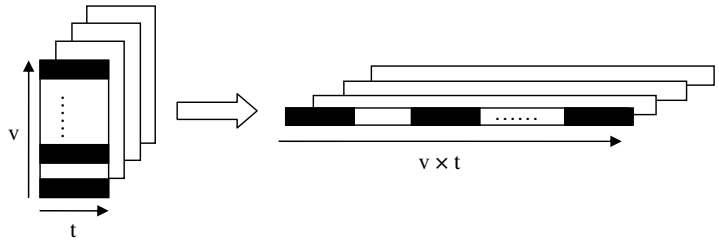


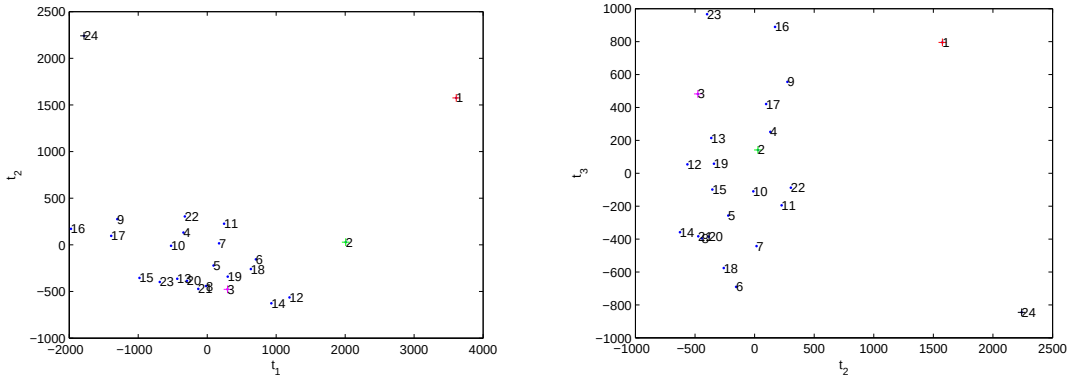
Figure 3.13: Method for unfolding a 2-D OES data matrix into a vector for a batch of wafers (v denotes the wavelengthes and t denotes the time samples).

3.6 Conventional PCA Analysis

The advantage of using time series data (PC scores as shown above) to detect the trends across wafers is that the difference between individual wafers can be highlighted in time. However, when comparisons are required over a large number of wafers, 2-D and 3-D visualisation becomes ambiguous and computationally intensive. An alternative solution is to reduce each time series data to a small number of features, or even to a single point. In this section, methods of achieving single point score representation of wafers using PCA are described. Essentially these methods are designed to transfer the 2-D OES data into 1-D vector format.

3.6.1 Unfolding Two Dimensional OES into One Dimension

One approach for unfolding is to rearrange the 2-D OES data along the time direction. As shown in Fig. 3.13, each wavelength (vertical direction) is measured over a number of time samples (horizontal direction). For unfolding, each wavelength time series is appended to its neighbour, resulting in a single vector with $v \times t$ sample points (v and t denoting the number of wavelengthes and time samples, respectively) for each wafer. Using this approach a lot of w wafers of OES data can be represented as a $w \times (v \times t)$ matrix, (here w = measurements and $v \times t$ = number of variables) allowing direct application of PCA. The obtained data is referred to as the unfolded-1 data.

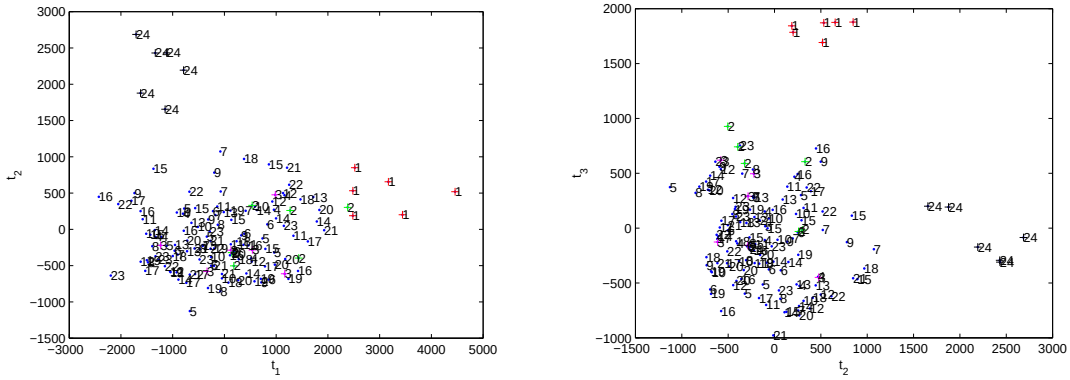


(a) (b)

Figure 3.14: Applying PCA to the unfolded-1 OES data for wafers in lot 10: (a) The first vs the second score; (b) The second vs the third score.

Applying PCA to the unfolded-1 data for lot10-IDS3, the variance captured by each of the first three PCs is 61.27%, 18.22% and 10.33%, respectively (89.8% accumulatively for the first three PCs), justifying the use of the first three PCs to represent the OES data for lot10-IDS3. As shown in Fig. 3.14, the distinctive performance of the first two and last wafers are recognised from the plot of $\mathbf{t}_1$ against $\mathbf{t}_2$, while in the plot of $\mathbf{t}_2$ against $\mathbf{t}_3$, only the first and last wafers show up as outliers.

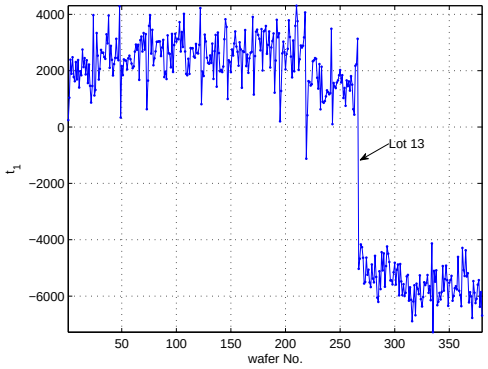
Further investigating the intra-lot wafer patterns, 6 lots (lot7-IDS3 to lot12-IDS3) are examined. PCA analysis on these data shows that 56.56%, 16.55% and 12.15% of the variance are captured by each of the first three PCs, respectively (85.28% accumulatively by the first three PCs). The 2-D plots of the first three PC scores (in Fig. 3.15) show that the performance of the first and last wafers is distinct from the other wafers in the same lot.



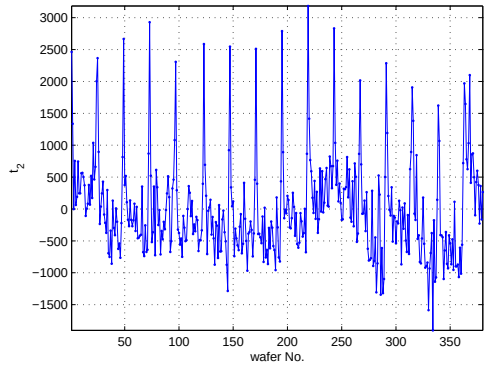
(a) (b)

Figure 3.15: Applying PCA to the unfolded-1 OES data for wafers in 6 lots (from lot7-IDS3 to lot12-IDS3): (a) The first vs the second score; (b) The second vs the third score.

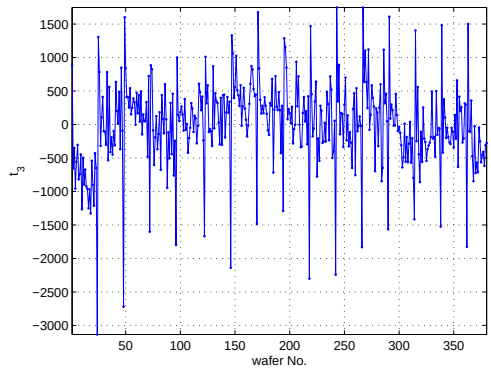
Extending the PCA analysis to all 380 wafers in the IDS3 data set, a high accumulative percentage (94.53%) of variance is captured by the first three PCs (87.99%, 3.79% and 2.76% of the variance by each of the first three PCs, respectively). As can be seen in Fig. 3.16 (a), the sharp change occurs at lot13-IDS3 in the first score, which is consistent with the pattern shown by the wafer-by-wafer monitoring of PC-loading directions, described in section 3.4.2. In Fig. 3.16 (b), the big spike occurs at every first wafer in each lot and in Fig. 3.16 (c), the big spikes occur at every first wafer and to a lesser extent at the last wafer. These patterns are more clearly captured in the 2-D plots of PC scores as shown in Fig. 3.17.



(a)

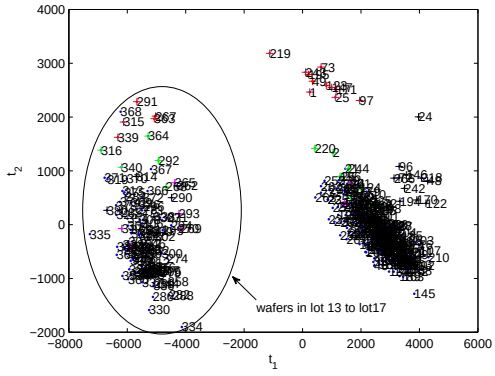


(b)

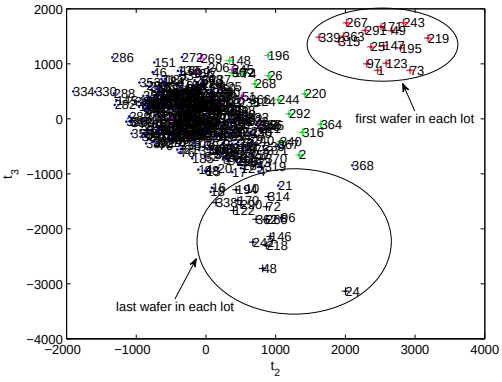


(c)

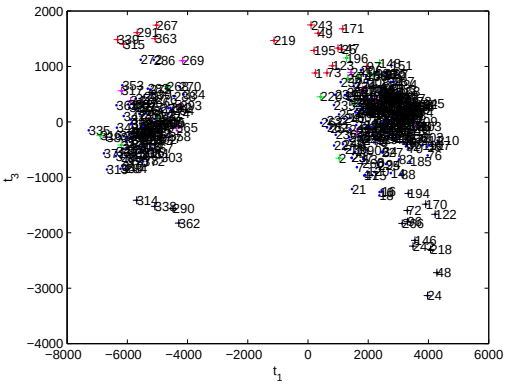
Figure 3.16: Applying PCA to the unfolded-1 OES data for all wafers: (a) The first score; (b) The second score; (c) The third score.



(a)



(b)



(c)

Figure 3.17: Applying PCA to the unfolded-1 OES data for all wafers: (a) The first vs the second score; (b) The first vs the third score; (c) The second vs the third score.

3.6.2 Time Series Data Summarised by Standard Deviation

Time series information contained in OES data can also be summarised by data statistics, such as mean, standard deviation, kurtois and skewness, *etc.* As an example, standard deviation is employed with the result that each wafer can be represented by a single score point using PCA. As illustrated in Fig. 3.18, the information contained in the time series for each wavelength is summarised by the standard deviation. The obtained data is referred to as the unfolded-2 data.

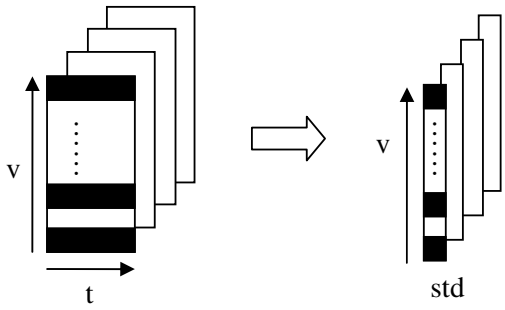
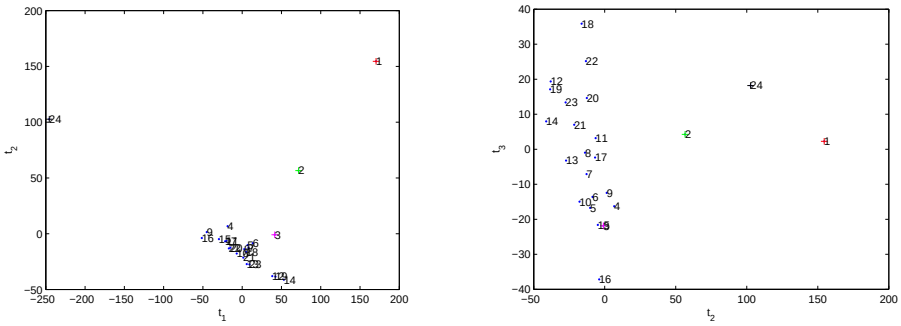


Figure 3.18: Method for unfolding a 2-D OES data matrix into a vector for a batch of wafers, std=standard deviation.

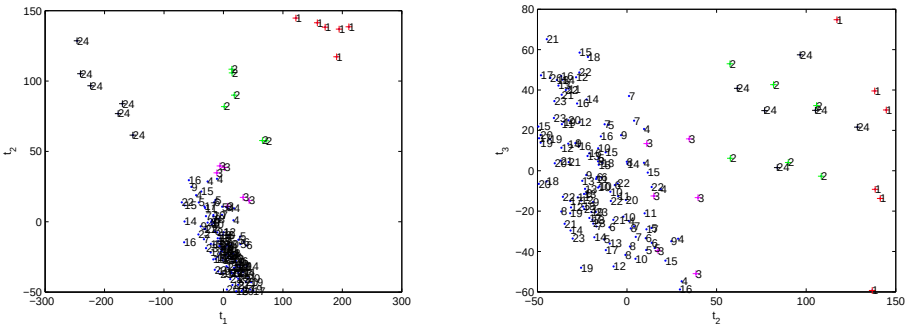
Applying PCA to unfolded-2 data for lot10-IDS3, 65.4%, 27.2% and 4.14% of the variance are captured in the first, second and third PC, respectively (96.7% accumulatively by the first three PCs). The first two and last wafers show up as outliers in the 2-D plots of the PC scores as shown in Fig. 3.19 and the pattern is confirmed by extending the examined data to 6 lots (from lot7-IDS3 to lot12-IDS3) as shown in Fig. 3.20.

Extending the PCA analysis to all 380 wafers in IDS3, 91.12%, 4.75% and 2.49% of the variance are captured by each of the first three PCs, respectively (98.36% accumulatively by the first three PCs). The 1-D and 2-D plots of the first three PC scores are shown in Fig. 3.21 and Fig. 3.22, respectively. As can be observed a sharp change occurs at lot 13 and the first two and last wafers show up as outliers. Hence there are no more new patterns found in the analysis of unfolded-2 data than the previous analysis, while as compared to the unfolded-1 data, the unfolded-2 data is computationally



(a) (b)

Figure 3.19: Applying PCA to the unfolded-2 OES data for wafers in lot10-IDS3: (a) The first vs the second score; (b) The second vs the third score.



(a) (b)

Figure 3.20: Applying PCA to the unfolded-2 OES data for wafers in 6 lots (lot7-IDS3 to lot12-IDS3): (a) The first vs the second score; (b) The second vs the third score.

more efficient. To investigate which variables are the main factors contributing to the difference in the scores, the PC loadings are employed. As shown in Fig. 3.23, the obvious difference between the first two loadings occurs at channels ranged from 400-600, 1300-1400 and 1550-2000, while to a less extent, the difference occurs at channels ranged from 200-400, 600-800 and 1100-1300. However, it is still difficult to identify the vital variables directly using PCA.

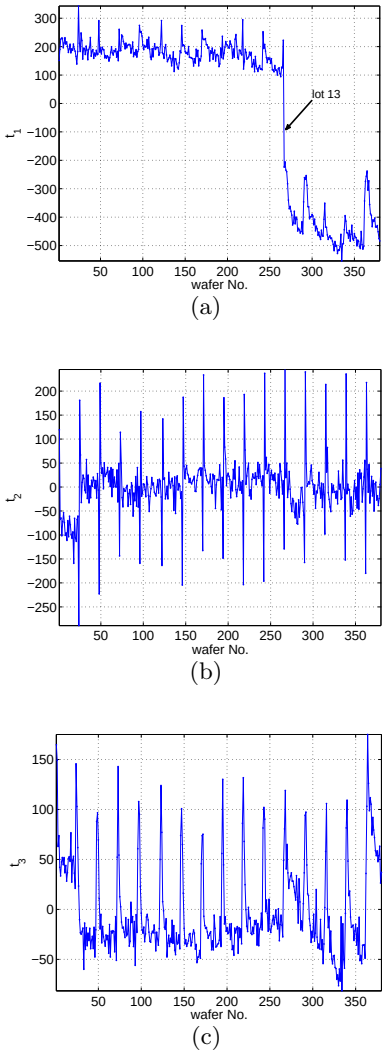


Figure 3.21: Applying PCA to the unfolded-2 OES data for all wafers in IDS3: (a) The first score; (b) The second score; (c) The third score.

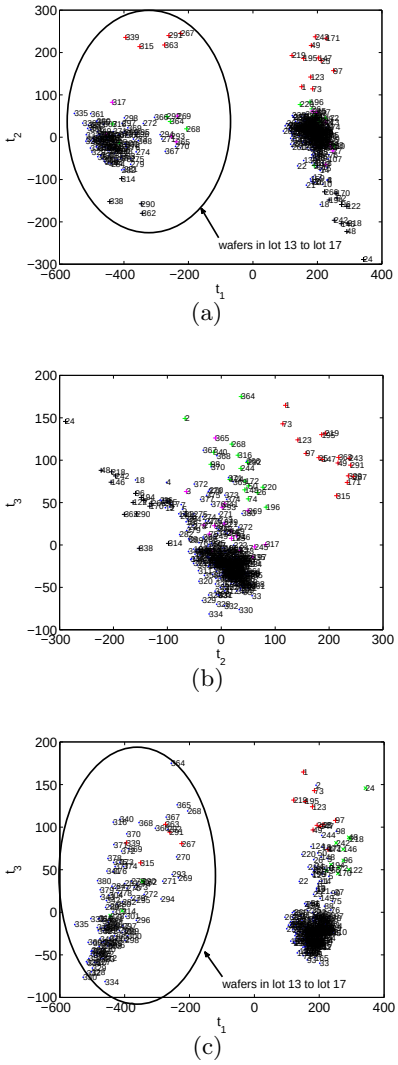
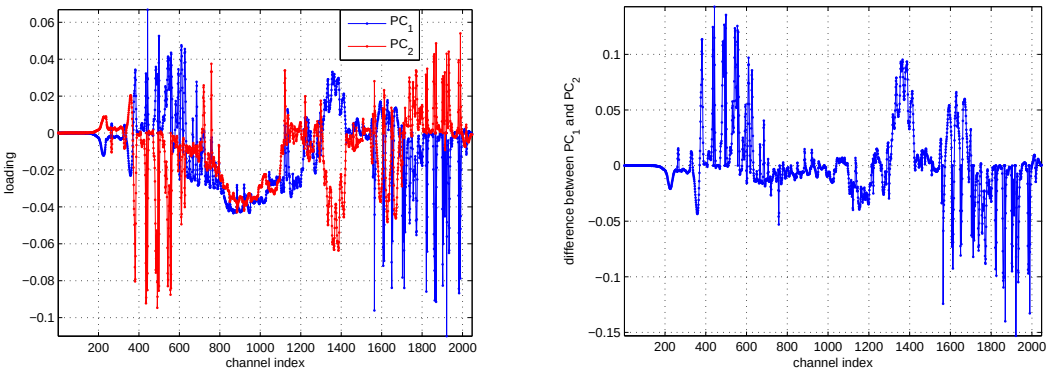


Figure 3.22: Applying PCA to the unfolded-2 OES data for all wafers in IDS3: (a) The first vs the second score; (b) The first vs the third score; (c) The second vs the third score.



(a) (b)

Figure 3.23: (a) The first two PC loadings for the PCA analysis of unfolded-2 data set; (b) Difference between these two loadings.

3.7 Noise Analysis

In the practice of measuring OES signals, noise is inevitably included in the measurements. Noise level or strength dictates the quality of measured signals. When significant noise is present, the signals can be completely destroyed by noise, and even to a less extent, noise makes the identification and interpretation of the signal patterns unreliable. Therefore, appropriate filtering is needed to clean up the signals. PCA analysis shows that the main OES variations are captured in the first few PCs, while in the residual data, no evident features are observable, which helps to confirm the existence of noise.

3.7.1 Noise Sources

Before we present the noise analysis, it is beneficial to have an awareness of different noise generation sources in OES. Normally, we consider the noise as being the high frequency variations in the signals. However, in our data, not all the variations are caused by noise. As such, we divide different variation sources into two categories: process variation and sensor noise, where the process variation refers to any variations occurring in the plasma emission system and sensor noise refers to the noise occurring in the plasma measuring system. The inspiration for this categorization can be found

in [48].

Process variations can be categorized as follows:

- recipe changes, which are set according to different etching process and different manufacturing product requirements (low frequency changes);
- controlled variable fluctuations, such as pressure, gas flow rate, power, *etc* (frequency range determined by closed loop bandwidth);
- plasma interaction with the wafers, which can cause variations in the wafer temperature (low frequency);
- exchange of chemicals due to the etching process itself (low frequency);
- plasma sputtering or deposition on the chamber walls; This refers to the physical interaction between plasma and the etching chamber walls, which affects the chemistry of the plasma, *i.e.* a combination of leaching of chemicals into the plasma from the chamber walls and absorption of plasma chemicals by the chamber walls;
- external disturbances; This refers to environmental changes around the etch chamber. In practice, these circumstances are insignificant as the chamber wall is sufficiently thick to insulate the plasma from external temperature changes;
- process transients; *i.e.* the transients between different process steps or different wafers, where the plasma is ignited, builds up and becomes stable and the chamber wall gradually heats up to its operating temperature;
- instability of etch by-products; Plasma generates volatile etch by-products at room temperature. The interactions between the molecules and atoms of different etch by-products are unpredictable, causing great uncertainty in the chemical optical emissions (high frequency).

Some of the process variations are identifiable, for example, the recipe changes, which can lead to obvious phase changes in the signals. Another example is the process

transients which can normally be observed in the first few samples of the signals after startup. However, other process variations such as those caused by the plasma interaction with the wafers and plasma sputtering on the chamber walls are not easily identified. This is largely due to the instability of the plasma, the performance of which is generally unpredictable.

For a better understanding of the sensor noise, we will firstly give a brief description of the OES detection mechanism, as shown in Fig 3.24. The light intensity emitted by the plasma, $I(\lambda)$, is considered as a function of wavelength and the output at a certain wavelength, $O(\lambda)$, is measured as the number of photons emitted in a given time, t. The relationship between the input and output can be formulated as follows [48]:

$$O(\lambda) = K \cdot f \cdot I(\lambda) \cdot t_r(\lambda) \cdot q(\lambda) \cdot t \tag{3.24}$$

where K is the gain of the CCD, f is the view factor of the plasma, t_r is the transmission function of the OES transmission system, and q is the quantum efficiency of the detector. Note that t_r and q are both frequency-sensitive.

Based on the form of the OES measurement system (shown in Fig. 3.24), the sources of sensor noise can be itemized as follows:

- the view factor, f, which can be affected by the location of observation and the reduction in the clarity of the OES viewing window due to sputtering,
- quantum efficiency, the percentage of photons being converted into a photoelectron, when the photon hits the CCD detector,
- the CCD integration time, t,
- shot noise, caused by the random fluctuations in photon arrival times. The strength of shot noise increases with signal strength. Thus, for large signals, shot noise generally dominates the noise.
- thermal noise (or dark noise), generated by thermal agitation of electrons in a conductor; By cooling the CCD detector, dark noise can be dramatically reduced;

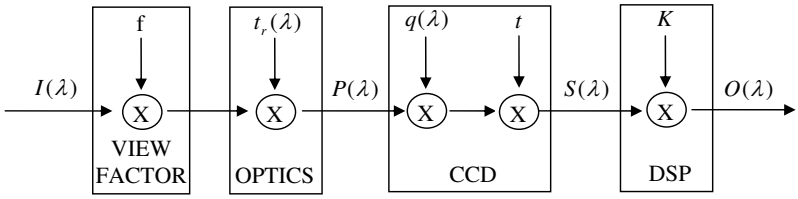


Figure 3.24: Diagram showing the OES measuring system

- and readout noise, including the conversion from an analogue signal to a digital number (the conversion is not perfectly repeatable even for the case of reading out the same pixel twice and each time with the same charge, the value could be slightly different) and the random fluctuations in the electronics.

From the list of the sensor noise sources, we can see that most noise can be regarded as white noise (broadband noise). Low pass filtering of the signal can be used to suppress the noise, but only at the expense of also suppressing the contributions of the high frequency process variations. However, this is not a concern as we are only interested in patterns generated by recipe changes, phase transitions and process transients that operate at a much lower frequency.

3.7.2 Selecting the Filter Bandwidth Based on Single Channels

To obtain a clearer representation of the signal patterns, we have to consider using filtering to remove the high frequency noise. Here, a low-pass Butterworth filter is used in preference to other widely used filters such as the Chebyshev Type I/Type II filter and the elliptic filter [10], as it is characterized by having a flat gain in its pass band and hence, provides minimal distortion of the filtered signals. Due to the slow roll-off into the stop band, a high-order Butterworth filter is needed to obtain faster roll-off. Here we employ a 4^{th}-order Butterworth filter. To determine the cut-off frequency of the low-pass filter, we begin by employing the DFT (Discrete Fourier Transform) [9] to view the signal frequency distribution.

Taking the strongest OES signal as an example, Fig. 3.25 (a) and (b) show the DFT and the cumulative PSD (Power Spectral Density) analysis of the signal, respectively. The cumulative PSD analysis shows that 76.3% of the signal power is contained in the bandwidth of 0.1Hz and 88.2% of the signal power contained in the bandwidth of 0.3Hz. As a second example, the DFT and the cumulative PSD (Power Spectral Density) analysis of the signal with the third highest power is shown in Fig. 3.26.

Let $\mathbf{x}_i(\mathbb{R}^{m\times 1})$ denote the raw signal from the i^{th} OES channel, for a given a low-pass Butterworth filter, the filtered signal ($\mathbf{x}_i^{\text{f}}$) is obtained by

$$\mathbf{x}_i^{\text{f}} = filt(\mathbf{x}_i) \tag{3.25}$$

where $filt(\cdot)$ is the butterworth filtering function and

$$\mathbf{x}_i^{\text{r}} = \mathbf{x}_i - \mathbf{x}_i^{\text{f}}, \tag{3.26}$$

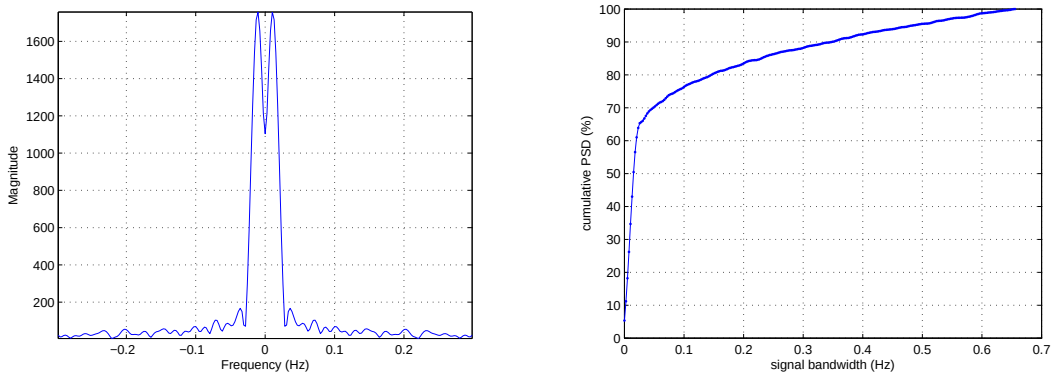
where $\mathbf{x}_i^{\text{r}}$ denotes the residual. To select the best low-pass filter (LPF) bandwidth, f^{B}, out of the set of sample frequencies, $\mathbf{f}$, tested, we define

$$f^{\text{B}} = \arg\min_{f} |corr(\mathbf{x}_i^{\text{f}}, \mathbf{x}_i^{\text{r}})|, \forall f \in \mathbf{f} \tag{3.27}$$

where $corr(\cdot)$ denotes the correlation function. Eq. (3.27) defines the optimum cut-off frequency as the one that gives the lowest correlation between the filtered signal ($\mathbf{x}_i^{\text{f}}$) and the residual signal ($\mathbf{x}_i^{\text{r}}$). A correlation analysis of the filtered signal and residual signal, as a function of LPF bandwidth is shown in Fig. 3.27 for the two selected signals.

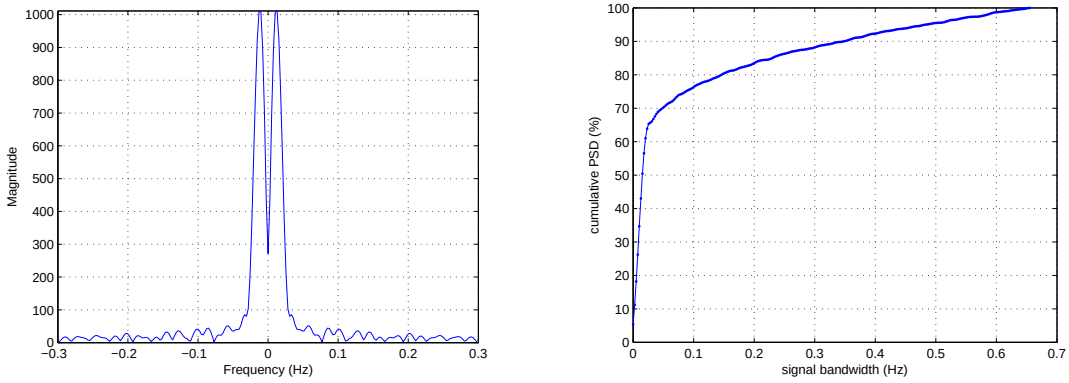
In Fig. 3.27 the shaded area shows the 95% confidence interval for the correlation coefficient estimates. Fig. 3.27 (a) shows that the minimal correlation is obtained when $f = 0.09$Hz and that the correlation is statistically insignificant for $f \geq 0.0755$Hz. Using the same method, $f^{\text{B}} \geq 0.0708$Hz for the second signal.

To evaluate the consistency of this approach, f^{B} was computed for each of the OES signals. Based on a DFT and cumulative PSD analysis, the search range for f^{B} is set between 0.001Hz and 0.2Hz. A plot of f^{B} for all the OES channels is shown in Fig. 3.28



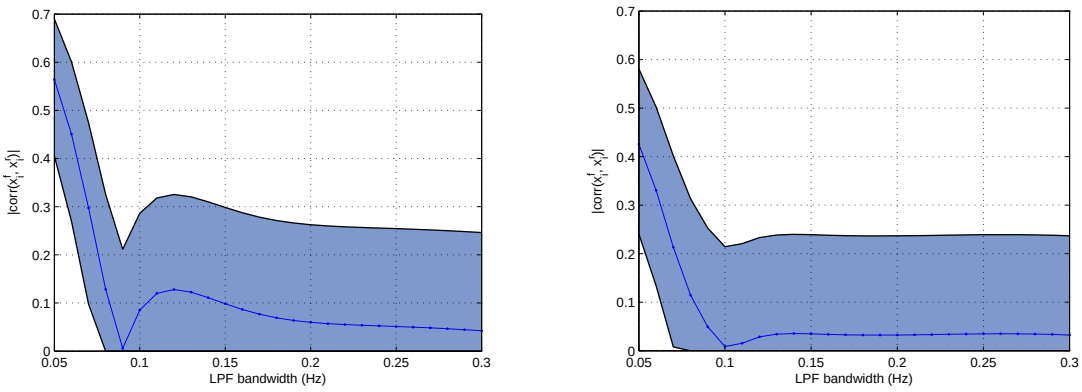
(a) (b)

Figure 3.25: Analysis of the strongest signal (IDS1): (a) DFT analysis; (b) Cumulative PSD analysis



(a) (b)

Figure 3.26: Analysis of the signal with the third highest power (IDS1): (a) DFT analysis; (b) Cumulative PSD analysis



(a) (b)

Figure 3.27: Correlation between the filtered signal and residual signal (IDS1), obtained by using different filter cut-off frequencies (shaded area showing the 95% confidence interval for the correlation coefficient estimates): (a) Strongest signal; (b) The signal with the third highest power

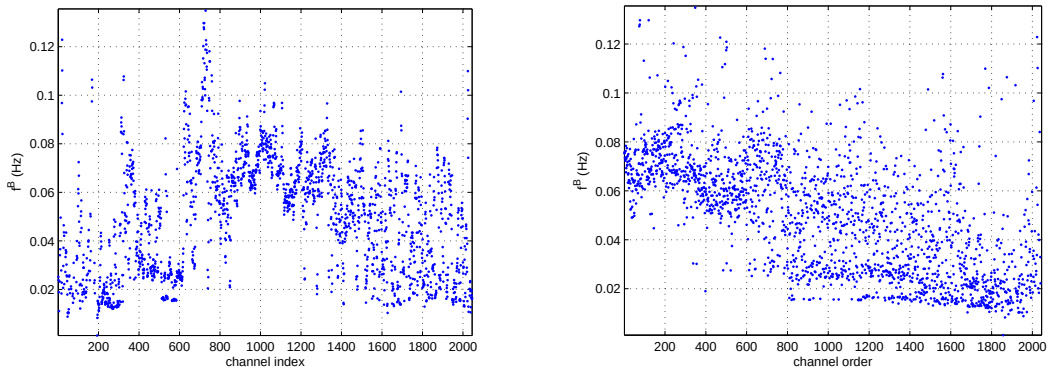


Figure 3.28: f^{B} for all the OES signals (IDS1): (a) Channel ordered sequentially; (b) Channel sorted in descending power order

(a), which shows that there is significant uncertainty in the value of f^{B} across channels and that there is some local correlation in values. Fig. 3.28 (b) reveals that the value of f^{B} varies as a function of signal power with the value of f^{B} at which correlation becomes insignificant, decreasing as the power in the signal decreases. In addition the spread in f^{B} increases substantially as signal power decreases (from 0.06-0.08Hz for high power signals to 0.01-0.12Hz for low power signals).

Since f^{B} estimates in Fig. 3.28 (b) are lower bounds on suitable LPF bandwidths and the values computed for the largest signal powers are the most reliable, $f^{\mathrm{B}} \geq 0.09$Hz represents a good compromise (96.2 % of channels) and choosing $f^{\mathrm{B}} = 0.1$Hz achieves minimal correlation for most channels (98.1 % of channels).

3.7.3 Principal Component Analysis of the Residual Signals

The single channel analysis in the previous subsection suggests that an appropriate low-pass filter (LPF) bandwidth is 0.1Hz. In this section, PCA is used to look at the patterns contained in the residuals across channels.

The residuals obtained from filtering all channels with a 0.1Hz bandwidth LPF are shown in Fig. 3.29 (a). A PCA analysis of the residuals reveals that the first PC explains 12.61% of the variance, with 4.48% and 3.98% variance explained by the second

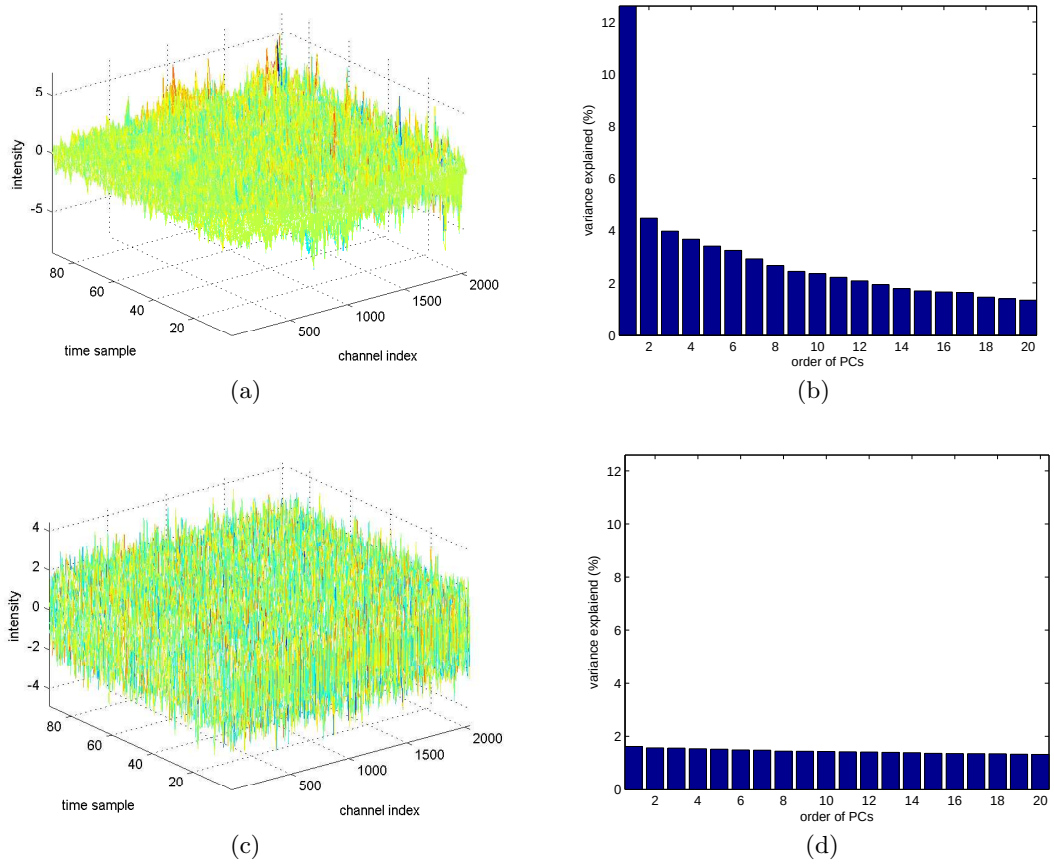


Figure 3.29: (a) Residual signals, obtained by subtracting the filtered signals from the original signals (LPF bandwidth = 0.1Hz); (b) Variance explained by each PC in a PCA analysis of the residual signals; (c) White noise signals, having the same size as the OES residual signals; (d) Variance explained by each PC

and third PCs, respectively. The variance explained by the first 20 PCs is shown in Fig. 3.29 (b). This confirms that inter-channel patterns exist in the residual data (*i.e.* some of the residual signals are correlated). If the residual signals were independent and identically distributed random noise, then the variance explained by each PC will be nearly the same, as shown in Fig. 3.29 (d). As such, 0.1Hz may not be the optimum choice for the filter bandwidth.

Table 3.2 and Fig. 3.30 show the variance explained by the first three PCs of a PCA analysis of the residuals, obtained with different filter bandwidths. As one can see, when the cut-off frequency exceeds 0.2Hz, variance explained by each of the first three

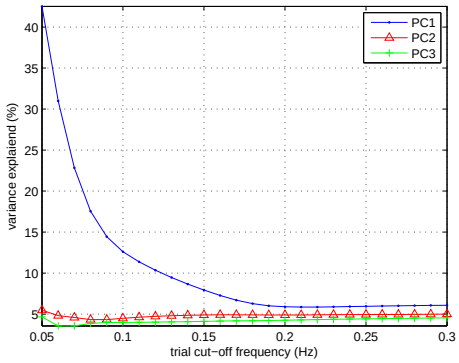


Figure 3.30: Variance explained by the first three PCs by PCA analysis of the residual data.

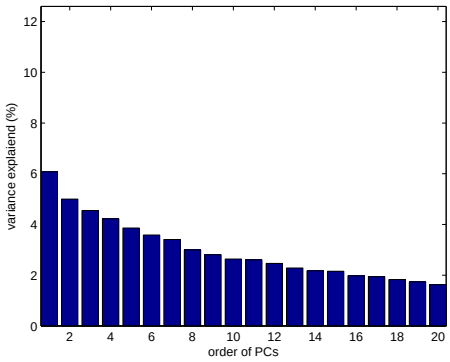


Figure 3.31: Variance explained by the first 20 PCs (the cut-off frequency equals to 0.2Hz).

PCs is nearly equal. However, the variance explained by the first PC (5.89%) is still a lot bigger than the variance explained by the 20^{th} PC (1.51%), as shown in Fig. 3.31. No matter what filter bandwidth is selected, inter-channel correlation still exists in the residuals. This suggests that there may be an intrinsic correlation in the signals. In fact, because of the limited spectral resolution of OES spectroscope, the optical emission at a given wavelength will be detected over a number of adjacent channels, leading to local correlation. In the next subsection, an analysis of the extent of this local correlation is given. Note that the existence of local correlation in the residuals corroborates the existence of high frequency process variations.

3.7.4 Local Correlation

Due to the limited OES spectroscope resolution, the optical emission signals are detected simultaneously by a number of adjacent OES channels. Taking the channels between 1300 and 1305 as an example, the over-time intensity changes of these channels, as shown in Fig. 3.32 are quite similar.

To assess the extent of this spectral spread, the local correlation between channels is investigated. The local correlation, r_j^L, is defined as the correlation between the signal ($\mathbf{x}_i$) from channel i and the signal from the channel, j channels away from channel i,

LPF Bandwidth (Hz)	Variance Explained by the First 3 PCs (%)
0.05	43.4515, 5.5774, 4.5752
0.06	31.4501, 4.8652, 3.5872
0.07	23.0399, 4.6428, 3.5641
0.08	17.6652, 4.3772, 3.8517
0.09	14.5262, 4.3616, 3.9650
0.10	12.6803, 4.5093, 3.9789
0.11	11.4298, 4.6479, 3.9970
0.12	10.3979, 4.7516, 4.0246
0.13	9.4913, 4.8215, 4.0525
0.14	8.6842, 4.8688, 4.0793
0.15	7.9504, 4.9032, 4.1117
0.16	7.2819, 4.9247, 4.1475
0.17	6.7072, 4.9284, 4.1747
0.18	6.2770, 4.9146, 4.1890
0.19	6.0164, 4.8969, 4.2134
0.20	5.8945, 4.8913, 4.2532
0.21	5.8570, 4.8995, 4.2946
0.22	5.8624, 4.9146, 4.3324
0.23	5.8870, 4.9298, 4.3659
0.24	5.9187, 4.9413, 4.3958
0.25	5.9517, 4.9481, 4.4233
0.26	5.9833, 4.9506, 4.4496
0.27	6.0128, 4.9549, 4.4748
0.28	6.0390, 4.9632, 4.4995
0.29	6.0603, 4.9786, 4.5243
0.30	6.0751, 5.0025, 4.5498

Table 3.2: Variance explained by the first three PCs of the residuals for different LPF bandwidths.

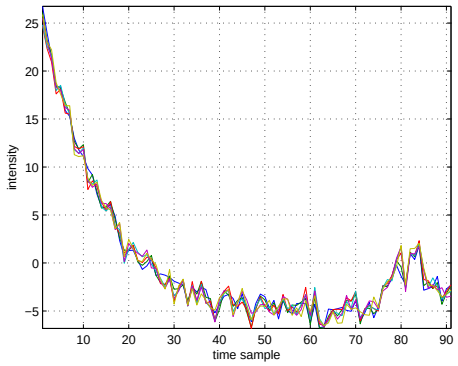


Figure 3.32: Intensity changes of OES channels from 1300 to 1305 (IDS1).

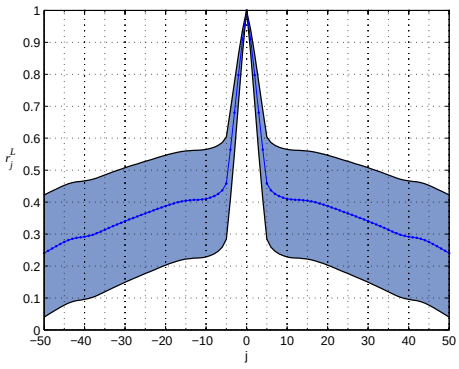


Figure 3.33: Local correlation, r_j^L, for the 0.1Hz LPF residual (IDS1).

(*i.e.* $\mathbf{x}_{i+j}$) averaged over all channel positions, that is:

$$r_j^L = \frac{1}{n-j}\sum_{i=1}^{n-j} corr(\mathbf{x}_i^r, \mathbf{x}_{i+j}^r), \text{ for } j \geq 0 \tag{3.28}$$

$$= \frac{1}{n+j}\sum_{i=1-j}^{n} corr(\mathbf{x}_i^r, \mathbf{x}_{i+j}^r), \text{ for } j \leq 0. \tag{3.29}$$

A plot of r_j^L for the 0.1Hz LPF residual is given in Fig. 3.33. It can be seen that the spectral spread is ± 5 on either side of a given channel. The shaded area shown in Fig. 3.33 denotes the 95% confidence interval associated with each r_j^L.

3.7.5 Crosscorrelation of the Residual Signals

To estimate the inter-channel correlation between the residual signals, the following average crosscorrelation is used:

$$\bar{r}^C = \frac{1}{n-l}\sum_{i=1}^{n-l}\left(\frac{\sum_{j=i+l}^{n} corr(\mathbf{x}_i^r, \mathbf{x}_j^r)}{n-i-l+1}\right), \tag{3.30}$$

where $\mathbf{x}_i^r$ and $\mathbf{x}_j^r$ are the i^{th} and j^{th} residual signals, respectively and l denotes the number of adjacent channels that are correlated due to spectral leakage. From the analysis in Section 3.7.4, $l = 6$. By omitting the locally correlated channels, the value of $\bar{r}^C$ better reflects the filter bandwidth dependent correlation in the residual signals. Fig. 3.34 shows a plot of $\bar{r}^C$ as a function of LPF bandwidth and the shaded area

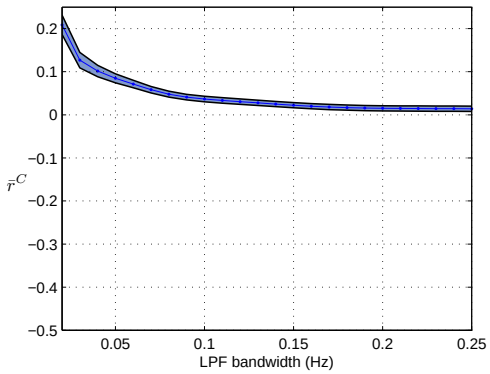


Figure 3.34: Changes of $\bar{r}^C$ as a function of LPF bandwidth

denotes the 95% confidence interval. The value of $\bar{r}^C$ decreases with increasing LPF bandwidth and drops below 0.1 for $f^{\mathrm{B}} \geq 0.05$Hz and 0.05 for $f^{\mathrm{B}} \geq 0.1$Hz.

3.7.6 Crosscorrelation between the Residual Signals and Filtered Signals

Crosscorrelation analysis is used to estimate the correlation between the filtered signals and residuals. Denoting $\bar{r}_i$ as the correlation between $\mathbf{x}_i^{\mathrm{f}}$ and $\mathbf{x}_i^{\mathrm{r}}$ (as defined in Eq. (3.25) and Eq. (3.26), respectively), the averaged correlation coefficient ($\bar{r}$) for all the channels is defined as

$$\bar{r} = \frac{1}{n}\sum_{i=1}^{n} \bar{r}_i, \tag{3.31}$$

where $\bar{r}_i = corr(\mathbf{x}_i^{\mathrm{f}}, \mathbf{x}_i^{\mathrm{r}})$ and n is the number of channels. Fig. 3.35 shows the change of $\bar{r}$ as a function of LPF bandwidth. It can be seen that the averaged correlation between the filtered signals and residuals is insignificant for $f^{\mathrm{B}} \geq 0.1$Hz.

3.7.7 Autocorrelation of the Residual Signals

Autocorrelation analysis is employed to estimate the correlation between the residual signal and its time-lagged values. If the residual signal is noise, then no significant correlation should exist. To measure the autocorrelation for all the residual signals, we define a new function, $\bar{r}^A$:

$$\bar{r}^A = \frac{1}{2(m-1)n} \sum_{j=-m+1, j\neq 0}^{m-1} |\sum_{i=1}^{n} corr(\mathbf{x}_i^r, \mathbf{x}_i^r(j))|, \tag{3.32}$$

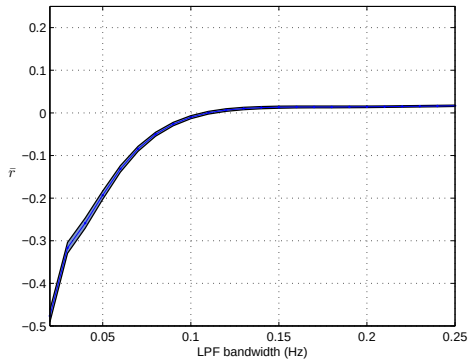


Figure 3.35: Changes of $\bar{r}$ for different LPF bandwidths

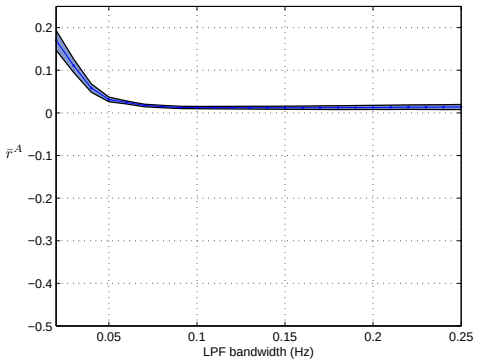


Figure 3.36: Changes of $\bar{r}^A$ as a function of LPF bandwidth

Method	LPF Bandwidth, f^B, (Hz)
Single signal based	$f^B \geq 0.09$
$\bar{r}^C$	$f^B \geq 0.1$
$\bar{r}$	$f^B \geq 0.1$
$\bar{r}^A$	$f^B \geq 0.05$

Table 3.3: LPF bandwidth selected by different methods

where n is the number of OES channels and $\mathbf{x}_i^r(j)$ is the j^{th} lagged signal of $\mathbf{x}_i^r$, ($\mathbf{x}_i^r \in \mathbb{R}^{m\times 1}$). Hence, $\bar{r}^A$ measures the averaged correlation levels between the residual signal and its lagged signals over all channels. The smaller the value of $\bar{r}^A$, the lower the correlation in the residuals. Fig. 3.36 shows the variation in $\bar{r}^A$ as a function of LPF bandwidth and the shaded area denotes the 95% confidence interval. The value of $\bar{r}^A$ decreases with increasing LPF bandwidth and drops below 0.05 for $f^{\text{B}} \geq 0.05$Hz and close to 0 for $f^{\text{B}} \geq 0.1$Hz.

3.7.8 Selection of the LPF Bandwidth

While the analysis in the previous sections cannot provide an exact optimal solution to the LPF bandwidth, it is clear that there is no method that can be universally applicable. As shown in Table 3.3, the lower bounds on f^{B} identified using the different techniques are relatively consistent. Thus, 0.1Hz is selected as the LPF bandwidth.

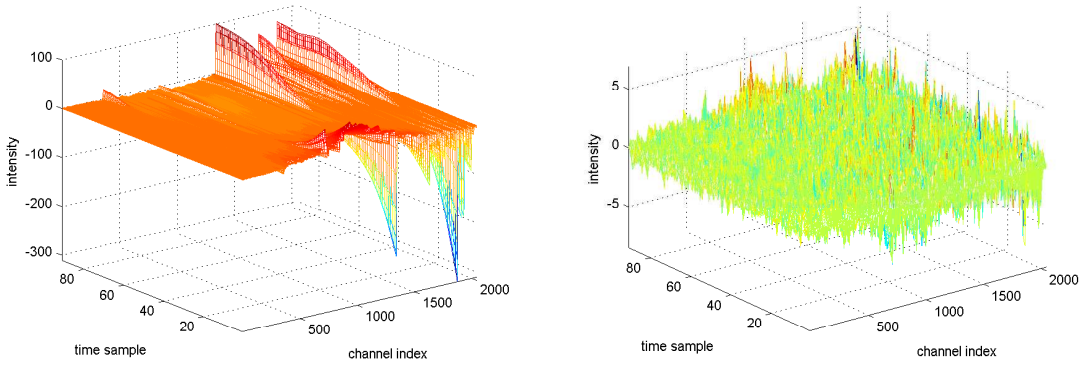


Figure 3.37: The raw OES data filtered by a 4^{th} order low-pass Butterworth filter with cut-off frequency set at 0.1Hz: (a) Filtered signals; (b) Residuals

3.7.9 Filtering Result Visualization

The results of filtering the raw IDS1 OES signals using a 0.1Hz bandwidth 4^{th} order low-pass Butterworth filter are shown in Fig. 3.37. Plot (a) shows the filtered signals and plot (b) the residuals, respectively. To observe the effect of filtering on individual channels, the results for three channels are shown in Fig. 3.38 (b), (c) and (d), respectively. These channels correspond to the 33.33%, 66.67% and 98.85% division points of the cumulative signal power plot (sorted in descending order) as shown in Fig. 3.38 (a). This confirms that the selection of 0.1Hz as the LPF bandwidth is reasonable for filtering the OES signals.

3.7.10 Signal to Noise Ratio

Signal-to-noise ratio (SNR) is the standard method for measuring the strength of a signal relative to the noise. Here, SNR of a given signal ($\mathbf{x}_i$) is estimated as

$$SNR_i = \frac{pow(\mathbf{x}_i^{\text{f}})}{pow(\mathbf{x}_i^{\text{r}})}, \tag{3.33}$$

where $pow(\cdot)$ denotes the signal power. For a signal $\mathbf{x} \in \mathbb{R}^{n\times 1}$, this is computed as

$$pow(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i)^2. \tag{3.34}$$

Patterns in low SNR signals will be seriously corrupted by noise and cannot be detected reliably. Therefore, as a final pre-processing step, these channels should be removed.

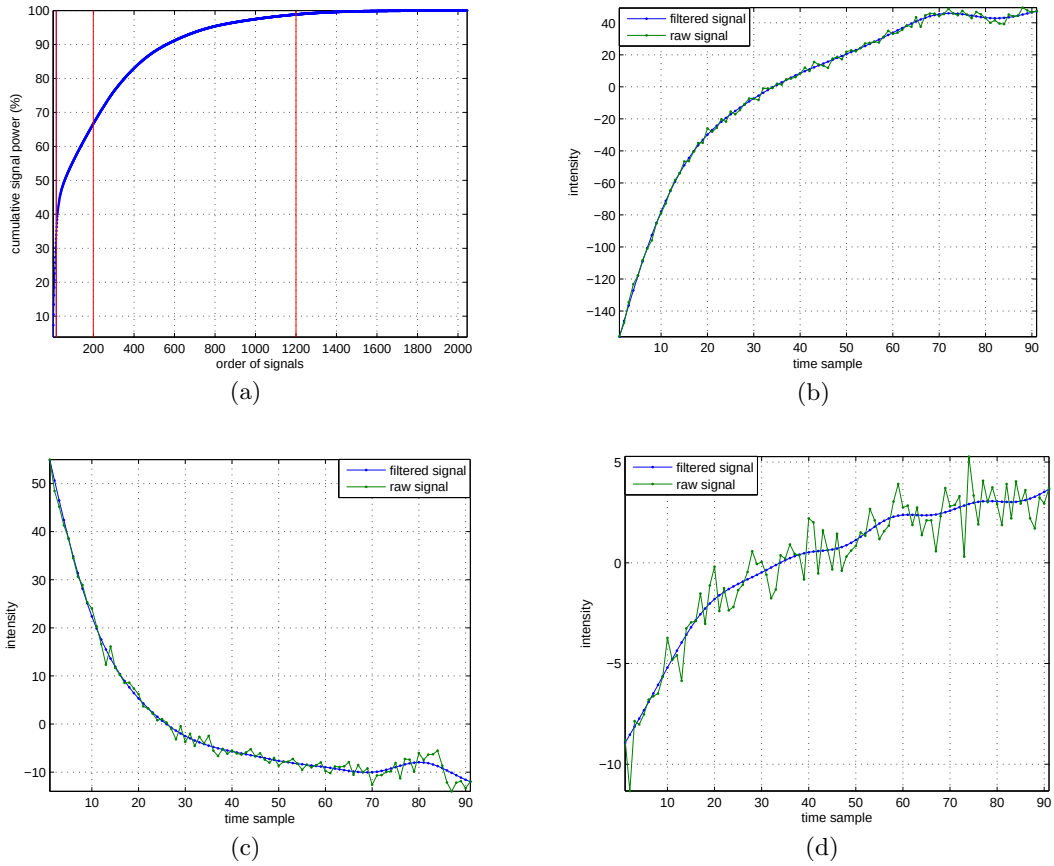


Figure 3.38: Filtering results for three selected signals: (a) Cumulative signal power plot for signals sorted in descending power order; (b), (c) and (d) show the filtering results for the signals corresponding to the 33.33%, 66.67% and 98.85% division points of the cumulative signal power plot

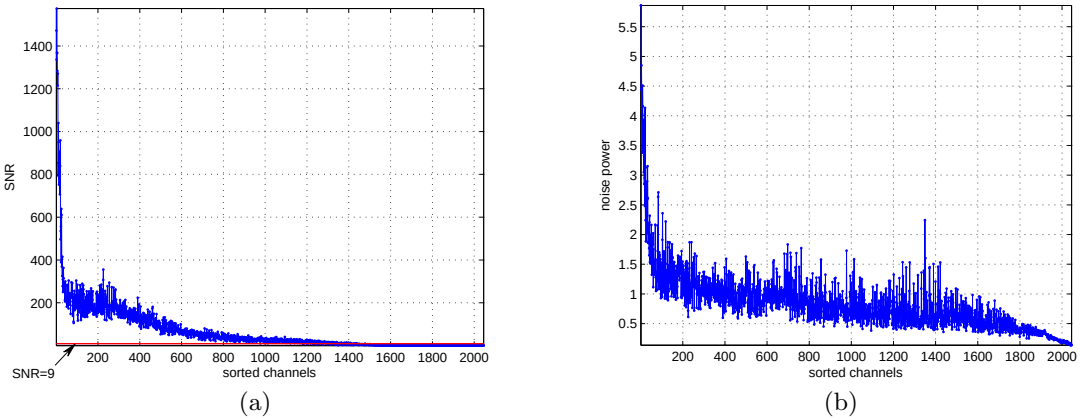


Figure 3.39: (a) SNR across channels; (b) Noise power across channels (The channels are sorted in descending power order.)

Here, 9 is selected as the cut-off point for SNR, leading to the removal of 691 channels for IDS1. A plot of SNR (sorted in descending power order) and a plot of the noise power across channels are shown in Fig. 3.39 (a) and (b), respectively. It can be seen that the signals with SNR < 9 correspond to the low power OES signals.

Using the same method filtering IDS2 leads to the removal of 1473 channels. As noted in Section 2.3.1, the resulting filtered and preprocessed data sets are referred to as IDS1Filt and IDS2Filt, respectively. The selection of 9 as the SNR is determined by the requirements of the max separation clustering algorithm and will be discussed in Section 6.6.6.

3.8 Discussion and Conclusions

PCA, one of the most widely used multivariate data analysis algorithms has been introduced in this chapter and a detailed description of the theory and numerical solutions is given. Graphical displaying of the score patterns in 2-D format has shown to be effective in disclosing the process variations across wafers, while computationally expensive.

As low cost alternatives, two methods have been proposed, conventional PCA analysis of unfolded data and a simple method for monitoring changes in the directions of the PC loading vectors. The basic concepts involved in these two methods are not new. What is shown in this chapter is that the methods are of great practical use for summarizing the information contained in high-volume OES data and are effective for easy visualization of the process variations across wafers and lots. However, the issue with PCA is that since the PCs are linear combination of all underlying variables, it cannot be used to identify key process variables and hence cannot be used to spot the root reason that causes the variations. As such, sparse principal component analysis, a modification of PCA which attempts to address this deficiency, is investigated in the next chapter.

Another important contribution made in this chapter is the noise analysis. A detailed

description of noise sources in etch processes is presented. The method proposed for selecting the noise filter bandwidth is new, achieved systematically by the auto-correlation and cross-correlation analyses of the filtered signals and residual data, respectively. In addition, a new method for estimating the local correlation has been proposed to provide an effective way for estimating the OES spectroscope resolution.

Chapter 4

Sparse Principal Component Analysis

4.1 Introduction

The success of principal component analysis (PCA) lies in the fact that as a dimension reduction tool, it can reconstruct high dimensional data via a limited number (2 or 3 in general) of principal components and retain most of the variation in the data. However, the loadings obtained by PCA are linear combinations of all variables and the variable coefficients are typically nonzero. This makes it difficult to use PCA directly for variable selection.

Research on obtaining sparse solutions (solutions with zero coefficients) has been conducted for over five decades. The earliest method, proposed in 1958, is referred to as varimax [80]. Using varimax rotation, a number of the coefficients of the loading vectors can be adjusted to have greater values than the remaining coefficients. Such adjustment can help in the selection of key variables, but it is hard to quantify the distinction between small and large coefficients.

Jeffers [74] proposed a straight-forward method for achieving PCA sparsity. For each loading, any coefficients that are less than 70% of the greatest one are set to zero, regardless of their sign. This method can lead to a selection deficiency in two cases,

one where the variables have small coefficients and the other where the variables have high mutual correlations [11].

In [163], the 'simple principal components' is proposed. This focuses on restricting the coefficients of the loadings to have integer values, such as -1,0 and 1, to help simplify variable selection.

The first true algorithmic method for achieving sparse loadings was proposed in 2003 by Jolliffe *et al.* [79] and is known as SCoTLASS (Simplified Component Technique for Least Absolute Shrinkage and Selection). This employs a penalty term referred to as the Least Absolute Shrinkage and Selection Operator (LASSO) [160] to force loadings to be sparse. Nevertheless, it is not practical due to the relatively high computational cost [195].

A recently proposed algorithm, known as semidefinite programming, is described in [27]. Using this method, the normal loadings are constrained by a cardinality condition, that is, a limit on the number of the nonzero elements in each loading. By relaxing this constraint, the problem is converted into a convex optimization problem and hence, can use semidefinite programming as a solution. The generated PCs are shown to be able to explain larger variance than competing algorithms, but the computational cost is high.

In 2004 Zou *et al.* [195] proposed an alternative approach to solve the sparse principal component analysis (SPCA) problem, which they refer to as elastic net for SPCA (EN-SPCA). EN-SPCA can be implemented in two forms. One is similar to an approach used to solve the LASSO problem and the other is the so called soft thresholding algorithm, designed for handling large data sets (thousands of variables). Both EN-SPCA implementations are computational alternatives to semidefinite programming, but the latter implementation has the key advantage that it can scale to much larger problems than the semidefinite programming algorithm.

Another alternative algorithm for solving SPCA is proposed in [141], known as sparse

PCA via regularised SVD (sPCA-rSVD). sPCA-rSVD is implemented based on the close connection between PCA and singular value decomposition (SVD) and promotes sparsity in PC loadings via the introduce of regularization penalties. The key advantage of the sPCA-rSVD algorithm is that the matrix ill-conditioning problem is effectively avoided using element-based calculation. Details of sPCA-rSVD are presented in the next chapter, with respect to its close relationship with our new proposed adaptive weighting SPCA algorithm.

This chapter focuses on EN-SPCA. In the first part, the theoretical framework involved in EN-SPCA is introduced. This includes the introduction of least squares, ridge, least absolute shrinkage and selection operator and elastic net regression problems and the formulation of SPCA in an elastic net regression framework. In the second part, the numerical solution is provided, followed by the discussion of the variance explained by the sparse components. With the aid of artificial data, the properties of EN-SPCA are illustrated. Finally, the application of EN-SPCA to OES data is investigated using IDS1 and IDS1Filt.

4.2 Theoretical Framework

This section gives a theoretical description of the EN-SPCA algorithm and an overview of relevant background theory (supporting proofs are included in Appendix A.1). For a complete treatment of the theory, please consult [195, 194, 196, 193].

4.2.1 Least Squares Regression

The regression methods covered in this chapter all originate from least squares (LS) approximations. Given an $m \times n$ data matrix $\mathbf{X}$, m being the number of observations and n being the number of variables, $\mathbf{X}$ can be expressed as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, where $\mathbf{x}_i = [x_{1i}, \ldots, x_{mi}]^{\mathrm{T}}$.

In regression analysis, $\mathbf{X}$ is used as the input data set. The output data set, $\mathbf{y}(\in \mathbb{R}^{m\times 1})$ can be expressed as

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \varepsilon \tag{4.1}$$

where $\mathbf{b}(\in \mathbb{R}^{n\times 1})$ are called the regression coefficients or parameters and ε $(\in \mathbb{R}^{m\times 1})$ is the random disturbance or error [18]. The expected value and variance of ε can be expressed as

$$\mathrm{E}(\varepsilon) = \mathbf{0} \quad \text{and} \quad \mathrm{E}(\varepsilon\varepsilon^{\mathrm{T}}) = \sigma^2\mathbf{I}_m. \tag{4.2}$$

The LS estimate of $\mathbf{b}$ can be defined as

$$\widehat{\mathbf{b}} = \arg\min_{\mathbf{b}} ||\mathbf{y} - \mathbf{X}\mathbf{b}||_2^2, \tag{4.3}$$

where $||\cdot||_2$ denotes the L_2-norm. The solution for $\mathbf{b}$ is given by

$$\widehat{\mathbf{b}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}, \tag{4.4}$$

where $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}$ is the pseudoinverse of $\mathbf{X}$. The least square estimate minimises the mean square error $(\mathrm{E}(\varepsilon^{\mathrm{T}}\varepsilon))$ and is the best linear unbiased estimate of $\mathbf{b}$ [60].

If $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is singular (its inverse does not exist), the LS estimator cannot be used to calculate the regression coefficients. Furthermore, when $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is close to singular, the coefficients estimates become very unstable, varying greatly for small changes in $\mathbf{X}$. Practically, these situations occur when the problem is under determined $(m < n)$ or there is significant amount of collinearity in the data. A typical way of addressing this issue is the so-called ridge regression.

4.2.2 Ridge Regression

Ridge regression solves the singularity problem by regularising the parameter estimates. Defining $\mathbf{b}^{\mathrm{R}}$ as the ridge estimate of $\mathbf{b}$, the ridge regression problem can be expressed as [60]:

$$\widehat{\mathbf{b}}^{\mathrm{R}} = \arg\min_{\mathbf{b}} ||\mathbf{y} - \mathbf{X}\mathbf{b}||_2^2 + \gamma_2||\mathbf{b}||_2^2, \tag{4.5}$$

where γ_2 is the tuning parameter. The solution to Eq. (4.5) can be expressed as

$$\widehat{\mathbf{b}}^{\mathrm{R}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \gamma_2\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{4.6}$$

The ridge estimate optimisation problem Eq. (4.5) can also be formulated as:

$$\widehat{\mathbf{b}}^{\mathrm{R}} = \arg\min_{\mathbf{b}} ||\mathbf{y} - \mathbf{X}\mathbf{b}||^2, \text{ s.t. } ||\mathbf{b}||_2^2 \leq c_2, \tag{4.7}$$

where c_2 is the upper bound of the L_2-norm of the regression coefficients. For every γ_2, there exists a c_2 that gives the same constraint on the regression coefficients. Eq. (4.5) is referred to as the penalised formulation while Eq. (4.7) is referred to as the constrained formulation.

According to Eq. (4.4) and Eq. (4.6), the relationship between ridge and LS can be expressed as

$$\widehat{\mathbf{b}}^{\mathrm{R}} = [\mathbf{I}_n + \gamma_2(\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}]^{-1}\widehat{\mathbf{b}} = \mathbf{K}\widehat{\mathbf{b}}. \tag{4.8}$$

In the special case of an orthogonal $\mathbf{X}$ matrix, Eq. (4.8) reduces to

$$\widehat{\mathbf{b}}^{\mathrm{R}} = \frac{1}{1+\gamma_2}\widehat{\mathbf{b}} = k\widehat{\mathbf{b}}, \tag{4.9}$$

where k is a scalar. Equation (4.6) shows that in ridge regression, the singular problem is actually solved by adding a fixed positive value to all the elements in the main diagonal of $\mathbf{X}^{\mathsf{T}}\mathbf{X}$. The effect of this adjustment is that parameter estimates are shrunk towards zero, as highlighted by Eq. (4.9), leading to biased estimates. Despite this bias, when $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is ill-conditioned, the ridge estimator can provide better 'prediction accuracy' than the LS estimator. This can be demonstrated by expressing prediction accuracy as the mean squared error (MSE) in parameter estimates, that is, $J_{MSE}(\gamma_2) = E[(\widehat{\mathbf{b}}^{\mathrm{R}} - \mathbf{b})^{\mathsf{T}}(\widehat{\mathbf{b}}^{\mathrm{R}} - \mathbf{b})]$. Substituting for $\widehat{\mathbf{b}}^{\mathrm{R}}$ and expanding gives

$$\begin{aligned} J_{MSE}(\gamma_2) =& E[(\widehat{\mathbf{b}} - \mathbf{b})^{\mathsf{T}}\mathbf{K}^{\mathsf{T}}\mathbf{K}(\widehat{\mathbf{b}} - \mathbf{b})] + (\mathbf{K}\mathbf{b} - \mathbf{b})^{\mathsf{T}}(\mathbf{K}\mathbf{b} - \mathbf{b})) \\ =& J_1(\gamma_2) + J_2(\gamma_2), \end{aligned} \tag{4.10}$$

where $\mathbf{K}$ is a function of γ_2 as defined in Eq. (4.8). Thus, two terms contribute to the MSE. The first term, $J_1(\gamma_2)$, is the variance of the parameter estimates and the second term, $J_2(\gamma_2)$, is the squared distance from $\mathbf{K}\mathbf{b}$ to $\mathbf{b}$ (squared bias). Hence we have a bias-variance trade-off controlled by γ_2.

Fig. 4.1 presents a comparison of the MSE as a function of ridge and LS estimates. The dashed line shows the overall MSE for the ridge estimate, and the horizontal line is the LS estimate (which is constant since it is not a function of γ_2.) Also shown on the graph are the variance and bias squared terms which contribute to the ridge estimate

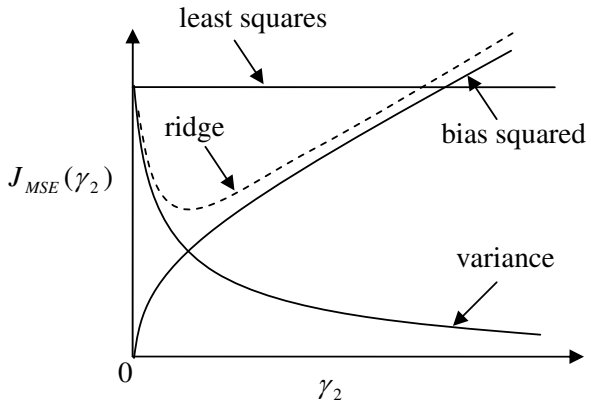


Figure 4.1: Mean squared error in ridge and LS estimates [60].

MSE. As can be seen if γ_2 is appropriately chosen, then

$$J_{MSE}(\gamma_2) < J_{MSE}(0). \tag{4.11}$$

Hence, the ridge estimate is superior to the LS estimate.

4.2.3 Least Absolute Shrinkage and Selection Operator

Least absolute shrinkage and selection operator (LASSO) is a method that tries to obtain the minimum of the residual sum of squares subject to a constraint on the sum of the absolute values of the regression coefficients [160]. Defining $\mathbf{b}^{\mathrm{L}}$ as the LASSO estimate of $\mathbf{b}$, the LASSO problem can be expressed as

$$\widehat{\mathbf{b}}^{\mathrm{L}} = \arg\min_{\mathbf{b}} ||\mathbf{y} - \mathbf{X}\mathbf{b}||_2^2 + \gamma_1 ||\mathbf{b}||_1, \tag{4.12}$$

where $\gamma_1 \geq 0$ is the tuning parameter and $||\cdot||_1$ denotes the L_1-norm, that is

$$||\mathbf{b}||_1 = \sum_{i=1}^{n} |b_i|. \tag{4.13}$$

Eq. (4.12) can also be expressed as a constrained optimisation problem, that is:

$$\widehat{\mathbf{b}}^{\mathrm{L}} = \arg\min_{\mathbf{b}} ||\mathbf{y} - \mathbf{X}\mathbf{b}||_2^2, \text{ s.t. } ||\mathbf{b}||_1 \leq c_1, \tag{4.14}$$

where c_1 is the upper bound on the L_1-norm of the regression coefficients. For every γ_1, there exists a c_1 that gives the same constraint on the regression coefficients.

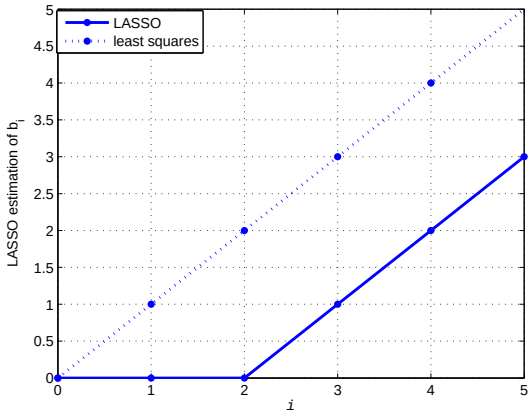


Figure 4.2: An example showing that zero coefficients can be produced by LASSO [160].

Unlike LS or ridge regression the LASSO estimate does not have a linear solution and cannot in general be expressed in closed form. However, when $\mathbf{X}$ is an orthogonal matrix, the LASSO solution can be expressed as

$$\begin{aligned} \widehat{b}_i^{\mathrm{L}} &= sign(\widehat{b}_i) max(0, \psi) \\ \psi &= |\widehat{b}_i| - \frac{\gamma_1}{2}, \end{aligned} \tag{4.15}$$

for $i = 1, \cdots, n$ [196]. $sign(\cdot)$ denotes the sign of the measured variable and $max(0, \psi)$ acts as threshold operator on ψ. Provided $\psi > 0$, $max(0, \psi) = \psi$; otherwise, $max(0, \psi) = 0$. Eq. (4.15) shows that when γ_1 is large, $(|\widehat{b}_i| - \frac{\gamma_1}{2})$ is more likely to be negative and therefore, more coefficients are likely to be shrunk to zero. Thus the larger the tuning parameter in LASSO, the more sparsity will be brought to the solutions. The effect of the LASSO constraint is illustrated graphically in Fig. 4.2. This property makes LASSO a valuable tool for variable selection [194].

Thus, while the ridge penalty shrinks parameters toward zero, the LASSO penalty has a tendency to force certain coefficient to be exactly zero. Fig. 4.3 provides a graphical illustration of how LASSO can yield sparse solutions, while ridge generally does not. The elliptical contours correspond to different values of the LS quadratic cost function and the central point represents the LS estimate of the regression coefficients. The shaded area shown in Fig. 4.3(a) represents the feasible region that satisfies the

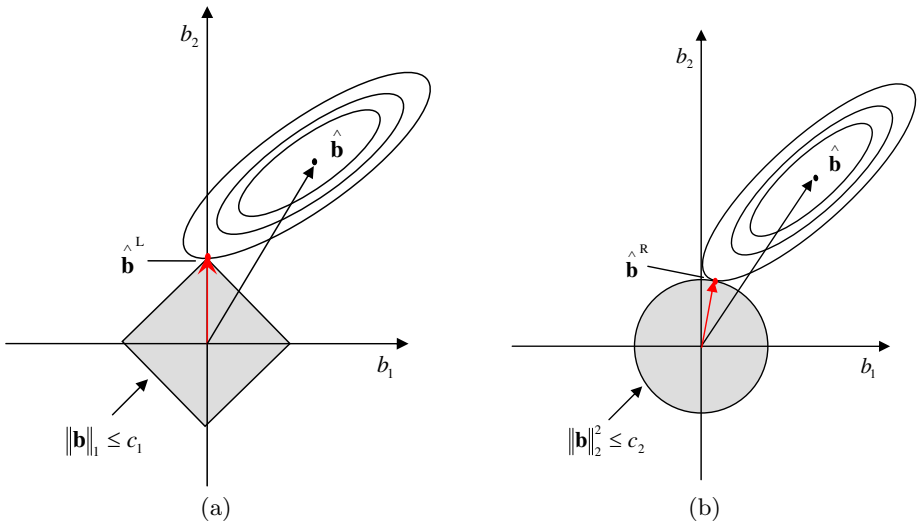


Figure 4.3: Comparison between LASSO and ridge estimators. The shaded regions correspond to the feasible region defined by (a) the LASSO constraint ($||\mathbf{b}||_1 < c_1$); (b) the ridge constraint ($||\mathbf{b}||_2^2 < c_2$). The elliptical contours correspond to different values of the LS quadratic cost function and the central point represents the LS estimate of the regression coefficients.

LASSO constraint, $||\mathbf{b}||_1 \leq c_1$. The LASSO solution is achieved when the contours first touch the boundary of the feasible region and will often correspond to the corner points, *i.e.* the coordinate points. This means that some coefficients of the parameter vector will be zero. For ridge regression, the feasible region is defined by $||\mathbf{b}||_2^2 \leq c_2$, which corresponds to the circle area shown in Fig. 4.3 (b). In this case, the contours and feasible region generally do not intresect on the coordinate axis. Hence zero coefficients are much less likely to be obtained.

4.2.4 Elastic Net

The elastic net regression method uses a combination of the constraints from ridge regression and LASSO to simultaneously address the matrix singularity problem and bring sparsity to the resulting solutions. Defining $\mathbf{b}^{\mathrm{N}}$ as the naive elastic net estimate of $\mathbf{b}$, the naive elastic net problem can be expressed as

$$\mathbf{b}^{\mathrm{N}} = \arg\min_{\mathbf{b}} ||\mathbf{y} - \mathbf{X}\mathbf{b}||_2^2 + \gamma_2||\mathbf{b}||_2^2 + \gamma_1||\mathbf{b}||_1. \tag{4.16}$$

The reason why the solution of Eq. (4.16) is referred to as naive is because bias is introduced twice, while the estimation variance is not reduced as compared to either the LASSO or ridge estimate. As proposed in [194], the excessive shrinkage can be compensated by re-scaling the naive elastic net estimator, *i.e.* elastic net. The elastic net estimate, $\mathbf{b}^{\mathrm{EN}}$, can be expressed as

$$\begin{aligned}\mathbf{b}^{\mathrm{EN}} =&(1+\gamma_2)\mathbf{b}^{\mathrm{N}} \\ =&(1+\gamma_2)\arg\min_{\mathbf{b}}||\mathbf{y}-\mathbf{X}\mathbf{b}||_2^2+\gamma_2||\mathbf{b}||_2^2+\gamma_1||\mathbf{b}||_1,\end{aligned} \tag{4.17}$$

which is equivalent to (detailed proof in [194])

$$\mathbf{b}^{\mathrm{EN}}=\arg\min_{\mathbf{b}}\mathbf{b}^{\mathrm{T}}(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}+\gamma_2\mathbf{I}}{1+\gamma_2})\mathbf{b}-2\mathbf{y}^{\mathrm{T}}\mathbf{X}\mathbf{b}+\gamma_1||\mathbf{b}||_1. \tag{4.18}$$

4.2.5 Grouping Effect

In the 'large n, small m' problem, *i.e.* when there are many more variables than measurements [167], the 'grouped variables' issue is an especially important concern [194]. The grouping effect refers to the property of variables that are highly corrected being assigned similar regression coefficient values by a regression estimator. In extreme situations, if the variables are exactly identical, identical regression coefficients should be assigned to the identical variables [194]. Mathematically, the grouping effect can be expressed as follows. For the proofs, please refer to [194] for a a complete treatment.

A generic penalization method can be defined as

$$\widehat{\mathbf{b}}=\arg\min_{\mathbf{b}}||\mathbf{y}-\mathbf{X}\mathbf{b}||_2^2+\gamma J(\mathbf{b}), \tag{4.19}$$

where $J(\mathbf{b})>0$ for $\mathbf{b}\neq\mathbf{0}$.

Lemma 1 Assume that $\mathbf{x}_i=\mathbf{x}_j, i,j\in 1\ldots n$.

(a) If $J(\mathbf{b})$ is strictly convex, then $\widehat{b}_i=\widehat{b}_j, \forall\gamma>0$.

(b) If $J(\mathbf{b})=||\mathbf{b}||_1$, then $\widehat{b}_i\widehat{b}_j\geq 0$.

Lemma 2 The elastic net penalty is strictly convex, i.e.

$$\gamma_2||\mathbf{b}||_2^2+\gamma_1||\mathbf{b}||_1 \tag{4.20}$$

is strictly convex.

Lemma 1 shows a clear distinction between strictly convex penalty functions and the LASSO penalty. As shown in Lemma 2, the elastic net penalty is strictly convex, so it is guaranteed to produce identical regression coefficients for identical variables. In contrast, while grouping can occur with LASSO, it is not guaranteed.

4.2.6 Formulating Principal Component Analysis in a Ridge Regression Framework

In PCA, the PCs can be obtained by singular value decomposition of matrix $\mathbf{X}$, with $\mathbf{p}_i = \mathbf{v}_i$, where $\mathbf{v}_i$ is the i^{th} right singular vector of $\mathbf{X}$. Equivalently, the PC loadings can be computed as a ridge regression problem by solving the regression cost function [195]

$$J(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{m} ||\mathbf{z}_i - \mathbf{a}\mathbf{b}^{\mathrm{T}}\mathbf{z}_i||_2^2 + \gamma_2||\mathbf{b}||_2^2, \text{ s.t. } \mathbf{a}^{\mathrm{T}}\mathbf{a} = 1 \tag{4.21}$$

where $\mathbf{z}_i$ is the i^{th} column of matrix $\mathbf{Z}$, $\mathbf{Z} = \mathbf{X}^{\mathrm{T}}$ and the product of $\mathbf{a}\mathbf{b}^{\mathrm{T}}$ is an $n \times n$ matrix, which defines a rotation of $\mathbf{z}_i$. Specifically, Theorem 1 proposed by Zou [195] establishes that if

$$(\widehat{\mathbf{a}}^{\mathrm{R}}, \widehat{\mathbf{b}}^{\mathrm{R}}) = \arg\min_{\mathbf{a},\mathbf{b}} J(\mathbf{a}, \mathbf{b}), \tag{4.22}$$

the first loading can be obtained as

$$\mathbf{p}_1 = \widehat{\mathbf{b}}^{\mathrm{R}}(1 + \frac{\gamma_2}{\sigma_1^2}), \tag{4.23}$$

where σ_1 is the largest singular value of $\mathbf{X}$. An independent proof of this theorem is attached in the Appendix A.1.

Theorem 1 can be extended to the simultaneous computation of multiple PCs. If the first k PCs are required, the ridge regression cost function is expressed as [195]

$$J(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{m} ||\mathbf{z}_i - \mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{z}_i||_2^2 + \gamma_2 \sum_{j=1}^{k} ||\mathbf{b}_j||_2^2, \text{ s.t. } \mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{I}. \tag{4.24}$$

where $\mathbf{A} \in \mathbb{R}^{n \times k}$, $\mathbf{A} \in \mathbb{R}^{n \times k}$ and $\mathbf{b}_j$ is the j^{th} column of matrix $\mathbf{B}$. Computing the ridge estimate of $\mathbf{A}$ and $\mathbf{B}$

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} J(\mathbf{A}, \mathbf{B}), \tag{4.25}$$

the required PC loadings are obtained as

$$\mathbf{p}_j = \widehat{\mathbf{b}}_j(1 + \frac{\gamma_2}{\sigma_j^2}), \tag{4.26}$$

where σ_j is the j^{th} largest singular value.

4.2.7 Formulating Sparse Principal Component Analysis in the Elastic Net Regression Framework

The most direct approach to achieve sparse PCs using the elastic net regression framework is to exploit the regression relationship between scores and loadings in PCA, *i.e.* $\mathbf{t}_i = \mathbf{X}\mathbf{p}_i$ [143]. Letting $\mathbf{t}_i$ represent the output vector, a sparse loading, $\mathbf{b}_i$, can be computed as

$$\widehat{\mathbf{b}}_i = \arg\min_{\mathbf{b}_i} ||\mathbf{t}_i - \mathbf{X}\mathbf{b}_i||_2^2 + \gamma_2||\mathbf{b}_i||_2^2 + \gamma_1||\mathbf{b}_i||_1. \tag{4.27}$$

The drawback of this approach is that all solutions are constrained to be close to regular PCA.

An alternative and more general approach is to employ the regression formulation of PCA as outlined in Section 4.2.6 and to add a LASSO penalty to obtain sparsity. This was proposed as the elastic net for sparse PCA by Zou and Hastie [194] and is defined as:

$$(\widehat{\mathbf{A}}^{\text{EN}}, \widehat{\mathbf{B}}^{\text{EN}}) = (1+\gamma_2)\arg\min_{\mathbf{A},\mathbf{B}}\{\sum_{i=1}^{m}||\mathbf{z}_i - \mathbf{A}\mathbf{B}^{\mathtt{T}}\mathbf{z}_i||_2^2 + \gamma_2\sum_{j=1}^{k}||\mathbf{b}_j||_2^2 + \sum_{j=1}^{k}\gamma_{1j}||\mathbf{b}_j||_1\}, \quad \text{s.t. } \mathbf{A}^{\mathtt{T}}\mathbf{A} = \mathbf{I}. \tag{4.28}$$

The i^{th} sparse loading, $\widehat{\mathbf{p}}_i^{\text{S}}$, equals to

$$\widehat{\mathbf{p}}_i^{\text{S}} = \frac{\widehat{\mathbf{b}}_i^{\text{EN}}}{||\widehat{\mathbf{b}}_i^{\text{EN}}||_2}, \tag{4.29}$$

and the sparse scores, $\widehat{\mathbf{T}}^{\text{S}}$, equals to

$$\widehat{\mathbf{T}}^{\text{S}} = \mathbf{X}\widehat{\mathbf{P}}^{\text{S}}. \tag{4.30}$$

In Eq. 4.28, $\mathbf{Z} = \mathbf{X}^{\mathtt{T}}$ ($\mathbf{Z} \in \mathbb{R}^{n\times m}$), $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_m]$, and $\mathbf{z}_i = [z_{1i}, \ldots, z_{ni}]^{\mathtt{T}}$. $\mathbf{B}(\in \mathbb{R}^{n\times k}) = [\mathbf{b}_1, \cdots, \mathbf{b}_k]$ and $\mathbf{A} \in \mathbb{R}^{n\times k}$. In Eq. (4.28), $\mathbf{B}^{\mathtt{T}}\mathbf{z}_i$ yields the projection of $\mathbf{z}_i$

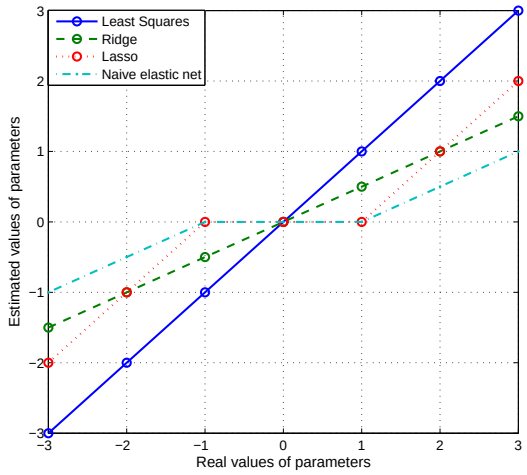


Figure 4.4: Solution for different estimators: $\gamma_1 = 2$ and $\gamma_2 = 1$

onto the principal axes (loading vectors) of $\mathbf{B}$. $\mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{z}_i$ takes the scores of $\mathbf{B}^{\mathrm{T}}\mathbf{z}_i$ and transforms them back into the original space. The orthogonality constraint on $\mathbf{A}$ tries to force $\mathbf{B}$ to be near orthogonal. The term $||\mathbf{z}_i - \mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{z}_i||_2^2$ measures the reconstruction errors for all $\mathbf{z}_i$. The functions of the LASSO and ridge constraints are the same as discussed in the elastic net regression, driving $\mathbf{B}$ to be sparse and avoiding the matrix singularity problem. γ_2 is selected for all loadings and γ_{1j} may be set to different values to allow more flexibility of penalisation to each individual loading. Note that if $\gamma_{1j} = 0\ \forall j$, the normalised columns of $\widehat{\mathbf{B}}^{\mathrm{EN}}$ are identical to the loadings obtained by regular PCA. A detailed mathematical proof of this result can be referred to [195].

Sparsity in the solutions is mainly achieved in two steps: ridge-type scaling followed by LASSO-type thresholding. As shown in Fig. 4.4, the ridge estimator (marked by a dashed line) is a scaled solution of the LS estimator (marked by a solid line). The LASSO estimator (marked by a dotted line) is a soft-thresholding solution of the LS estimator. The naive elastic net estimator (marked in the green line) combines the effect of ridge and LASSO estimators.

4.3 Numerical Solutions

The solutions of least squares and ridge regression, $\widehat{\mathbf{b}}$ and $\widehat{\mathbf{b}}^{\mathrm{R}}$, can be expressed as matrix algebra functions of $\mathbf{X}$, $\mathbf{y}$ and γ_2. This is not true for LASSO and elastic net.

Efron *et al.* [34] proposed a new regression method called least angle regression (LARS) to solve the LASSO problem, where S refers to its close relation to stagewise regression and LASSO. LARS starts with all coefficients at zero and successively adds more variables into solutions until variables are all included in which case, the least squares solution is obtained. Thus LARS provides solutions for all possible γ_1 for a given γ_2. A proper solution can for example be selected using cross validation or a *priori* selection of the number of variables [143].

LARS-EN is a computationally efficient algorithm for solving the elastic net regression problem. Instead of using all quantities as in LARS, LARS-EN records only the non-zero coefficients and the active variable set and uses them for calculation. To solve the elastic net principal component problem, Zou *et al.* proposed the general SPCA algorithm. Assuming $\mathbf{A}$ is known, $\mathbf{B}$ can be obtained by solving k independent elastic net problems. After $\mathbf{B}$ is obtained, $\mathbf{A}$ can be calculated using a singular value decomposition, *i.e.* if $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, then $\mathbf{A} = \mathbf{U}\mathbf{V}^{\mathrm{T}}$. These two steps are repeated until $\mathbf{B}$ converges. As proposed in [194], $\mathbf{A}$ is initiated as the first k loadings of regular PCA. Details of the general SPCA algorithm are as follows.

4.3.1 General SPCA Algorithm

The key to understanding the general SPCA algorithm is its relationship to the elastic net problem, which can be expressed as

$$\min_{\mathbf{B}}\{\sum_{i=1}^{m}||\mathbf{z}_i - \mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{z}_i||_2^2 + \gamma_2\sum_{j=1}^{k}||\mathbf{b}_j||_2^2 + \sum_{j=1}^{k}\gamma_{1j}||\mathbf{b}_j||_1\}. \tag{4.31}$$

If $\mathbf{A}$ is specified, the problem can be converted to solving k independent elastic net problems

$$\widehat{\mathbf{b}}_j = \arg\min_{\mathbf{b}_j}\{||\mathbf{Z}\mathbf{a}_j - \mathbf{Z}\mathbf{b}_j||_2^2 + \gamma_2||\mathbf{b}_j||_2^2 + \gamma_{1j}||\mathbf{b}_j||_1\},\ \text{for } j = 1, 2, \cdots, k. \tag{4.32}$$

or

$$\widehat{\mathbf{b}}_j = \arg\min_{\mathbf{b}_j} \mathbf{b}_j^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \gamma_2)\mathbf{b}_j - 2\mathbf{a}_j^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b}_j + \gamma_{1j}||\mathbf{b}_j||_1, \text{ for } j = 1, 2, \cdots, k. \quad (4.33)$$

If instead $\mathbf{B}$ is given, $\mathbf{A}$ can be obtained according to the following theorem.

Theorem 1. Let $\mathbf{A}$ and $\mathbf{B}$ be $n \times k$ matrices and $\mathbf{B}$ has rank k. Given the constrained maximization problem

$$\widehat{\mathbf{A}} = \arg\max_{\mathbf{A}} \mathrm{Tr}(\mathbf{A}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{B}) \text{ s. t. } \mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{I}, \quad (4.34)$$

if the SVD of $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{B}$ is $\mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, then $\mathbf{A} = \mathbf{U}\mathbf{V}^{\mathrm{T}}$.

The proof of Theorem 6 can be found in [195, 196]. The general SPCA algorithm can be expressed as follows:

Step 1: Initiate $\mathbf{A}$ as $\mathbf{P}[, 1 : k]$, the loadings of the first k ordinary principal components of $\mathbf{X}$ and set $\mathbf{Z} = \mathbf{X}^{\mathrm{T}}$.

Step 2: Given fixed $\mathbf{A}$, solve the naive elastic net problem for $j = 1, 2, \cdots, k$

$$\widehat{\mathbf{b}}_j = \arg\min_{\mathbf{b}_j} \mathbf{b}_j^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \gamma_2)\mathbf{b}_j - 2\mathbf{a}_j^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b}_j + \gamma_{1j}||\mathbf{b}_j||_1. \quad (4.35)$$

Step 3: With $\mathbf{B}$ fixed, compute the SVD of $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, then update $\mathbf{A} = \mathbf{U}\mathbf{V}^{\mathrm{T}}$.

Step 4: Repeat Steps 2 and 3, until $\mathbf{B}$ converges.

Step 5: Rescaling:

$$\widehat{\mathbf{p}}_j = \frac{\mathbf{b}_j}{||\mathbf{b}_j||_2}, \ j = 1, 2, \ldots, k.$$

Step 6: The sparse PCs are obtained as $\widehat{\mathbf{P}}^{\mathrm{S}} = [\widehat{\mathbf{p}}_1, \cdots, \widehat{\mathbf{p}}_k]$ and $\widehat{\mathbf{T}}^{\mathrm{S}} = \mathbf{X}\widehat{\mathbf{P}}^{\mathrm{S}}$.

Some remarks:

1. In Step 2, $\mathbf{b}_j$ is estimated using LARS-EN.
2. γ_2 is used to address the matrix singularity problems. Hence, it can be set to zero if there is no need to do so. The authors in [195, 196] show that $\widehat{\mathbf{b}}_j$ changes slowly with changes in γ_2.

3. γ_{1j} is the key parameter for determining the sparsity in the j^{th} loading. The larger the value of γ_{1j}, the more coefficients of the loading are restricted to zero, that is, a higher level of sparsity is achieved. At the expense of sparsity, less variance can be captured in each sparse component. Therefore, there exists a trade-off between sparsity and variance explained, which must be taken into account when determining the approximate level of sparsity for a given problem.

4.3.2 Soft Thresholding SPCA Algorithm

Theoretically, the general SPCA algorithm is applicable to all kinds of data. However, when the number of the variables is much larger than the number of the observations ($n \gg m$), the computational cost is very high. The soft thresholding SPCA algorithm in [196] is proposed as a simplified implementation for this class of problem.

The special case of the elastic net occurs when $\gamma_2 \to \infty$. By Eq. (4.18), $\widehat{\mathbf{b}} \to \widehat{\mathbf{b}}(\infty)$ as $\gamma_2 \to \infty$, where

$$\widehat{\mathbf{b}}(\infty) = \arg\min_{\mathbf{b}} \mathbf{b}^{\mathrm{T}}\mathbf{b} - 2\mathbf{y}^{\mathrm{T}}\mathbf{X}\mathbf{b} + \gamma_1||\mathbf{b}||_1. \tag{4.36}$$

The solution to Eq. (4.36) can be expressed as

$$\widehat{b}(\infty)_i = (|\mathbf{y}^{\mathrm{T}}\mathbf{x}_i| - \frac{\gamma_1}{2})^+ sign(\mathbf{y}^{\mathrm{T}}\mathbf{x}_i),\ i = 1, 2, \cdots, n, \tag{4.37}$$

where $\mathbf{y}^{\mathrm{T}}\mathbf{x}_i$ is the univariate regression coefficient of the i^{th} predictor. Comparing Eq. (4.36) and Eq. (4.35), it can be shown that the soft thresholding solutions of the sparse components [195] are equal to

$$\widehat{b}_{ij} = (|\mathbf{a}_j^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{x}_i| - \frac{\gamma_{1j}}{2})^+ \mathrm{sign}(\mathbf{a}_j^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{x}_i),\ i = 1, 2, \cdots, n. \tag{4.38}$$

The soft-thresholding algorithm provides a gentle transition-in of the function from one stage to the other as illustrated in Figure 4.5. A consequence of employing this special case is that correlation between variables is ignored in computing the sparse components [194]. However, empirical evidence suggests that this does not significantly impact on performance [196].

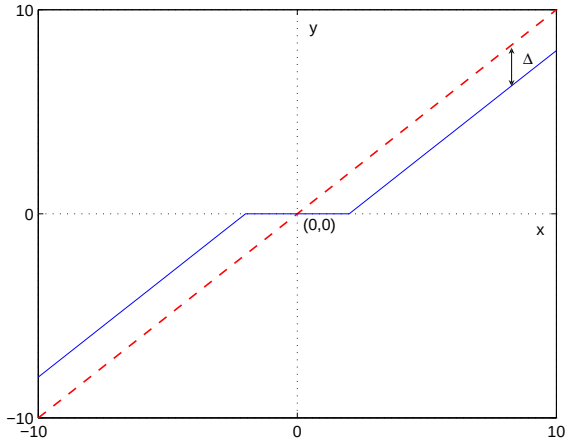


Figure 4.5: An illustration of soft-thresholding estimation $y = (|x| - \Delta)^{+}\text{sign}(x)$ with $\Delta = 2$.

4.4 Variance Explained by the Sparse Principal Components

4.4.1 Adjusted Variance Explained

SPCA employs a similar approach to PCA, known as adjusted variance [195], to measure the estimation accuracy. However, unlike PCA, $\widehat{\mathbf{P}}^{\mathrm{S}}$ is not orthogonal, so the variances explained by individual PCs are not independent of each other. Consequently the approach normally used for calculating the variance in PCA is not valid for SPCA. To address this issue, the sparse scores matrix needs to be orthogonalised. Thus the complete variance estimation algorithm can be expressed as follows [46, 143].

- Orthogonalize $\widehat{\mathbf{t}}_j^{\mathrm{S}}$ (the j^{th} column vector of $\widehat{\mathbf{T}}^{\mathrm{S}}$ defined in Eq. (4.30)) by applying the recursion

$$\widehat{\mathbf{t}}_j^{\mathrm{S}*} = \widehat{\mathbf{t}}_j^{\mathrm{S}} - \widehat{\mathbf{T}}_{(j-1)}^{\mathrm{S}}[(\widehat{\mathbf{T}}_{(j-1)}^{\mathrm{S}})^{\mathrm{T}}(\widehat{\mathbf{T}}_{(j-1)}^{\mathrm{S}})]^{-1}(\widehat{\mathbf{T}}_{(j-1)}^{\mathrm{S}})^{\mathrm{T}}\widehat{\mathbf{t}}_j^{\mathrm{S}},$$

  for $j = 1, \ldots, k$ (k is the number of sparse principal components), where $\widehat{\mathbf{T}}_{(j)}^{\mathrm{S}} = [\widehat{\mathbf{t}}_{(1)}^{\mathrm{S}}, \ldots, \widehat{\mathbf{t}}_{(j)}^{\mathrm{S}}]$.

- Collect the orthogonalized vectors into a matrix $\widehat{\mathbf{T}}^{\mathrm{S}*}$, *i.e.*

$$\widehat{\mathbf{T}}^{\mathrm{S}*} = [\widehat{\mathbf{t}}_1^{\mathrm{S}*}, \ldots, \widehat{\mathbf{t}}_j^{\mathrm{S}*}, \ldots, \widehat{\mathbf{t}}_p^{\mathrm{S}*}]. \tag{4.39}$$

- Compute the variance explained (V_e) by the k sparse components as

$$V_e = \text{trace}\{(\widehat{\mathbf{T}}^{\text{S}^*})^{\text{T}}\widehat{\mathbf{T}}^{\text{S}^*}\}. \tag{4.40}$$

The j^{th} diagonal entry of $(\widehat{\mathbf{T}}^{\text{S}^*})^{\text{T}}\widehat{\mathbf{T}}^{\text{S}^*}$ corresponds to the variance explained by the j^{th} sparse component.

4.4.2 SPMSE

When using sparse PCs to reconstruct a data set, many columns of the reconstructed data are in fact zero, because of the zero elements in the sparse loadings. Therefore, a fairer assessment of the accuracy of reconstruction is to only compare the reconstruction against the original data over the regions where the reconstruction exists. Here, a sparse mean square error measure is proposed, denoted SPMSE, where S stands for sparse, P for percentage and MSE for mean square error. This is given by

$$SPMSE = \frac{||\widehat{\mathbf{X}}_{\text{s}} - \mathbf{X}_{\text{s}}||_f^2}{||\mathbf{X}_{\text{s}}||_f^2} \times 100\%, \tag{4.41}$$

where $||\cdot||_f$ is the Frobenius norm, $\widehat{\mathbf{X}}_{\text{s}}$ consists of the nonzero columns of the reconstructed data matrix $\widehat{\mathbf{X}}$ and $\mathbf{X}_{\text{s}}$ is the corresponding subset of the original data matrix $\mathbf{X}$. Based on SPMSE, SV_e, variance explained by the sparse components, is proposed as

$$SV_e = 100\% - SPMSE. \tag{4.42}$$

In contrast to V_e (Eq. 4.40), SV_e only calculates the variance for the non-zero reconstructed channels, so SV_e can more effectively reflect the reconstruction accuracy. Note, that since the sparse components are not orthogonal, the reconstruction of $\mathbf{X}$ is defined as

$$\widehat{\mathbf{X}} = \widehat{\mathbf{T}}^{\text{S}}(\underline{\widehat{\mathbf{P}}}^{\text{S}})^{\text{T}}[\underline{\widehat{\mathbf{P}}}^{\text{S}}(\underline{\widehat{\mathbf{P}}}^{\text{S}})^{\text{T}}]^{-1}. \tag{4.43}$$

4.5 Study of SPCA Properties on Artificial Data

In this section, the properties of EN-SPCA, sparsity and the grouping effect, are discussed with the aid of artificial data sets. Note that when applying EN-SPCA to analyse data sets different solution methods can be employed for different problem dimensions

leading to different representations of the tunable L_1 and L_2 parameters. Specifically when using the general SPCA algorithm, the L_1, L_2 parameters are expressed as γ_2, c_{1j} while for the soft thresholding SPCA algorithm, they are expressed as ∞, γ_{1j}.

4.5.1 Generation of Data Set

The idea for constructing an artificial data set comes from Zou *et al.* [195]. Here, we will employ similar datasets to test the properties of SPCA. The designed data set, denoted by $\mathbf{Z}$, consists of 1000 observations of 10 variables. Let

$$\mathbf{d}_1 \sim N(0, 100),\ \mathbf{d}_2 \sim N(0, 121),\ \text{and}\ \mathbf{d}_3 = -0.2\mathbf{d}_1 + 0.9\mathbf{d}_2,$$

where $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3 \in \mathbb{R}^{1000\times 1}$ and $N(m, v)$ indicates that the samples are drawn from a normal distribution with mean, m, and variance, v. The data set is then constructed as three groups of highly correlated variables as follows

$$\mathbf{z}_i = \mathbf{d}_1 + \mathbf{e}_i^1,\ i = 1, 2, 3, 4 \tag{4.44}$$

$$\mathbf{z}_i = \mathbf{d}_2 + \mathbf{e}_i^2,\ i = 5, 6, 7, 8 \tag{4.45}$$

$$\mathbf{z}_i = \mathbf{d}_3 + \mathbf{e}_i^3,\ i = 9, 10 \tag{4.46}$$

where $\mathbf{e}_i^j$ $(j = 1, 2, 3)$ are independent identically distributed noise sequence, $\mathbf{e}_i^j \sim N(0, 1)$. Thus, $\{\mathbf{z}_i, i = 1, 2, 3, 4\}$ are uncorrelated with $\{\mathbf{z}_i, i = 5, 6, 7, 8\}$. By design, $\{\mathbf{z}_i, i = 9, 10\}$ are highly correlated with $\{\mathbf{z}_i, i = 5, 6, 7, 8\}$, apart from the high correlation between themselves. The reason for designing the variables to be correlated in this fashion is to examine the grouping effect.

4.5.2 Sparsity

Because the generated artificial data matrix is non-singular, γ_2 can be assigned to zero. As such, only the tuning parameter for LASSO needs to be specified. Considering the size of the data set is small, the general SPCA algorithm is employed, leaving c_{1j} to be specified. In table 4.1, the upper bounds of the LASSO constraint (c_{1j}) are set to 2, 1,8 and 1.3 for the first three loadings, respectively. The sparsity is measured as the number of the nonzero elements in each loading. As can be observed the number of nonzero elements in the first three sparse loadings are 4, 4 and 2, respectively, as

	SPCA ($\gamma_2 = 0$)			**PCA**		
Order of Variables	PC1	PC2	PC3	PC1	PC2	PC3
1	0	-0.5642	0	0.0719	0.4929	0.6392
2	0	-0.4769	0	0.0707	0.4912	0.1096
3	0	-0.4945	0	0.0724	0.4911	-0.4620
4	0	-0.4580	0	0.0721	0.4902	0.0027
5	-0.4902	0	0	-0.4125	0.0893	-0.2555
6	-0.4828	0	0	-0.4134	0.0950	0.1768
7	-0.5453	0	0	-0.4139	0.0908	-0.3624
8	-0.4788	0	0	-0.4127	0.0932	0.0861
9	0	0	-0.7054	-0.3846	-0.0115	0.3599
10	0	0	-0.7089	-0.3859	-0.0177	0.0346
Number of Nonzero Elements	4	4	2	10	10	10
Variance Explained (%)	46.74	32.21	0.13	67.03	32.24	0.10
Cumulated Variance (%)	46.74	78.95	79.08	67.03	99.27	99.37

Table 4.1: Comparison between PCA and SPCA: Loadings, Sparsity and Variance. The upper bounds of the LASSO constraint for the sparse PCs are 2, 1.8 and 1.2, respectively.

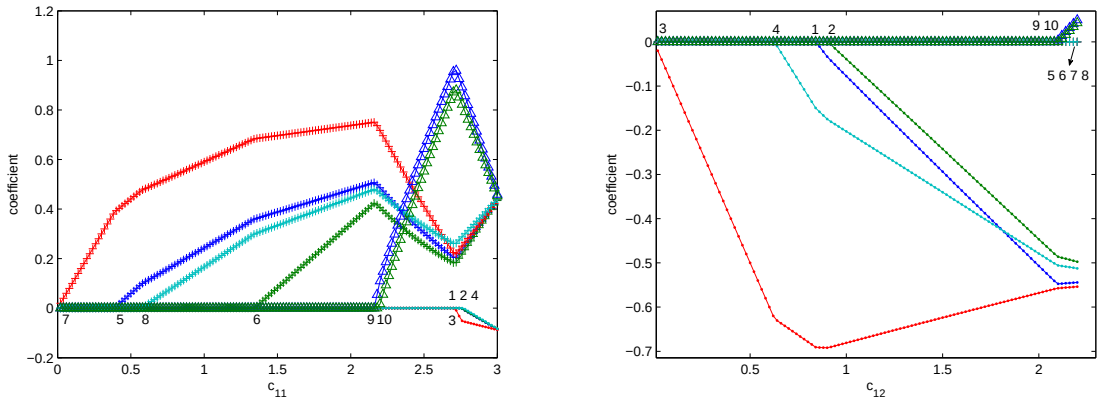
compared to 10, 10 and 10 for the regular PC loadings.

The variance explained by the first sparse component (46.74%) is much less than that (67.03%) of the first regular PC. For the second PC, the variance explained is nearly equal and for the third PC, more variance is in fact explained by the corresponding sparse component. Thus it is possible that individual sparse PCs may have greater variance explained than corresponding PCA components, but the accumulative variance explained is much less.

Table 4.2 shows that relaxing the constraint on the upper bound of the LASSO constraint (*i.e.* larger c_{1j}), more nonzero coefficients are obtained and as a consequence, more variance is captured, accumulative variance in particular. Note that since $\gamma_2 = 0$, *i.e.* the penalty term is not strictly convex, the grouping effect is not fully guaranteed.

SPCA ($\gamma_2 = 0$)						
	Upper bound of LASSO constraint for the first 3 PCs					
	c_{11}	c_{12}	c_{13}	c_{11}	c_{12}	c_{13}
Order of Variables	2	1.8	1.3	2.3	2.1	1.8
1	0	-0.5642	0	0	-0.5070	-0.1496
2	0	-0.4769	0	0	-0.4967	0.0074
3	0	-0.4945	0	0	-0.4983	0
4	0	-0.4580	0	0	-0.4935	0.0618
5	-0.4902	0	0	-0.4624	0	0
6	-0.4828	0	0	-0.3857	-0.0141	-0.2468
7	-0.5453	0	0	-0.5867	0	0.4247
8	-0.4788	0	0	-0.4609	-0.0025	0
9	0	0	-0.7054	-0.0840	0.0265	-0.8414
10	0	0	-0.7089	-0.2715	0.0588	-0.1566
Number of Nonzero Elements	4	4	2	6	8	7
Variance Explained (%)	46.74	32.21	0.13	57.70	32.64	0.11
Cumulated Variance (%)	46.74	78.95	79.08	57.70	90.34	90.45

Table 4.2: Effect of the LASSO constraint on Sparsity. Increasing the upper bound, c_{1j}, (*i.e.* relaxing the constraint) allows more variables to be included in each PC.



(a) (b)

Figure 4.6: Elastic net solution paths ($\gamma_2 = 0.1$): (a) Changes in the first sparse loading coefficients as a function of c_{11}; (b) Changes in the second sparse loading coefficients as a function of c_{12} ($c_{11} = 2$) (the three groups of variables are marked by '.','+' and ' $\triangle$ ', respectively).

4.5.3 Grouping Effect

An effective way to investigate the grouping effect is to explore the solution path as function of the sparsity constraint. As is required for guaranteeing the grouping effect, γ_2 is set to 0.1. The solution path for the first sparse loading as a function of c_{11} is shown in Fig. 4.6 (a), for the range 0 to 3. The figure shows a clearly 'grouped selection', *i.e.* variable 7, 5, 8 and 6 in the first group, variable 9 and 10 in the second group and variable 3, 1, 2 and 4 in the last group. When $c_{11} = 3$, the constraint is too relaxed to restrict the coefficients, leading to the solution equivalent to regular PCA. Setting $c_{11} = 2$, the solution path for the second sparse loading as a function of c_{12} is shown in Fig. 4.6 (b). As can be observed, the second sparse component shows the same 'grouped selection' as the first sparse component.

To further highlight the grouping effect, it is useful to consider the case where the data set contains identical variables. As such, the artificial data is revised so that the noise is removed from the first four variables making them identical, *i.e.* $\mathbf{z}_i = \mathbf{d}_1, i = 1, 2, 3, 4$. Table 4.3 gives the experimental results of applying EN-SPCA to the artificial data, where as an example, the tuning parameters are set as $\gamma_2 = 1$ and $c_{11} = 3.5, c_{12} =$

SPCA ($\gamma_2 = 1$)			
	Upper bound of		
	LASSO constraint		
	PC_1	PC_2	PC_3
Order of Variables	$c_{11} = 3.5$	$c_{12} = 3.5$	$c_{13} = 0.1$
1	0	-0.5	0
2	0	-0.5	0
3	0	-0.5	0
4	0	-0.5	0
5	0.5098	0	0
6	0.2808	0	1
7	0.6782	0	0
8	0.4487	0	0
9	0	0	0
10	0	0	0
Number of Nonzero Elements	4	4	1
Variance Explained (%)	39.94	37.48	0.001
Accumulated Variance (%)	39.94	77.42	77.52

Table 4.3: Grouping effect with identical variables contained in the data set

$3.5, c_{13} = 0.1$ for the first three sparse components, respectively. As can be seen, in the second sparse component, identical coefficients are assigned to the first four variables, while for variable 5, 6, 7 and 8, although selected at the same time, the obtained coefficients are not identical.

4.6 EN-SPCA Applied to SDS1

4.6.1 PCA, A Special Case of SPCA

Theoretically, when giving no constraint to the LASSO penalty term, SPCA should be equivalent to PCA. Provided the sparse loadings are scaled to unit length, the solutions of PCs should be identical. The experimental results shown in Table 4.4 provides a comparison between PCA and SPCA in terms of variance and the number of nonzero

	Variance Explained (%)		Number of Nonzero Elements	
Order of PCs	PCA	SPCA	PCA	SPCA
PC1	47.27	47.27	20	20
PC2	39.6	39.6	20	20
PC3	10.89	10.89	20	20

Table 4.4: PCA, a special case of SPCA ($\gamma_2 = 1, c_{11} = c_{12} = c_{13} = 4$)

elements. As can be seen, each PC explains the same variance and contains the same number of nonzero elements. Plotting the first three PCs (in Fig. 4.7) also shows that there is no difference between the PCs obtained by either SPCA or PCA.

4.6.2 Selecting the Tuning Parameters

In this section, the L_2 penalty tuning parameter (γ_2) is set to 1 (to guarantee the grouping effect) leaving only the L_1 penalty parameters (c_{1j} - one for each sparse component, j=1 ... k) to be determined experimentally. These parameters essentially determine the sparseness of the corresponding components.

As an example, Fig. 4.8 (a) demonstrates the relationship between the number of nonzero elements (N_{NE}) and variance explained (V_e) for the first sparse component for SDS1. It provides a useful guide for making a judgment call on the trade-off between sparsity and variance explained. As can be seen, in this instance there is a significance increase in variance in the graph at $N_{NE} = 8$ and $V_e = 33.31\%$. This corresponds to choosing c_{11} as 4.91. Similarly, for the second sparse component, as shown in Fig. 4.8 (b), there is a more rapid increase in variance at $N_{NE} = 5$ than any other points. This corresponds to choosing c_{12} as 3.91 and $V_e = 18.05\%$.

One difficulty that arises when selecting the constraints for each component is that they cannot be done independently. When selecting the second sparse component, the solution to the first component varies even though the parameter for the first sparse component is fixed. This is illustrated in Fig. 4.9 which shows the changes in the number of nonzero elements and variance explained by the first sparse component as

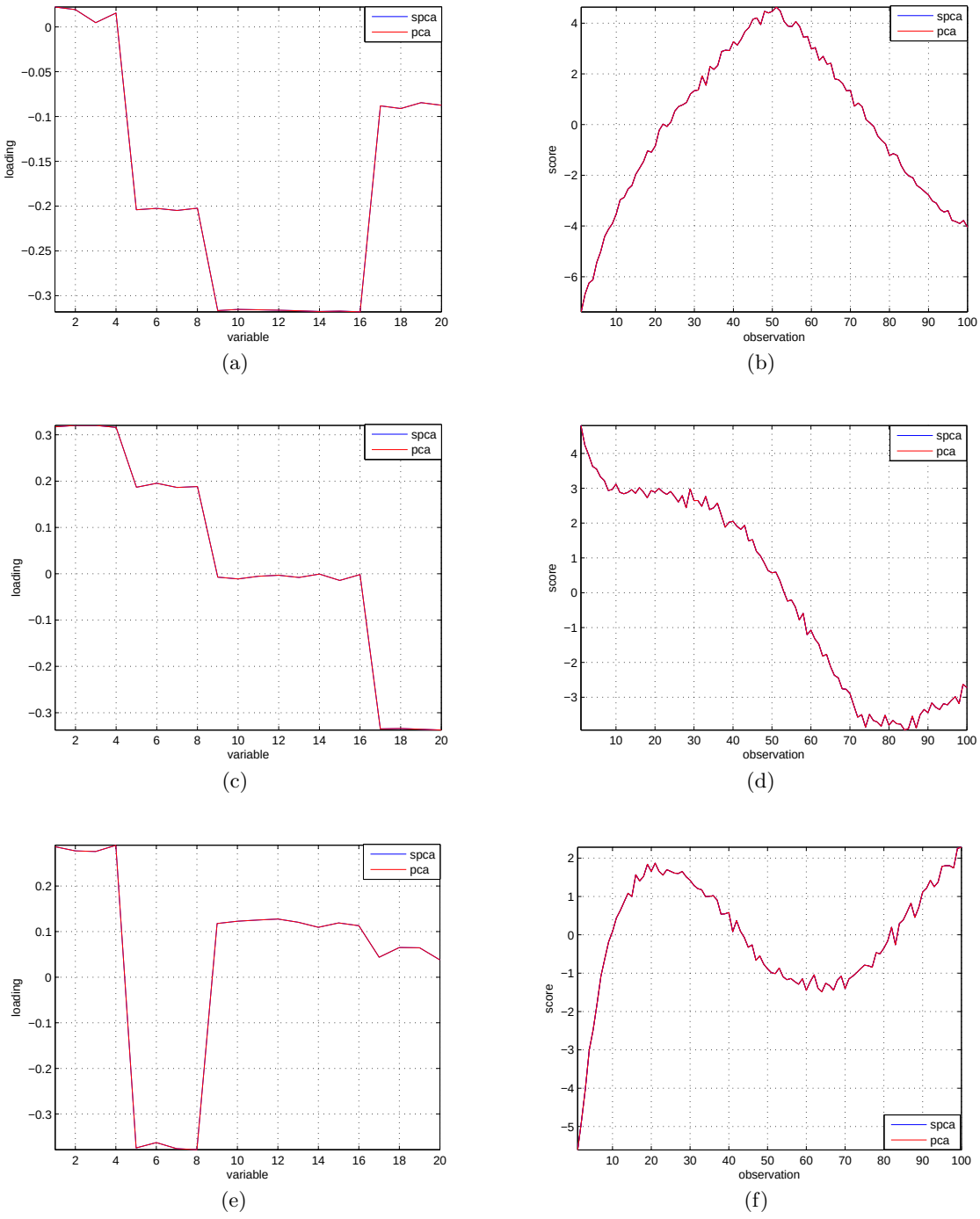


Figure 4.7: The first 6 loadings of PCA and SPCA ($\gamma_2 = 1, c_{11} = c_{12} = c_{13} = 4$): (a) The first loading; (b) The first score; (c) The second loading; (d) The second score; (e) The third loading; (f) The third score.

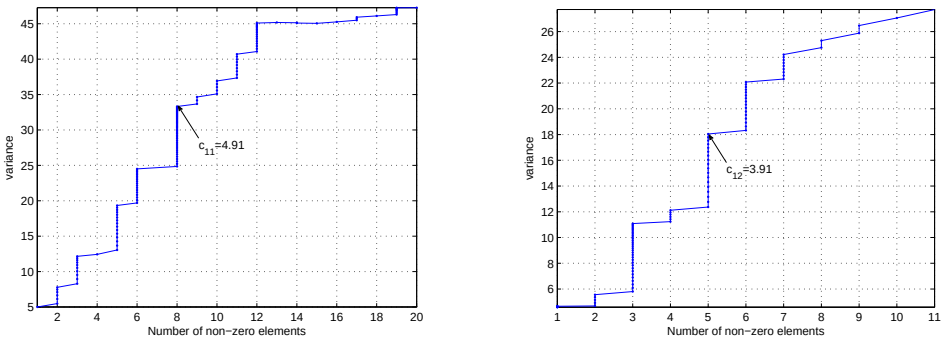
a function of the tuning parameter for the second sparse component with the tuning parameter for the first sparse component fixed ($c_{11} = 4.91$). The same happens with the selection of the third sparse component, that is, even if the tuning parameters for the first two parameters are fixed, the solutions for the first two also vary. This was not highlighted by Zou *et. al.* [196] in their work. This makes EN-SPCA difficult to tune and unreliable for practical applications.

4.7 Experiments on OES Data

In this section, IDS1 and IDS1Filt, are employed to explore the application of EN-SPCA to the analysis of OES data. In the OES data sets, the number of variables is much bigger than the number of observations ($m = 90, n = 2046$). The general SPCA algorithm can be applied in this situation if a positive γ_2 is selected, but the computational cost is very high. As an alternative, soft thresholding is proposed [196]. Soft thresholding is in essence the general SPCA algorithm with $\gamma_2 = \infty$. However, soft thresholding is computationally efficient. Thus when applying EN-SPCA to OES, soft thresholding is the preferred implementation. In this form only the tuning parameters for LASSO, *i.e.*, γ_{1j} need to be specified.

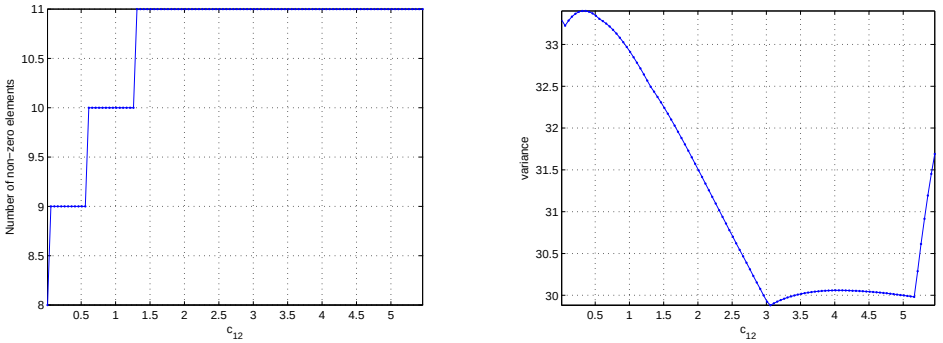
4.7.1 Selecting the LASSO Tuning Parameters

Table 4.5 shows how the number of nonzero elements (N_{NE}) varies as a function of γ_{11}, for the first sparse component for IDS1. The table also shows how the variance explained by the first sparse component changes as measured by the adjusted variance (V_e) and SPMSE (SV_e). As might be expected, V_e decreases as more and more coefficients are forced to zero. In contrast, the SV_e value remains large for all values of γ_{11} demonstrating that the sparse component achieves good accuracy for those channels where a reconstruction exists.



(a) (b)

Figure 4.8: Selection of c_{1j} according to the relationship between the variance and number of nonzero elements: (a) The first sparse component; (b) The second sparse component.



(a) (b)

Figure 4.9: An illustration of constraint parameter selection dependency: (a) Changes in the number of nonzero elements for the first sparse component as a function of the tuning parameter for the second sparse component; b) Changes in the variance explained for the first sparse component as a function of the tuning parameter for the second sparse component.

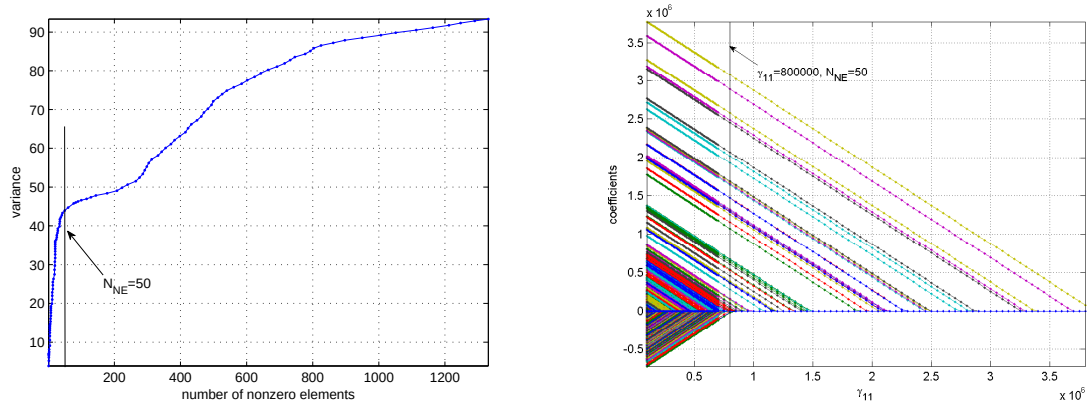
γ_{11}	N_{NE}	$V_e(\%)$	$SV_e(\%)$
640000	144	47.85	79.05
650000	129	47.38	80.49
660000	116	46.97	81.84
670000	100	46.60	83.99
680000	90	46.29	85.31
690000	83	46.03	86.40
700000	77	45.79	87.18
750000	60	44.73	89.13
800000	**50**	**43.94**	**90.39**
850000	43	43.32	91.41
900000	41	42.73	90.93
950000	38	42.16	91.01
1000000	35	41.67	91.44
1050000	35	41.12	90.24
1100000	34	40.52	89.51
1150000	34	39.86	88.06
1200000	29	39.29	90.39

Table 4.5: N_{NE}, V_e and SV_e, corresponding to different γ_{11} values for the first sparse PC

Fig. 4.10 (a) demonstrates the corresponding relationship between N_{NE} and V_e and provides a useful guide for making a judgment call on the trade-off between sparsity and variance explained. As can be seen, in this instance there is a 'knee' in the graph at $N_{NE} = 50$, beyond which there is a marked decrease in the rate of variance increase with included variables. This corresponds to choosing γ_{11} as 800000. The solution path of the regression coefficients as a function of γ_{11} is shown in Fig. 4.10 (b). As can be seen as γ_{11} is increased, more coefficients are constrained to zero.

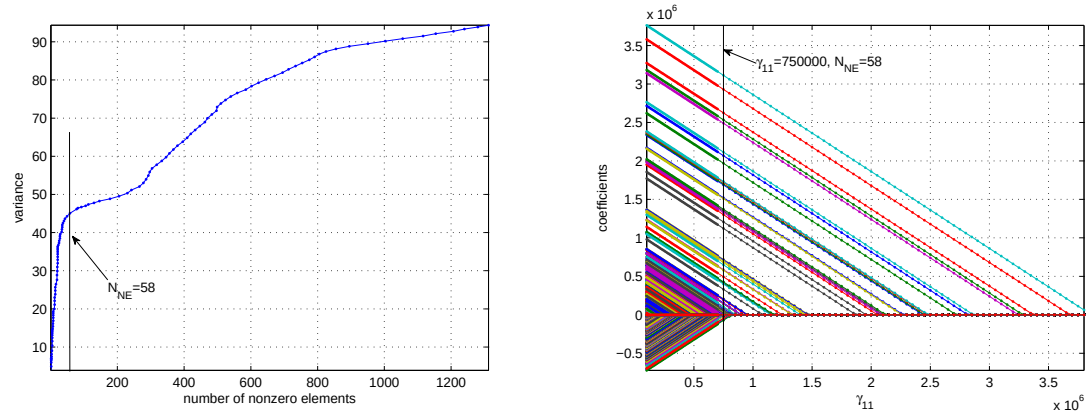
The variance explained by the sparse component with $\gamma_{11} = 800000$ is 43.94%, which compares to 95.82% for the unrestricted principal component. The corresponding

SV_e value, which measures the reconstruction accuracy on the non-zero reconstructed-channels, is 90.39%.



(a) (b)

Figure 4.10: Selecting γ_{11} for IDS1: (a) The relationship between N_{NE} and V_e; (b) The solution path as a function of γ_{11}.



(a) (b)

Figure 4.11: Selecting γ_{11} for IDS1Filt: (a) The relationship between N_{NE} and V_e; (b) The solution path as a function of γ_{11}.

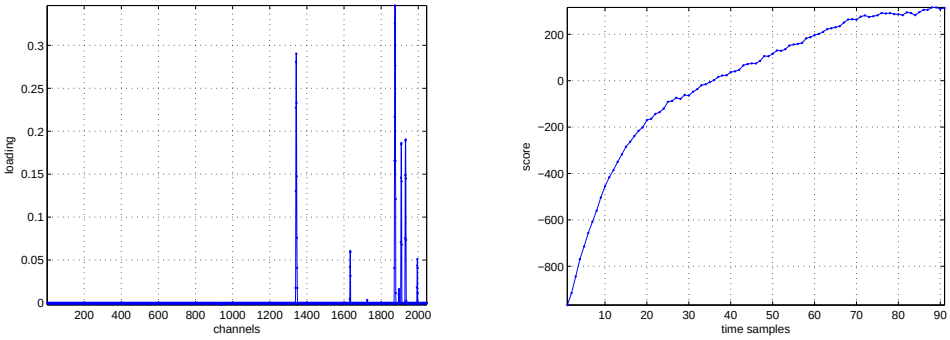
The corresponding relationship between N_{NE} and V_e for IDS1Filt is shown in Fig. 4.11 (a). A 'knee' in the graph occurs at $N_{NE} = 58$, beyond which there is a marked decrease in the rate of variance increase with included variables. This corresponds to choosing

γ_{11} as 750000 and $V_e = 45.07\%$. The solution path of the regression coefficients as a function of γ_{11} is shown in Fig. 4.11 (b). As can be seen, the results for IDS1 and IDS1Filt are similar.

4.7.2 Grouping Effect

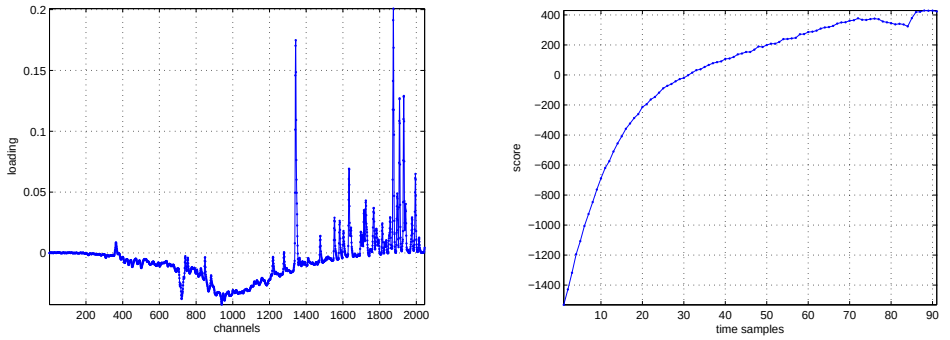
Figure 4.12 shows the distribution of the first sparse component computed with $\gamma_{11} = 800000$ for IDS1. For comparison purposes the distribution of the first regular PC is included in Figure 4.13. The PCA loading elements are all non-zero with several clusters of large values centered on the active OES channels. These clusters arise because of the spectral bleed between adjacent channels. In contrast, the SPCA loading has only 50 non-zero entries and these are in seven distinct clusters of points. Analysis of these seven sets of points show that they are all highly correlated as can be seen in Fig. 4.15 (a) (|correlation coefficient| > 0.925). This is a direct consequence of the grouping effect that is a feature of EN-SPCA.

The first sparse component for IDS1Filt is shown in Fig. 4.14. The sparse loading has 58 non-zero elements, distributed in nine distinct clusters. The patterns of these 58 variables are shown in Fig. 4.15 (b). Correlation analysis of these variables shows that the absolute value of the lowest correlation coefficient between any two variables is 0.9044. Table 4.6 gives a comparison of the channel distribution of the nonzero elements in the first sparse component between IDS1 and IDS1Filt. The channels that are not present in both components are highlighted in bold.



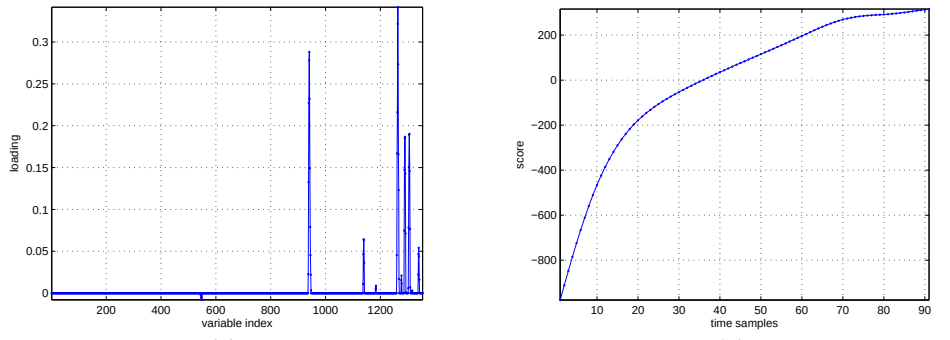
(a) (b)

Figure 4.12: The first sparse principal component obtained for IDS1 using soft thresholding with $\gamma_{11} = 800000$: (a) Sparse loading; (b) Sparse score.



(a) (b)

Figure 4.13: The first principal component of IDS1: (a) PC loading; (b) PC score.



(a) (b)

Figure 4.14: The first sparse principal component obtained for IDS1Filt using soft thresholding with $\gamma_{11} = 800000$.

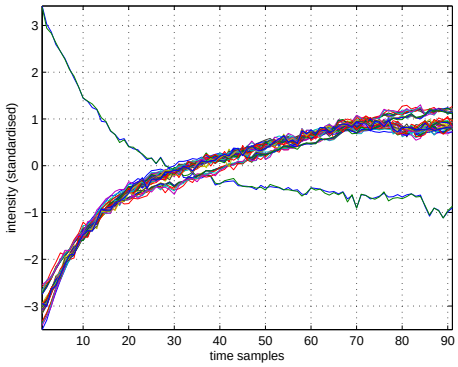


Figure 4.15: Intensity changes of the nonzero loadings (over time) for the first sparse PC for IDS1.

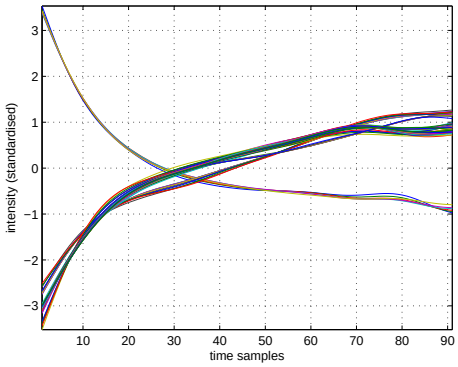


Figure 4.16: Intensity changes of the nonzero loadings (over time) for the first sparse PC for IDS1Filt.

Data	Nonzero channels in the first sparse component
IDS1	939, 940, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1346, 1347, 1348, 1631, 1632 1633, 1634, 1635, 1724, 1725, 1871, 1872, 1873, 1874, 1875, 1876, 1877, 1878, 1879 1895, 1896, 1897, 1906, 1907, 1908, 1909, 1910, 1911, 1928, 1929, 1930, 1931, 1932 1933, 1934, 1935, 1993, 1994, 1995, 1996, 1997
IDS1Filt	**937, 938**, 939, 940, **941, 942**, 1339, 1340, 1341, 1342, 1343, 1344, 1345, 1346 1347, 1348, **1349**, 1631, 1632, 1633, 1634, 1635, 1724, 1725, 1871, 1872, 1873, 1874 1875, 1876, 1877, 1878, 1879, 1895, 1896, 1897, 1906, 1907, 1908, 1909, 1910, 1911 **1912**, 1928, 1929, 1930, 1931, 1932, 1933, 1934, 1935, **1941, 1942**, 1993, 1994 1995, 1996, 1997

Table 4.6: Comparing the distribution of nonzero channels in the first sparse component of IDS1 and IDS1Filt.

The above analysis on IDS1 and IDS1Filt shows that EN-SPCA has a tendency to give equal weighting to strongly correlated variables and, as such, selects all the correlated variables as a group, rather than selecting a single representative example. This is useful when trying to identify groups of related variables, but is not ideal for a variable selection algorithm. Note that although the variables are highly reversely correlated, they still occur in the same sparse loading. Thus EN-SPCA is not effective for separating variables with reverse patterns.

4.8 Discussion and Conclusions

This chapter has introduced SPCA as a variable selection tool for the identification of key variables in large data sets. EN-SPCA, one of the true algorithmic methods for calculating sparse loadings has been introduced, together with a series of relevant topics namely least squares, ridge, LASSO and elastic net regressions. SPMSE has been proposed as a measure that better reflects the estimation accuracy of SPCA, given the sparse structure of the model and the issue of selecting the tuning parameters has been proposed as a tradeoff between variance explained and a sparse representation.

With the aid of an artificial data set and SDS1, the main properties of SPCA have been illustrated, particularly in relation to how it is able to provide sparse solutions to PC loadings and at the same time maintain the grouping effect. Results show that SPCA is useful for key variable extraction and is able to identify the variables that are highly correlated. However, one flaw in EN-SPCA has been spotted. In particular, it is highlighted that the sparse solutions are not stable, given changes to the tuning parameters of the following sparse components, the solution for the existing sparse component will be altered. Applying EN-SPCA to OES data from a plasma chamber shows that a small number of variables can be recognised using SPCA, making it easier for variable interpretation (relating variables to process chemicals) and data visualisation. However, it is not effective at separating the variables with reverse patterns and hence is not ideal for variable selection based on pattern differences.

Chapter 5

Adaptive Weighting SPCA

5.1 Motivation

As discussed in the last chapter, it is clear that the LASSO penalty (L_1 penalty) can provide a sparse solution to loadings in regular PCA. However, the problem is that using the same tuning parameter to penalize all coefficients in the same loading ignores the intrinsic difference between loading coefficients. Addressing this issue, Zou [193] and Wang and Leng [164] have proposed the adaptive LASSO penalty. The drawback of the adaptive LASSO is that it cannot guarantee the grouping effect, a key attribute of EN-SPCA, or the loading orthogonality, a key attribute of regular PCA. These two properties are important for appropriate variable selection. Tackling these problems leads to the proposal of adaptive weighting SPCA (AWSPCA).

The rest of the chapter is organized as follows. First, the methodology of AWSPCA is described. Topics covered include designing the adaptive weightings, the optimization criterion for achieving AWSPCA, numerical solutions and selecting tuning parameters. Then, experiments based on a simulated data set are used to illustrate the properties of AWSPCA. Finally, experimental results are presented for the application of AWSPCA to semiconductor OES data.

5.2 Methodology

5.2.1 Adaptive LASSO Penalty

The adaptive LASSO estimate provides the possibility of introducing more flexibility to the control of loading sparsity, The adaptive LASSO estimate of the regression coefficients can be expressed as

$$\widehat{\mathbf{b}}_j^{\mathrm{AL}} = \arg\min_{\mathbf{b}_j} ||\mathbf{y} - \mathbf{X}\mathbf{b}_j||_2^2 + \gamma_{1j} \sum_{i=1}^{n} w_{ij}||b_{ij}||_1, \tag{5.1}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $\mathbf{y} \in \mathbb{R}^{m\times 1}$ are the regression inputs and outputs, respectively. $\mathbf{b}_j \in \mathbb{R}^{n\times 1}$ is the regression coefficient vector with the i^{th} element denoted as b_{ij}. In the framework of EN-SPCA, $\mathbf{b}_j$ corresponds to the j^{th} loading. γ_{1j} is a nonnegative tuning parameter of the adaptive LASSO penalty for the j^{th} loading and w_{ij} is the adaptive weighting, used to penalize b_{ij}. According to [193], w_{ij} can be designed as

$$w_{ij} = 1/|p_{ij}|, \tag{5.2}$$

where p_{ij} denotes the i^{th} element in the j^{th} loading in regular PCA. Hence, there is an inverse relationship between w_{ij} and p_{ij}. For example, given large p_{ij}, w_{ij} is small, leading to less penalisation to the corresponding regression coefficient. The advantage of such a design is that w_{ij} can be easily obtained as a reverse value of p_{ij}, leaving only the value of γ_{1j} to be specified for each loading. This greatly simplifies the tuning parameter specification. However, the adaptive LASSO estimate cannot guarantee the grouping effect (selecting or removing high correlated variables at the same time) or the orthogonality of loadings (guaranteeing the independence of loadings).

5.2.2 Re-Designing w_{ij}

The objective of re-designing w_{ij} is to incorporate the grouping effect and loading orthogonality into the AWSPCA estimates. To achieve this, the w_{ij} are designed to satisfy the following conditions:

- Produce similar penalization for loading elements that have similar values in regular PCA.

- Assign every variable to only one loading thereby achieving orthogonality between loadings.
- Produce a trade-off between loading sparsity and model accuracy.

Mathematically, w_{ij} can be defined as:

$$w_{ij} = \frac{N^{2M_i}}{|p_{ij}|\ R_i}, \tag{5.3}$$

to penalise the i^{th} element in the j^{th} loading, where

$$\begin{aligned}
M_i &= \sum_{k=1}^{j-1} \widehat{p}_{ik}^{\,\text{S}}, && (5.4)\\
N &= \max(m, n), && (5.5)\\
S_F &= \{1, \cdots, n\}, && (5.6)\\
S_k &= \{i | \widehat{p}_{ik}^{\,\text{S}} \neq 0, \text{for } i = 1, \cdots, n\}, && (5.7)\\
S_j &= S_F - \bigcup_{k=1}^{j-1} S_k, && (5.8)\\
S_i &= \{q|\ |corr(\mathbf{x}_i, \mathbf{x}_q)| \geq \tau, q \in S_j\}, && (5.9)\\
R_i &= \text{card}(S_i), && (5.10)
\end{aligned}$$

where $\widehat{\mathbf{p}}_k^{\text{S}}$ $(\in \mathbb{R}^{n\times 1})$, $k = 1, \cdots, j-1$ denotes the k^{th} sparse loading in the existing $j-1$ loadings. M_i measures if the i^{th} variable has appeared in the existing $j-1$ sparse loadings. If so, M_i records the number of appearances; otherwise, $M_i = 0$. Because N is large for a given large data set, when $M_i = 0$, w_{ij} is much smaller than that when $M_i \neq 0$, leading to less constraint to the i^{th} element in the j^{th} loading. Hence, when one coefficient already occurs in the existing loadings, it is very unlikely to be included in the incoming (the j^{th}) loading, forcing the obtained loadings to be orthogonal.

m and n denote the number of observations and variables, respectively. S_F denotes the set of indices for all variables, S_k is the set containing the indices of the variables that do not occur in the k^{th} sparse loading and S_j is the set of indices of the variables that do not occur in the existing $j-1$ loadings (unassigned variables). S_i denotes the set of indices of unassigned variables that are high correlated with the i^{th} variable. card$(\cdot)$ denotes the number of elements in the set. R_i is thus a count of the number of unassigned variables that are highly correlated with the i^{th} variable. If the i^{th} variable

is highly correlated with many of the unassigned variables, R_i is large and hence, w_{ij} is small (Eq. (5.3)), leading to less penalisation to the i^{th} coefficient. Since R_i (≥ 1) is greater than $|p_{ij}|$ (≤ 1), R_i has a much bigger effect on w_{ij} than p_{ij}. As a consequence, variables that are highly correlated with other variables are more likely to be included in the sparse loadings.

5.2.3 Optimization Criterion

Following the design of w_{ij}, the next question is how to calculate AWSPCA. Before answering this question, it is necessary to define an optimization criterion. The existing sPCA-rSVD algorithm [141] provides a good indication. Denoting the first PC score and loading in the regular PCA as $\mathbf{u}$ and $\mathbf{v}$, respectively, the best rank-one approximation of the data matrix $\mathbf{X}$ can be expressed as $\mathbf{u}\mathbf{v}^{\mathrm{T}}$, subject to $||\mathbf{v}||_2 = 1$ [141]. According to sPCA-rSVD, for a given $\mathbf{u}$, $\mathbf{v}$ can be obtained as:

$$\widehat{\mathbf{v}} = \arg\min_{\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^{\mathrm{T}}||_2^2 \ s.t. \ ||\mathbf{v}||_2 = 1 \tag{5.11}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$, $\mathbf{u} \in \mathbb{R}^{m\times 1}$ and $\mathbf{v} \in \mathbb{R}^{n\times 1}$. The penalised regression estimation of $\mathbf{v}$ to achieve sparse solutions can be expressed as:

$$\widehat{\mathbf{v}} = \arg\min_{\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^{\mathrm{T}}||_2^2 + \lambda \sum_{i=1}^{n} ||v_i||_1, \ s.t. \ ||\mathbf{v}||_2 = 1, \tag{5.12}$$

where λ is the tuning parameter of the LASSO penalty, used to control the shrinkage level of the regression coefficients.

As discussed in [141], sPCA-rSVD is valid for solving any type of data matrix, even if the data matrix is ill-conditioned, an important superiority over EN-SPCA, under which the ill-conditioning problem can only be tackled by introducing the ridge penalty into the estimate regression framework. This inspired us to investigate if AWSPCA can be defined in a similar format so that the corresponding solutions can share the same superiority over EN-SPCA. As such, the optimization criterion of AWSPCA can be expressed as

$$\widehat{\mathbf{v}} = \arg\min_{\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^{\mathrm{T}}||_2^2 + 2\lambda \sum_{i=1}^{n} w_i||v_i||_1, \ s.t. \ ||\mathbf{v}||_2 = 1 \tag{5.13}$$

where $\mathbf{v}$ denotes the first sparse loading and $w_i = w_{i1}$, as defined in Eq. 5.3. The reason for setting $j = 1$ is that the calculation is performed for one loading at a time. To obtain additional loadings, the same calculation procedure can be applied repeatedly to the residual data that remains after the contribution of the already computed loadings is removed.

5.2.4 Numerical Solution

In sPCA-rSVD, calculation of the regression coefficients is element-based, so even if the data matrix is ill-conditioned, it won't affect the validity of the algorithm. Inspired by sPCA-rSVD, AWSPCA can be solved by a similar method. To achieve this, the following two Lemmas are introduced, which are extensions of the Lemmas presented for solving sPCA-rSVD in [141].

Lemma 1 For a given $\mathbf{v}$, the solution of $\mathbf{u}$ that minimizes Eq. (5.13) can be expressed as:

$$\widehat{u}_j = \frac{\sum_{i=1}^{n} x_{ji} v_i}{\sum_{i=1}^{n} v_i^2}. \tag{5.14}$$

Proof 1 Define J as a scalar cost function of $\mathbf{u}$ and $\mathbf{v}$, $J = ||\mathbf{X} - \mathbf{u}\mathbf{v}^{\mathrm{T}}||_2^2 + 2\lambda \sum_{i=1}^{n} w_i ||v_i||_1$. This can be expressed as

$$J = \sum_{j=1}^{m} \sum_{i=1}^{n} (x_{ji} - u_j v_i)^2 + 2\lambda \sum_{i=1}^{n} w_i ||v_i||_1, \tag{5.15}$$

$$= \sum_{j} [\sum_{i} x_{ji}^2 - 2 \sum_{i} x_{ji} u_j v_i + \sum_{i} u_j^2 v_i^2] + 2\lambda \sum_{i} w_i ||v_i||_1 \tag{5.16}$$

Minimising J for a given $\mathbf{v}$ (w_{ij} are known) is equivalent to minimising

$$\sum_{i} x_{ji}^2 - 2 \sum_{i} x_{ji} u_j v_i + \sum_{i} u_j^2 v_i^2, \tag{5.17}$$

which is a quadratic function of u_j and the minimum occurs at the vertex where

$$\widehat{u}_j = \frac{2 \sum_{i=1}^{n} x_{ji} v_i}{2 \sum_{i=1}^{n} v_i^2} = \frac{\sum_{i=1}^{n} x_{ji} v_i}{\sum_{i=1}^{n} v_i^2}. \tag{5.18}$$

Lemma 2 For a given $\mathbf{u}$, the solution of $\mathbf{v}$ that minimizes Eq. (5.13) can be expressed as

$$\widehat{v}_i = \frac{\mathrm{sign}(\sum_{j=1}^{m} x_{ji} u_j)[|\sum_{j=1}^{m} x_{ji} u_j| - \lambda w_i]_+}{\sum_{j=1}^{m} u_j^2}. \tag{5.19}$$

Proof 2 As defined in proof 1, the cost function can be expressed as

$$J = \sum_{j=1}^{m}\sum_{i=1}^{n}(x_{ji} - u_j v_i)^2 + 2\lambda\sum_{i=1}^{n} w_i|v_i|, \tag{5.20}$$

$$= \sum_{j}\sum_{i}[x_{ji}^2 - 2x_{ji}u_j v_i + u_j^2 v_i^2] + 2\lambda\sum_{i} w_i|v_i|, \tag{5.21}$$

$$= \sum_{i}[\sum_{j}(x_{ji}^2 - 2x_{ji}u_j v_i + u_j^2 v_i^2) + 2\lambda w_i|v_i|], \tag{5.22}$$

$$= \sum_{i}[\sum_{j} x_{ji}^2 - 2\sum_{j} x_{ji}u_j v_i + \sum_{j} u_i^2 v_i^2 + 2\lambda\text{sign}(v_i)v_i w_i] \tag{5.23}$$

$$= \sum_{i}[\sum_{j} x_{ji}^2 - 2[\sum_{j} x_{ji}u_j - \lambda\text{sign}(v_i)w_i]v_i + \sum_{j} u_j^2 v_i^2], \tag{5.24}$$

Minimising J for a given $\mathbf{u}$ (w_{ij} are known) is equivalent to minimising

$$\sum_{j} x_{ji}^2 - 2[\sum_{j} x_{ji}u_j - \lambda\text{sign}(v_i)w_i]v_i + \sum_{j} u_j^2 v_i^2, \tag{5.25}$$

which is a quadratic function of v_i. It is equivalent in format to minimising

$$a\beta^2 - 2b\beta + 2\lambda c|\beta| + d. \tag{5.26}$$

The minimizer of Eq. (5.26) can be expressed as

$$\widehat{\beta} = \frac{sign(b)(|b| - \lambda c)_+}{a}. \tag{5.27}$$

As such, the minimizer of Eq. (5.25) is given by

$$\widehat{v}_i = \frac{\text{sign}(\sum_{j=1}^{m} x_{ji}u_j)[|\sum_{j=1}^{m} x_{ji}u_j| - \lambda w_i]_+}{\sum_{j=1}^{m} u_j^2}. \tag{5.28}$$

According to Lemma 1 and Lemma 2, the proposed algorithm for solving AWSPCA can be described as follows.

- Step 1: Initialization: Apply the standard SVD to $\mathbf{X}$ and obtain the best rank-one approximation of $\mathbf{X}$ as $\mathbf{u}^*\mathbf{v}^{*\mathsf{T}}$. Set $\mathbf{v} = \mathbf{v}^*$ and $\mathbf{u} = \mathbf{u}^*$.

- Step 2: Update $\widehat{\mathbf{v}}_{new} = [\widehat{v}_1, \cdots, \widehat{v}_i, \cdots, \widehat{v}_n]$ and $\widehat{\mathbf{u}}_{new} = [\widehat{u}_1, \cdots, \widehat{u}_j, \cdots, \widehat{u}_m]$ as

  (a)
  $$\widehat{v}_i = \frac{\text{sign}(\sum_{j=1}^{m} x_{ji}u_j)[|\sum_{j=1}^{m} x_{ji}u_j| - \lambda w_i]_+}{\sum_{j=1}^{m} u_j^2}. \tag{5.29}$$

  (b)
  $$\widehat{u}_j = \frac{2\sum_{i=1}^{n} x_{ji}v_i}{2\sum_{i=1}^{n} v_i^2} = \frac{\sum_{i=1}^{n} x_{ji}v_i}{\sum_{i=1}^{n} v_i^2}. \tag{5.30}$$

- Step 3: Repeat Step 2 with $\mathbf{u} = \widehat{\mathbf{u}}_{new}$ and $\mathbf{v} = \widehat{\mathbf{v}}_{new}$ until convergence.
- Step 4: Standardize $\widehat{\mathbf{v}}_{new}$ as $\widehat{\mathbf{p}}^{\mathrm{S}} = \widehat{\mathbf{v}}_{new}/||\widehat{\mathbf{v}}_{new}||$, where $\widehat{\mathbf{p}}^{\mathrm{S}}$ is the sparse loading in AWSPCA.

The computation cost of each iteration is $O(nm)$.

The iterative produce of the proposed algorithm is used to obtain the first sparse loading vector. Subsequent sparse loadings can be obtained sequentially via rank-one approximation of the residual data. The number of sparse loadings can be specified in advance, but once all variables have been included, the algorithm will be terminated, even if the achieved number of loadings is less than that specified.

Given $v_i = 0$, it can be obtained from Eq. 5.19 that

$$\lambda_i = |\sum_{j=1}^{m} x_{ji}u_j|/w_i. \tag{5.31}$$

Since w_i is nonnegative, λ_i is also nonnegative. If setting all λ_i to zeros, the proposed algorithm is equivalent to solving PCA in the least-squares regression framework. When

$$\lambda_{max} = \max_i(|\sum_{j=1}^{m} x_{ji}u_j|/w_i), \tag{5.32}$$

all loading coefficients will be penalized to zero, while when

$$\lambda_{min} = \min_i(|\sum_{j=1}^{m} x_{ji}u_j|/w_i), \tag{5.33}$$

only one loading element will be penalized to zero. Hence the solutions of AWSPCA can be obtained by setting λ in the range of $[0,\ \lambda_{max}]$. This is superior to EN-SPCA, which has no method for determining an upper bound on λ.

5.2.5 Variance Explained by the Sparse Principal Components

In EN-SPCA, the obtained sparse loadings are not orthogonal, leading to the difficulty in calculating the variance explained by the sparse principal components. To address this issue, Zou *et al.* [196] proposed the matrix orthogonalization for removing the correlation occurring in the PC scores (details have been discussed in section 4.4). In

contrast, Shen and Huang [141] proposed to project the data matrix $\mathbf{X}$ onto the p-dimensional subspace spanned by the k sparse components. The projected data matrix can be expressed as:

$$\widehat{\mathbf{X}} = \mathbf{X}\mathbf{P}^{\mathrm{S}}[(\mathbf{P}^{\mathrm{S}})^{\mathrm{T}}\mathbf{P}^{\mathrm{S}}]^{-1}(\mathbf{P}^{\mathrm{S}})^{\mathrm{T}}, \tag{5.34}$$

where $\mathbf{P}^{\mathrm{S}}$ denotes the matrix with sparse loadings. Then, the variance explained by the first k sparse principal components can be expressed as $\mathrm{tr}(\widehat{\mathbf{X}}^{\mathrm{T}}\widehat{\mathbf{X}})$.

Due to the orthogonality of sparse loadings obtained by AWSPCA, the total variance explained by the first k sparse principal components (PCs) can be expressed as

$$V_e = \mathrm{tr}[(\mathbf{T}^{\mathrm{S}})^{\mathrm{T}}\mathbf{T}^{\mathrm{S}}], \tag{5.35}$$

where

$$\mathbf{T}^{\mathrm{S}} = \mathbf{X}\mathbf{P}^{\mathrm{S}}. \tag{5.36}$$

5.3 Study of AWSPCA Properties on Artificial Data

In this section, the application of AWSPCA to an artificial data set is used to illustrate the properties of AWSPCA. The artificial data set used is the one introduced in Chapter 4 (section 4.5.1) for investigation of the EN-SPCA algorithm. This consists of 10 variables measured over 1000 observations.

5.3.1 Sparsity

One of the key properties of AWSPCA is that it provides sparse solutions to the loadings in regular PCA. The key parameter in AWSPCA is λ, which affects the solution sparsity. As an example, Table 5.1 shows that how the changes in λ affect the resulting sparsity in the first loading. As can be observed, when $\lambda = 0$, there is no constraint added to the regression coefficients, leading to a solution identical to that of regular PCA. When λ is increased to λ_{min} (defined in Eq. 5.33), one loading coefficient is penalised to zero. As λ increasing, $\lambda = 10^5$ for example, more loading coefficients are shrunk to zero. When λ reaches λ_{max} (defined in Eq. 5.32), the constraint level is so high that all loading coefficients are penalised to zero. Hence, λ is an important parameter for determining the solution sparsity.

Order of	**AWSPCA**				**PCA**
Variables	$\lambda = 0$	$\lambda_{min} = 1.7247 \times 10^4$	$\lambda = 10^5$	$\lambda_{max} = 6.639 \times 10^5$	
1	-0.0834	-0.0046	0	0	-0.0834
2	-0.0812	0	0	0	-0.0812
3	-0.0818	-0.0013	0	0	-0.0818
4	-0.0813	-0.0002	0	0	-0.0813
5	0.4103	0.4164	0.4228	0	0.4103
6	0.4093	0.4153	0.4212	0	0.4093
7	0.4112	0.4173	0.4236	0	0.4112
8	0.4103	0.4163	0.4227	0	0.4103
9	0.3877	0.3922	0.3788	0	0.3877
10	0.3866	0.391	0.3771	0	0.3866

Table 5.1: The first loading obtained by AWSPCA as a function of λ and that obtained by PCA.

5.3.2 Grouping Effect

The grouping effect is one important property in EN-SPCA. However, as proved in [196], the grouping effect is only guaranteed when the penalty function is strictly convex. In AWSPCA, the LASSO penalty is not strictly convex, but as an alternative, it is implemented in the design of the adaptive weighting (details have been discussed in section 5.2.2).

The experimental results shown in Table 5.2 demonstrate that when $\tau = 0.9$, variable {1, 2, 3, 4} and variable {5, 6, 7, 8, 9, 10} are separated in the two sparse loadings. Increasing the value of τ to 0.98, *i.e.* strengthening the definition of high correlation, variable {5, 6, 7, 8} and variable {9, 10} are separated into two loadings. Hence, the grouping effect is achievable in AWSPCA.

5.3.3 Orthogonality

Experiments in this section focus on demonstrating the solution orthogonality of AWSPCA, so manual selection of the tuning parameters, λ^i $(i = 1, 2, 3)$ for the first three sparse components is applied. An automated method for tuning parameter selection is dis-

Order of	$\tau = 0.9$		$\tau = 0.98$		
Variables	PC1	PC2	PC1	PC2	PC3
1	0	-0.4991	0	-0.4991	0
2	0	-0.4990	0	-0.4990	0
3	0	-0.5021	0	-0.5022	0
4	0	-0.4998	0	-0.4997	0
5	0.4197	0	0.5007	0	0
6	0.4184	0	0.4967	0	0
7	0.4202	0	0.5022	0	0
8	0.4195	0	0.5004	0	0
9	0.3855	0	0	0	-0.7080
10	0.3843	0	0	0	-0.7062

Table 5.2: Grouping effect contained in AWSPCA.

cussed in detail in Section 5.4.2. Given $\tau = 0.9$, λ^1 and λ^2 are set to 8.623×10^3 and 3.9167×10^4, respectively. Increasing τ to 0.98, three sparse components can be obtained. The corresponding tuning parameters are set to $\lambda^1 = 1.9565 \times 10^5$, $\lambda^2 = 2.538 \times 10^4$ and $\lambda^3 = 1.956 \times 10^4$.

The sparse loadings obtained using AWSPCA are shown in Table 5.3. Because the distribution of non-zero coefficients in the sparse loadings is not overlapped, orthogonality between the sparse loadings is guaranteed. Denoting the sparse loading matrix by $\widehat{\mathbf{P}}^{\mathrm{S}}$, $(\widehat{\mathbf{P}}^{\mathrm{S}})^{\mathrm{T}}\widehat{\mathbf{P}}^{\mathrm{S}}$ returns an identity matrix, confirming that the sparse loadings are orthogonal to each other.

5.3.4 Variance Explained

As shown in Table 5.3, the first sparse PC obtained using AWSPCA ($\tau = 0.9$) captures nearly as much variance as the first regular PC, while using 40% less variables. In addition, the accumulative variance captured by the first two sparse loadings is also close to that captured by the first two regular PCs. One more example is presented in Table 5.3, where τ is set to 0.98. The experimental results show that three loadings can be obtained and the number of non-zero elements is 4, 4 and 2 for the first three sparse

	AWSPCA					**PCA**		
Order of	$\tau = 0.9$		$\tau = 0.98$					
Variables	PC1	PC2	PC1	PC2	PC3	PC1	PC2	PC3
1	0	-0.4991	0	-0.4991	0	0.0834	0.4883	0.5245
2	0	-0.4990	0	-0.4990	0	0.0812	0.4889	-0.1555
3	0	-0.5021	0	-0.5022	0	0.0818	0.4916	-0.4947
4	0	-0.4998	0	-0.4997	0	0.0813	0.4895	0.0531
5	0.4197	0	0.5007	0	0	-0.4103	0.1030	0.4093
6	0.4184	0	0.4967	0	0	-0.4093	0.1012	0.2983
7	0.4202	0	0.5022	0	0	-0.4112	0.0994	-0.2876
8	0.4195	0	0.5004	0	0	-0.4103	0.1020	-0.0942
9	0.3855	0	0	0	-0.7080	-0.3877	-0.0073	-0.0349
10	0.3843	0	0	0	-0.7062	-0.3866	-0.0083	-0.3223
VE (%)	61.40	37.29	43.35	37.29	18.68	62.08	37.15	0.1094
Accumulative VE (%)	61.40	98.69	43.35	80.64	99.33	62.08	99.23	99.34

Table 5.3: The loadings obtained using AWSPCA and PCA.

loadings, respectively. The variance explained by the first sparse PC is substantially less than that for PCA, but the accumulative percentage of variance explained over 3 PCs is 99.33%, compared to 99.34% for regular PCA. Hence, the sparse PCs obtained using AWSPCA are efficient at summarising the information contained in the data.

5.4 AWSPCA Applied to SDS1

5.4.1 PCA, A Special Case of AWSPCA

Theoretically, if the adaptive weighting LASSO penalty is set to zero, AWSPCA should be equivalent to PCA. Provided that the sparse loadings are scaled to unit length, the solutions of PCs should be identical. The experimental results shown in Table 5.4 provides a comparison between PCA and AWSPCA in terms of variance and the number of nonzero elements for the SDS1 data set. As can be seen, the same amount of variance is captured in each PC and the number of nonzero elements in each PC is identical. Plotting the first three PCs (in Fig. 5.1) shows that there is no difference

	Variance Explained (%)		Number of Nonzero Elements	
Order of PCs	PCA	AWSPCA	PCA	AWSPCA
PC1	47.27	47.27	20	20
PC2	39.6	39.6	20	20
PC3	10.89	10.89	20	20

Table 5.4: PCA, a special case of AWSPCA ($\lambda_1 = \lambda_2 = \lambda_3 = 0$.)

between the PCs obtained by AWSPCA and by PCA.

5.4.2 Selecting the Tuning Parameters

As discussed in section 5.2.4, different levels of component sparsity (number of zero variables in the component) can be achieved when setting λ_i in the range of $[\lambda_{min},\ \lambda_{max}]$ using AWPSCA. When $\lambda_i = \lambda_{min}$, only one variable in the component is shrunk to zero. Increasing λ_i to λ_{max}, all variables in the component are shrunk to zero. However, the analysis does not tell which λ_i to choose to obtain the optimal component sparsity. In this section, a method for selecting λ_i is proposed. Using the selected λ_i, variables with distinctive patterns can be separated into different components, while variables with similar patterns can be selected in the same component.

According to Eq. 5.31, λ_i is tied to each variable. Variables with similar patterns yield similar λ_i. As such, the difference between λ_i can be used as an indicator of the difference between the corresponding variables. Arranging λ_i in increasing order, the most significant difference between the adjacent λ_i indicates that the corresponding variables are the most distinctive. Denoting λ^a ($\lambda^a = [\lambda_{min}, \cdots, \lambda^a_{j-1}, \lambda^a_j, \cdots, \lambda_{max}]$) as the arranged sequence of λ_i, then the possible solution of λ, λ^*, can be expressed as

$$\lambda^* = \lambda^a_J, \tag{5.37}$$

where

$$J = \arg\max_j(\lambda^a_j - \lambda^a_{j-1}). \tag{5.38}$$

However, as the criterion for selecting λ, the condition defined in Eq. 5.37 is not complete. The reason is that it cannot guarantee the variables selected in the same

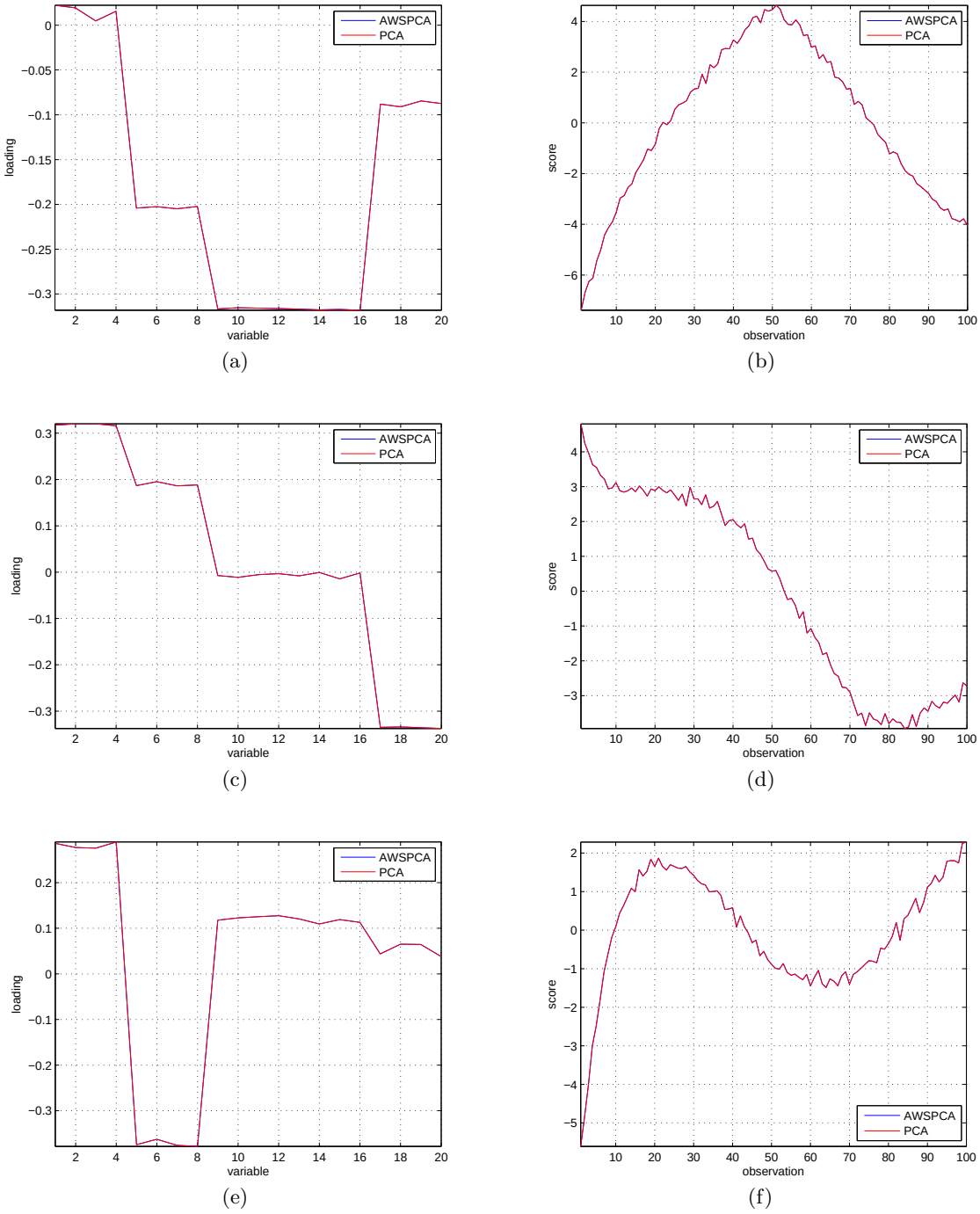


Figure 5.1: The first 3 loadings of PCA and AWSPCA components ($\lambda_1 = \lambda_2 = \lambda_3 = 0$): (a) The first loading; (b) The first score; (c) The second loading; (d) The second score; (e) The third loading; (f) The third score.

component have sufficient similarity. A statistical method, Pearson's correlation coefficient, is employed here as a measure of the similarity/correlation between variables. Denoting $\mathbf{p}^S$ as the obtained sparse component, $\mathbf{p}^S = [p_1^S, \cdots, p_i^S, \cdots, p_n^S]$ (n: data dimensions) and I as the set of non-zero variables, $I = [i|p_i^S \neq 0, i = 1, \cdots, n]$, the average correlation between these non-zero variables can be defined as

$$\rho = ave_{i,j\in I,\ i\neq j}[corr(\mathbf{x}_i, \mathbf{x}_j)], \tag{5.39}$$

where $ave(\cdot)$ and $corr(\cdot)$ denote the function of average and Pearson's correlation coefficient, respectively. Supposing ξ is the similarity threshold, the variables included are considered as highly correlated, as long as

$$\rho \geq \xi. \tag{5.40}$$

The proposed method of selecting λ is a combination of the above discussed criteria. The condition defined in Eq. 5.37 is considered first. If the result cannot guarantee that the average correlation between these non-zero variables is higher than the threshold (ξ), then the average correlation criterion is employed as the only criterion for selecting λ. The complete method of selecting λ is given as follows.

Step 1: Checking the effectiveness of λ_J^a. Given $\lambda^* = \lambda_J^a$, if the condition of $\rho \geq \xi$ can be satisfied, then

$$\lambda = \lambda_J^a; \tag{5.41}$$

otherwise, go to Step 2.

Step 2: Selecting λ according to the similarity level. For each λ_j^a, calculate the average correlation, ρ_j. Select the smallest λ_j^a that can satisfy $\rho_j \geq \xi$ as the final solution of λ, *i.e.*

$$\lambda = \min_j(\lambda_j^a|\rho_j \geq \xi). \tag{5.42}$$

To illustrate the effectiveness of the proposed method, SDS1 is used. Experimental results showing the selection of λ for the first four sparse components are shown in Fig. 5.2 with ξ set to 0.9. As can be observed, the big change in the λ_j^a curve (Fig. 5.2 (a)) occurs at $j = 13$ or $\lambda_j^a = 744.36$, which can be more clearly observed in the curve of $\lambda_j^a - \lambda_{j-1}^a$ (Fig. 5.2 (b)). The change in ρ_j as a function of λ_j^a is shown in Fig. 5.2

(c), which shows that when $\lambda = 744.36$, the average correlation between the non-zero variables in the sparse component is above 0.9. Hence, λ selected for the first sparse component is 744.36.

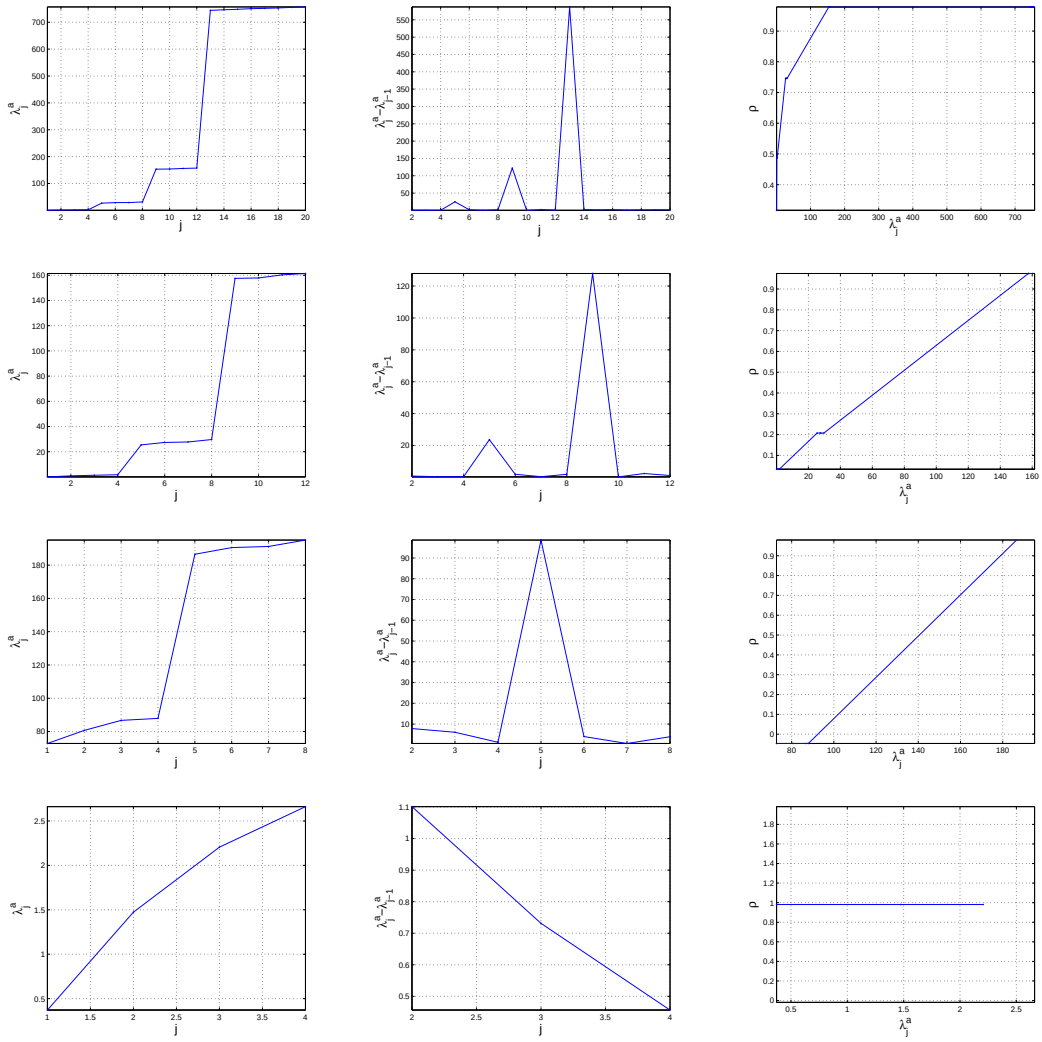


Figure 5.2: Selecting λ for the first four sparse components: (a) λ_j^a for the first component; (b) $\lambda_j^a - \lambda_{j-1}^a$ for the first component; (c) ρ_j as a function of λ_j^a for the first component; (d) λ_j^a for the second component; (e) $\lambda_j^a - \lambda_{j-1}^a$ for the second component; (f) ρ_j as a function of λ_j^a for the second component; (g) λ_j^a for the third component; (h) $\lambda_j^a - \lambda_{j-1}^a$ for the third component; (i) ρ_j as a function of λ_j^a for the third component; (j) λ_j^a for the fourth component; (k) $\lambda_j^a - \lambda_{j-1}^a$ for the fourth component; (l) ρ_j as a function of λ_j^a for the fourth component.

Note that according to Fig. 5.2 (c), for any value of λ_j^a greater than 153, ρ_j is above the threshold set at 0.9. However, λ_j^a is given the priority for selecting λ. The reason is that λ_i (defined in Eq. 5.31) is a direct measure of the variable features and is thereby more reliable than the statistical average, ρ_j. Using a similar approach to the first component, the λ selected for the second, third and fourth components are 157.58, 186.57 and 0.37, respectively. Note that in the calculation of the fourth component, all the residual variables are highly correlated (very similar), resulting in no significant difference in $\lambda_j^a - \lambda_{j-1}^a$ and ρ, as shown in Fig. 5.2 (k) and (i), respectively.

Table 5.5 shows the sparse loadings obtained when AWSPCA is applied to SDS1 using the selected tuning parameters. As one can see, the distribution of the non-zero elements in each loading does not overlap, so the sparse loadings are orthogonal. For comparison, the loadings obtained using regular PCA are also listed in Table 5.5. Although fewer variables are used in each sparse loading, the percentage of accumated variance explained by the four sparse components is comparable to that obtained with regular PCA. The patterns of the scores for the first four sparse components and regular PCs are shown in Fig. 5.3. It can be observed that the scores of sparse components are representative of the patterns contained in SDS1, while the scores of regular components don't have such a feature.

Order of	**AWSPCA**				**PCA**			
Variables	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4
1	0	0	0	-0.5128	0.0223	0.3173	0.286	0.2369
2	0	0	0	-0.5106	0.0193	0.3202	0.277	0.1896
3	0	0	0	-0.4718	0.0047	0.3204	0.2757	0.2821
4	0	0	0	-0.5037	0.0155	0.3161	0.2892	0.2715
5	0	-0.5015	0	0	-0.2041	0.1868	-0.3744	0.1854
6	0	-0.4942	0	0	-0.2026	0.1954	-0.3625	0.1222
7	0	-0.5077	0	0	-0.2049	0.1863	-0.3759	0.2005
8	0	-0.4964	0	0	-0.2023	0.1884	-0.3782	0.1743
9	-0.3445	0	0	0	-0.3166	-0.0074	0.1176	-0.181
10	-0.2412	0	0	0	-0.3153	-0.0112	0.1227	-0.2383
11	-0.2985	0	0	0	-0.3158	-0.0055	0.1253	-0.2292
12	-0.3301	0	0	0	-0.3161	-0.0033	0.1275	-0.1843
13	-0.3616	0	0	0	-0.3169	-0.008	0.1206	0.034
14	-0.3800	0	0	0	-0.3176	-0.0006	0.1094	0.046
15	-0.3943	0	0	0	-0.3172	-0.0144	0.119	0.0106
16	-0.4409	0	0	0	-0.3180	-0.0018	0.1129	-0.0222
17	0	0	0.4955	0	-0.0881	-0.3350	0.0438	0.3410
18	0	0	0.5139	0	-0.0911	-0.3341	0.0650	0.3534
19	0	0	0.4883	0	-0.085	-0.3358	0.06443	0.3804
20	0	0	0.5020	0	-0.0875	-0.3374	0.0378	0.2587
VE (%)	38.21	19.62	19.68	19.7	47.29	39.6	10.89	0.59
AVE (%)	38.21	57.84	77.52	97.22	47.29	86.87	97.76	98.34

Table 5.5: The first four loadings obtained using AWSPCA and PCA on SDS1 (VE=Variance Explained; AVE=Accumulated Variance Explained), the λ selected for the first four sparse components are 744.36, 157.58, 186.57 and 0.37.

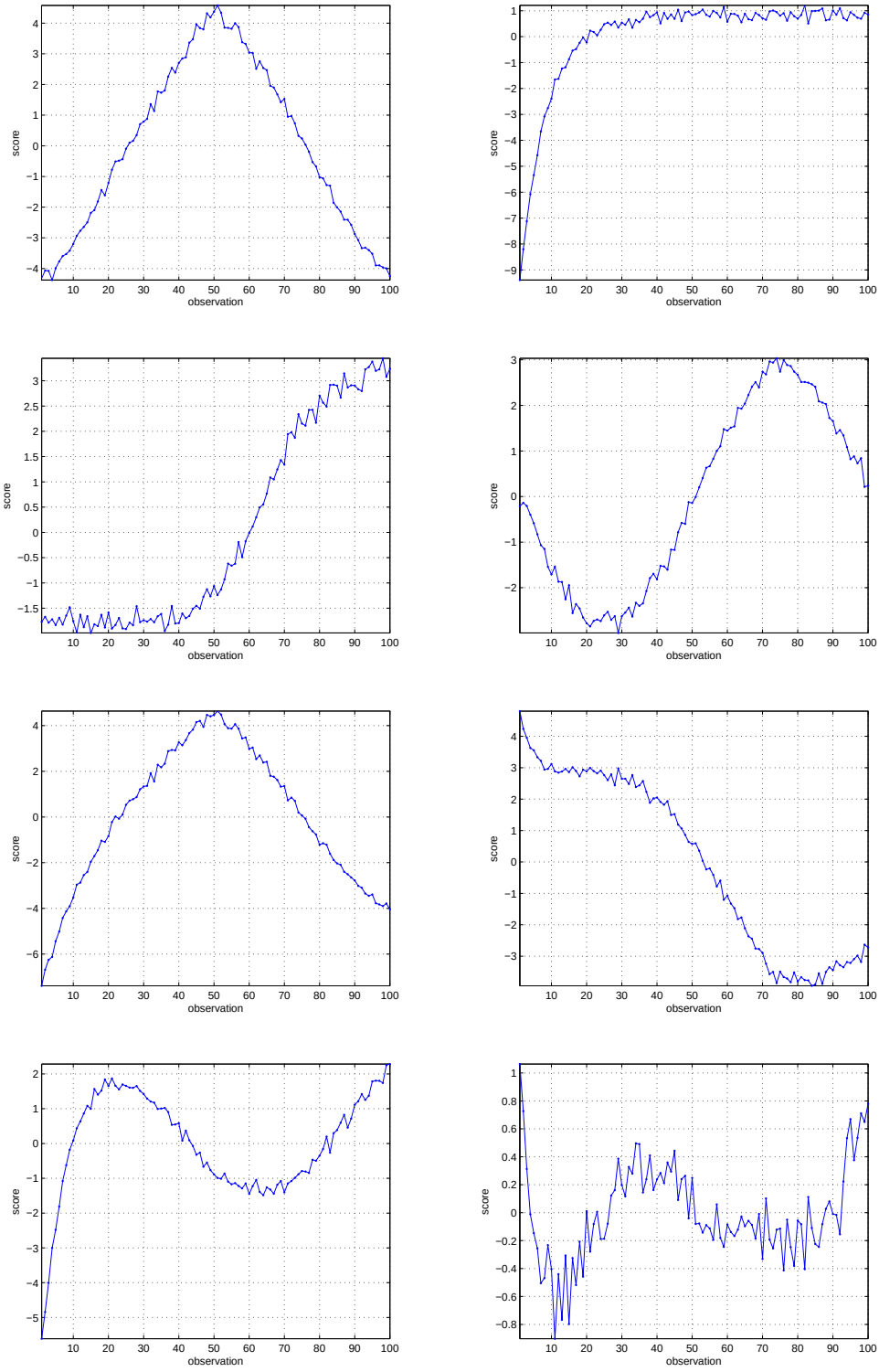


Figure 5.3: The patterns of the scores of the first four sparse components and regular PCs: (a) The score of the first sparse component; (b) The score of the second sparse component; (c) The score of the third sparse component; (d) The score of the fourth sparse component; (e) The score of the first regular PC; (f) The score of the second regular PC; (g) The score of the third regular PC; (h) The score of the fourth regular PC.

5.5 AWSPCA Applied to OES Data

In this section, the application of AWSPCA to OES data is investigated using the benchmark data sets IDS1 and IDS1Filt. As a pre-processing step, each OES channel is mean centered for IDS1 and normalised for IDS1Filt. The reason for not scaling the channels in IDS1 is to avoid amplifying the noise signals.

The method discussed in Section 5.4.2 is employed here for selecting the tuning parameters for IDS1 and IDS1Filt. As an example the tuning parameter selection for the first AWSPCA component on IDS1Filt is shown in Fig. 5.4. The maximum of $\lambda_j^a - \lambda_{j-1}^a$ occurs at $\lambda = 2.8226 \times 10^4$ and the corresponding ρ equals to 0.972, indicating the high level of similarity between the non-zero variables in the component. As such, the selected λ for the first sparse component is 2.8226×10^4 and the number of non-zero variables contained in the first component is 1064. Using this method, the other 7 components are obtained. The number of non-zero variables included in the 8 components is accumulatively, equal to the total number of variables in IDS1Filt.

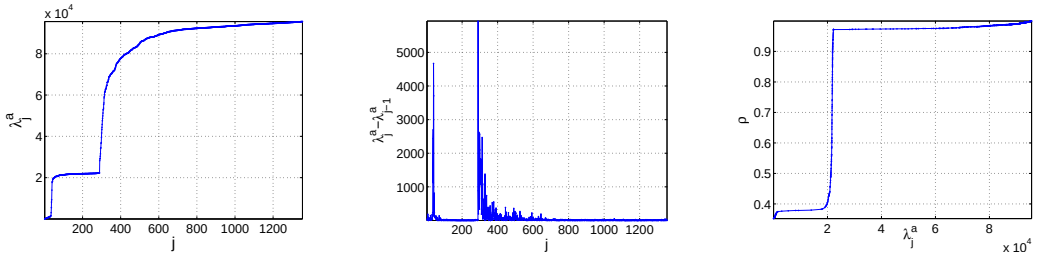


Figure 5.4: Selecting the tuning parameter for the first sparse component: (a) λ_j^a; (b) $\lambda_j^a - \lambda_{j-1}^a$; (c) ρ_j as a function of λ_j^a.

The tuning parameters selected for the first 8 sparse components are shown in Table 5.6, where the number of non-zero variables contained in each loading and the variance explained are also included. As can be seen, most of the variance (75.78%) is captured in the first sparse component and includes 78.75% of the variables (1064 over 1354). The second component capture 18.16% of the variance and uses 250 variables. Hence, the first two component together capture 94.93% of the variance, which is comparable to that explained by the first PC in regular PCA. Relative to the first two components,

the residual 6 components are less significant in terms of variance explained.

Similarly, the tuning parameters selected for IDS1 are shown in Table 5.6. It can be seen that the number of non-zero variables contained in each component and the corresponding variance explained are small. The reason is that because the intensity amplitude is not scaled, the weighting of the variable with high intensity is much larger than the counterpart with small intensity. As a result, the variables with high amplitude are selected. The tuning parameter selection is in effect signal amplitude based for IDS1. For illustration, the intensity changes of the non-zero variables selected in each component are shown in Fig. 5.5. As can be seen, the variables included in the 1^{st}, 2^{nd}, 5^{th}, 6^{th} and 8^{th} have similar patterns, but different amplitudes. A similar pattern occurs with the 3^{rd}, 4^{th} and 7^{th} components.

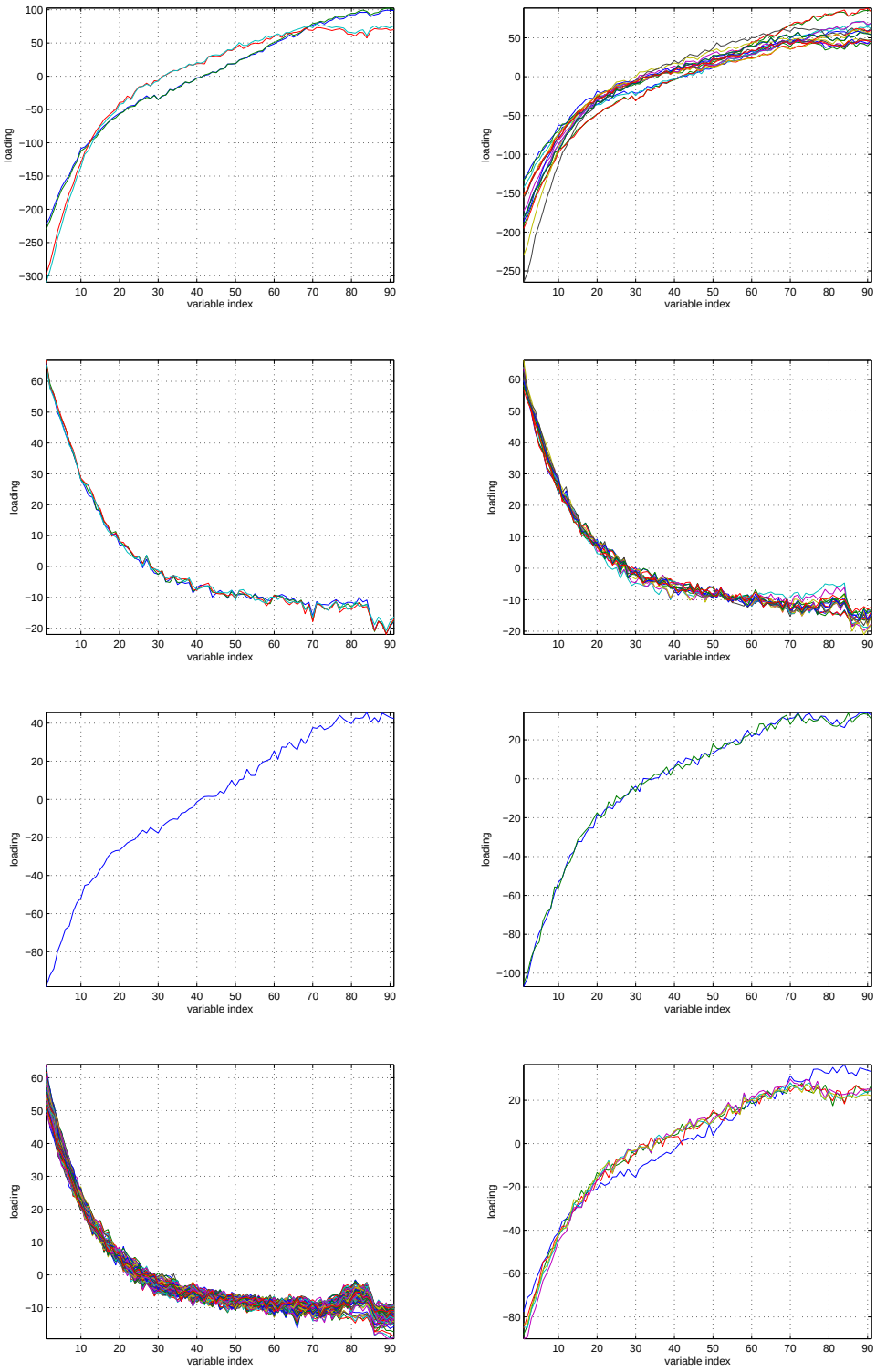


Figure 5.5: Intensity (mean-centered) changes of the nonzero variables in each component for IDS1: (a) First Component; (b) Second Component; (c) Third Component; (d) Fourth Component; (e) Fifth Component; (f) Sixth Component; (g) Seventh Component; (h) Eighth Component.

PC	**AWSPCA**			**PCA**	
i	λ_i	**Non-zero elements in each PC**	**VE** (%)	**Non-zero elements in each PC**	**VE** (%)
IDS1Filt					
1	2.8226×10^4	1064	75.78	2045	94.93
2	1.5827×10^4	250	18.16	2045	4.13
3	6.95×10^2	24	1.56	2045	0.65
4	1.48×10^2	6	0.43	2045	0.1
5	1.51×10^2	3	0.2	2045	0.06
6	44	3	0.22	2045	0.05
7	21	1	0.07	2045	0.03
8	21	2	0.15	2045	0.02
IDS1					
1	1.42×10^8	4	10.6	2045	95.817
2	3.75×10^7	17	23.8	2045	3.007
3	3.55×10^7	4	0.386	2045	0.42
4	2.71×10^7	17	1.81	2045	0.043
5	2.72×10^7	1	0.6	2045	0.038
6	2.64×10^7	2	0.96	2045	0.031
7	1.93×10^7	175	15.7	2045	0.028
8	1.56×10^7	6	0.77	2045	0.027

Table 5.6: Tuning parameters selected for the first 8 sparse components of IDS1 and IDS1Filt and the corresponding number of non-zero variables and variance explained by each component. PCA results are also included for comparison.

The loadings and scores of the eight sparse components obtained for IDS1Filt are shown in Fig. 5.6. The scores of the first two components seem similar. However, because the signs of the non-zero variables in their corresponding loadings are reverse, the patterns shown in those two scores are in fact reverse, indicating two different patterns. The plots of the intensity changes of the non-zero variables in each component are shown in Fig. 5.7. As can be observed, the variables with different patterns are separated in different components.

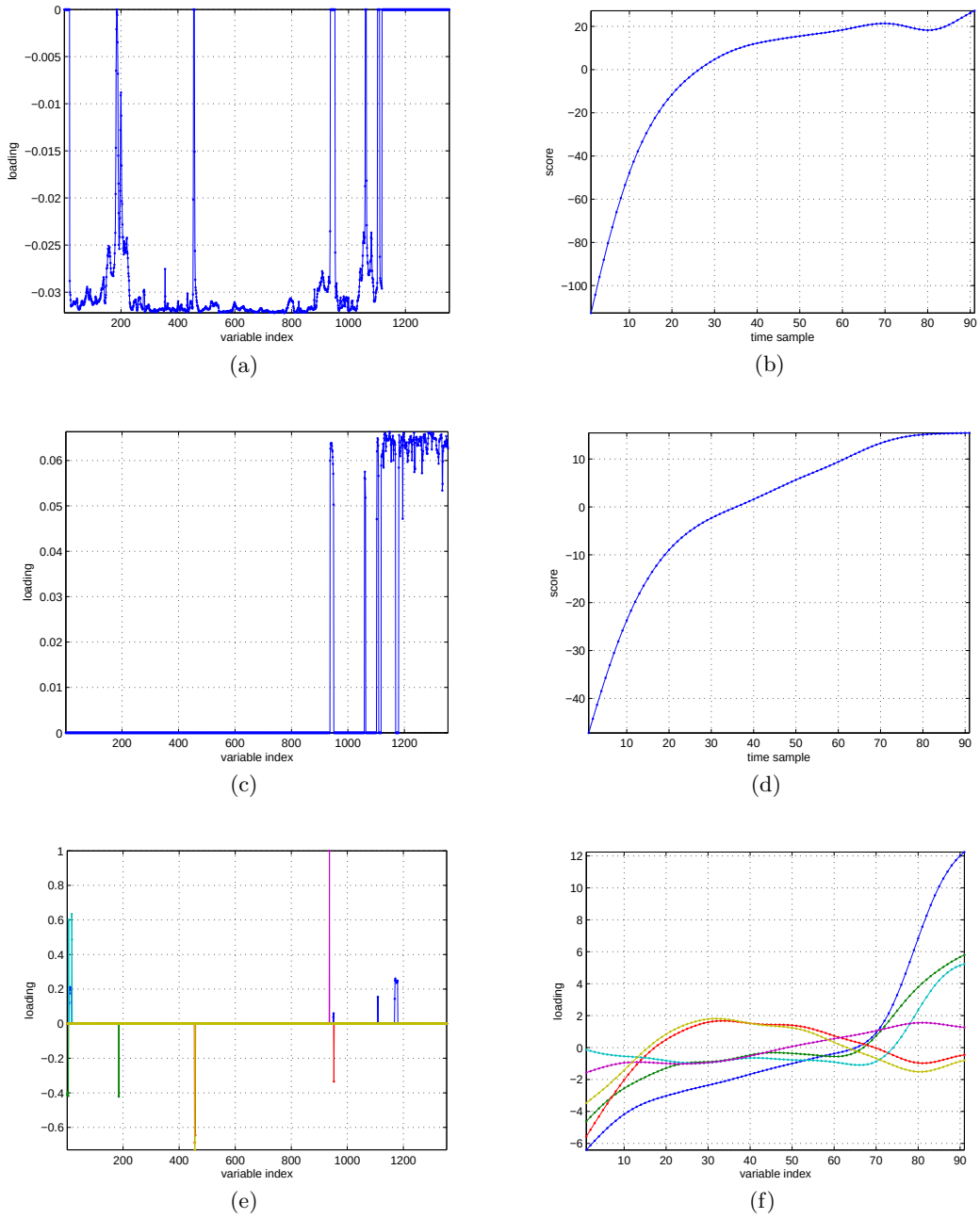


Figure 5.6: The loadings and scores of the sparse components obtained using AWSPCA on IDS1Filt: (a) Loading of the first component; (b) Score of the first component; (c) Loading of the second component; (d) Score of the second component; (e) Loadings of the residual components; (f) Scores of the residual components.

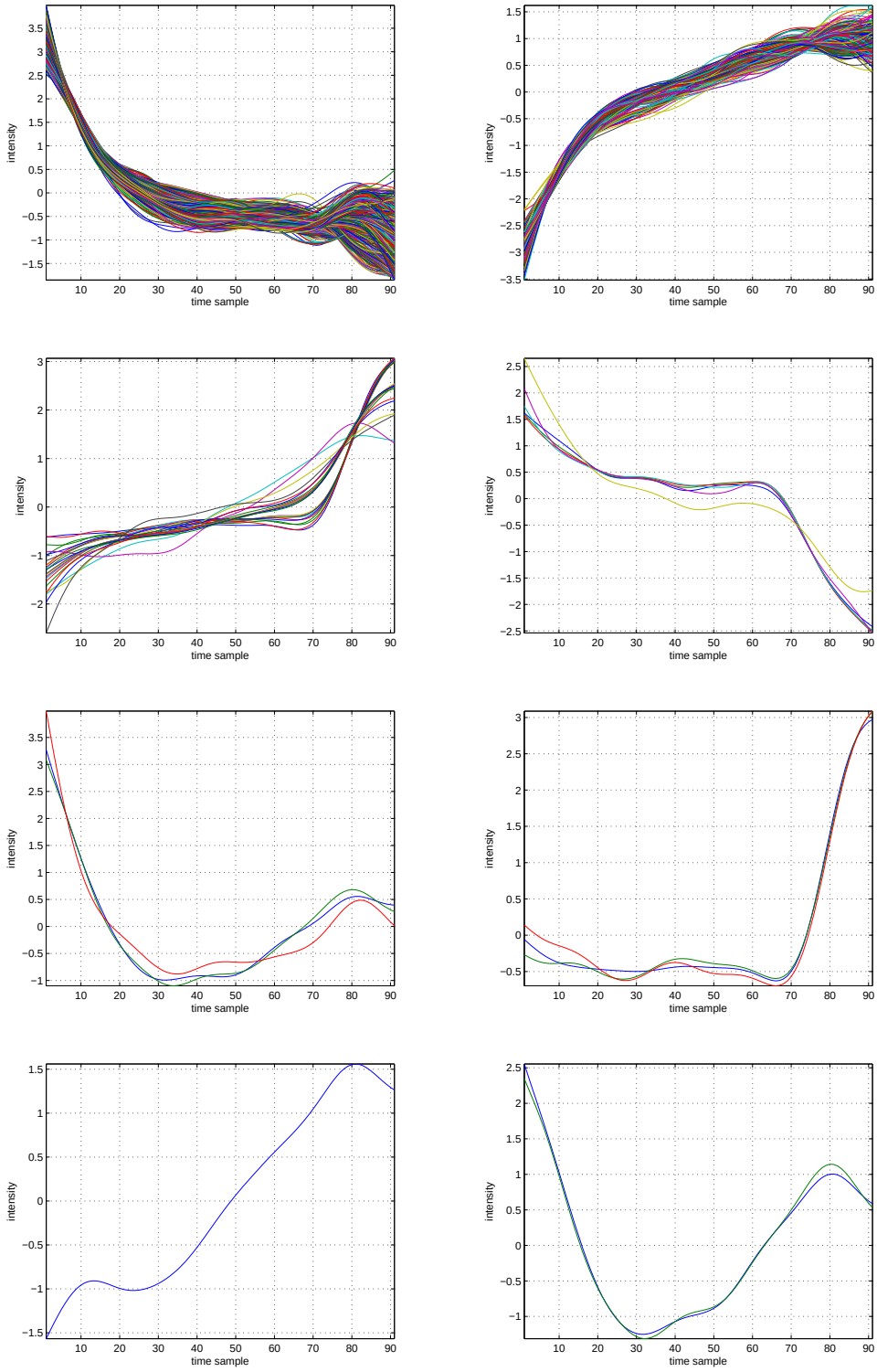


Figure 5.7: Intensity (normalised) changes of the nonzero variables in each of eight sparse components: (a) First Component; (b) Second Component; (c) Third Component; (d) Fourth Component; (e) Fifth Component; (f) Sixth Component; (g) Seventh Component; (h) Eighth Component.

5.6 Discussion and Conclusions

A new adaptive weighting SPCA (AWSPCA) algorithm has been proposed in this chapter as an improvement of the recently proposed EN-SPCA and the adaptive LASSO algorithm. The AWSPCA provides a solution that gives more control over the sparsity and distribution of channels in PCs, while enjoying good prediction accuracy and retention of the grouping effect. In addition, the AWSPCA encourages loading orthogonality, the key attribute possessed by regular PCA. A new numerical solution has been proposed for calculating AWSPCA, which is superior to LARS-EN for EN-SPCA in dealing with ill-conditioning data set.

With the aid of the artificial and simulated data sets, the properties of AWSPCA, *e.g.* sparsity, grouping effect, solution orthogonality and variance explained, have been completely investigated, provides empirical evidence of the properties of AWSPCA. In addition, the effectiveness of the proposed tuning parameter selection method has been explored and estimated.

The application of AWSPCA to OES has shown the potential of the algorithm as a variable selection method, AWSPCA is effective in selecting a subset of variables with similar patterns, while maintaining the grouping effect. Used in conjunction with the proposed tuning parameter selection method, variables with different profiles can be separated into different components, indicating that AWSPCA can be used for variable selection based on pattern differences. This is a significant improvement over the EN-SPCA algorithm.

Chapter 6

Non-Hierarchical Clustering

6.1 Introduction

The area of cluster analysis originated outside the mainstream of statistics, in fields such as psychology and numerical taxonomy (taxonomy refers to the theory and practice of classifying organisms in biology) [47]. In recent decades, cluster analysis has received considerable attention in the area of statistics, although a number of names have been employed depending on the area of application *e.g.* numerical taxonomy in biology, Q-analysis in psychology, unsupervised pattern recognition in the artificial intelligence field and segmentation in market research [37].

Nowadays, cluster analysis is used as a generic term referring to all these kinds of numerical methods used in multivariate data analysis for classifying objects/variables into the groups that have similar patterns, while retaining the distinctive patterns in different groups. The process of clustering is unsupervised [111, 71], *i.e.* one can classify the objects according to the rules made for a particular problem. This is distinct from the approaches known as discriminant analysis and decision analysis that aim at extracting features from the objects that are known to belong to certain groups. As such, one must be aware that using cluster analysis, different rules can produce different clustering results even for the same data set and that in the absence of other 'supervised' information the results are equally valid.

The approaches to searching for clusters can be divided into the so-called hierarchical

and non-hierarchical methods. Hierarchical methods seek to organize all the objects in a structured hierarchical tree, while non-hierarchical methods, such as well known K-means clustering and self organizing maps, seek to separate the objects into distinctive clusters.

A detailed discussion of hierarchical clustering approaches will be provided in Chapter 7. In this chapter, the focus is on non-hierarchical clustering approaches. Typical non-hierarchical clustering approaches can be divided into K-means (and its derivatives such as fuzzy c-means), quality threshold clustering, self organizing maps (SOM), graph-theoretical approaches, model-based clustering and density-based clustering.

Graph-theoretical approaches refer to a series of approaches that convert the problem of separating objects into distinctive clusters into such graph theoretical problems as finding the maximal separation of the objects, according to the so-called proximity graph [76]. In the proximity graph, each object is assigned to a vertex and every pair of objects are connected by so-called edges. For some clustering methods, edges are defined as the proximity values between two objects [140, 180]. For other clustering methods, proximity between two objects is mapped only to either 0 or 1, according to a specified threshold, and edges only exist where the proximity equals to 1 [5, 54]. So far, graph-theoretical approaches have seen wide use in gene data analysis [189].

Model-based clustering seeks to recover the original models from the data and classify each object into a cluster whose objects have the same probability distribution. A key feature of model-based clustering is that it provides a calculation of the probability of an object belonging to a given model [76, 40].

Density-based clustering includes a few recently developed density-based approaches, such as DBSCAN (density based spatial clustering of applications with noise) [36], OPTICS (ordering points to identify the clustering Structure) [108] and CLIQUE (clustering in quest) [1]. In this context, the density is defined as the number of objects contained in a given clustering area. Among these methods, DBSCAN has a wider use. Two main features of DBSCAN are that it is effective at discovering clusters of

arbitrary shape and is able to discriminate noise in the data.

The K-Means clustering, self organizing maps and quality threshold clustering are the techniques investigated in this chapter and are discussed in detail in the following sections, as a precursor to the main contribution of the chapter, a new clustering algorithm for OES data analysis, referred to as max separation clustering. The remainder of the chapter is structured as follows. First, experimental results for the application of K-Means, self organizing maps and quality threshold clustering approaches to simulated data and OES benchmark data sets, are provided, followed by a discussion of the motivation for developing a new max separation clustering (MSC) algorithm. The proposed MSC is described in detail and evaluated for clustering on the simulated data and OES data. Advantages and disadvantages of MSC are provided as a summary at the end.

6.2 K-Means Algorithm

6.2.1 Algorithm Description

One of the most common and widely applied, partition-based clustering algorithms is the so-called K-Means algorithm [120]. Using K-means, n objects can be clustered into K non-overlapping clusters. The algorithmic steps for the classical K-means algorithm can be summarized as follows:

Step 1: Set the number of clusters, K.

Step 2: Initialize K centroids. Randomly select data points as cluster centroids (one point for each cluster).

Step 3: Assign objects to the clusters according to the so-called minimum distance rule, which assigns the objects to their nearest centroids.

Step 4: Recalculate the K centroids, as the statistical mean values of the objects in the corresponding clusters, defined in Eq. (6.4).

Step 5: Repeat from Step 3, until the K centroids do not change any more.

Given a set of objects $\mathbf{X}$ ($\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\},\ \mathbf{x}_i \in \mathbb{R}^{m\times 1}$), K-Means can classify the n objects into K groups or clusters ($K \leq n$) while minimizing the total intra-cluster sum squared error, $TIC(\mathsf{G}, \mathsf{C})$, which is defined in [120] as:

$$TIC(\mathsf{G}, \mathsf{C}) = \sum_{k=1}^{K} IC(G_k, \mathbf{c}_k). \tag{6.1}$$

where G is the set of clusters ($\mathsf{G} = \{G_1, \ldots, G_K\}$), C is the set of the centroids ($\mathsf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$) and $IC(G_k, \mathbf{c}_k)$ is the intra-cluster sum squared error for the k^{th} cluster, G_k:

$$IC(G_k, \mathbf{c}_k) = \sum_{\mathbf{x}_i \in G_k} dist(\mathbf{x}_i, \mathbf{c}_k) \tag{6.2}$$

where $\mathbf{c}_k$ denotes the centroid of G_k and $dist(\cdot)$ is a distance function. The reason for using sum squared error instead of mean squared error is to account for the effect of the size of each cluster. Otherwise, the effect of the outliers on the clustering performance will be exaggerated, leading to an unfaithful measure of the resulting clustering.

As discussed in [88] from the various possible choices of $dist(\cdot)$ and $\mathbf{c}_k$, a popular definition of $dist(\mathbf{x}_i, \mathbf{c}_k)$ is selected:

$$dist(\mathbf{x}_i, \mathbf{c}_k) = \sum_{j=1}^{m} (x_{ij} - c_{kj})^2, \tag{6.3}$$

with $\mathbf{c}_k$ defined as the mean value of the objects contained in cluster k, *i.e.*

$$\mathbf{c}_k = \frac{1}{N_k} \sum_{\mathbf{x}_i \in G_k} \mathbf{x}_i, \tag{6.4}$$

with $N_k = card(G_k)$, where $card(\cdot)$ is the cardinality of G_k.

6.2.2 Choosing the Number of Clusters

A series of key issues associated with K-Means algorithms have not been fully addressed. These issues are choosing the pre-specified number of clusters (K) and the initialization of the centroids, which is known to cause inconsistency in the clustering results. A simple example is used here to highlight these two issues. A two-dimensional artificial data set with 80 objects is presented in Fig. 6.1, where, as can be observed visually, the intrinsic number of clusters is designed to be 4.

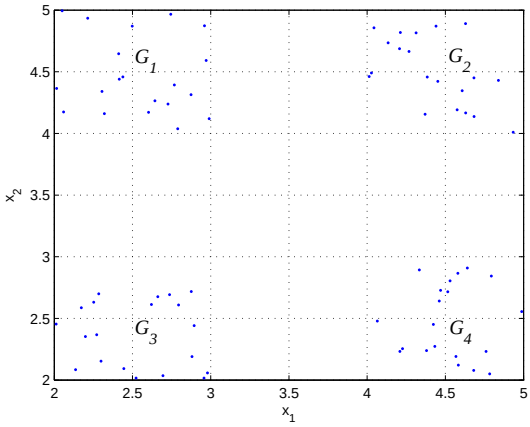


Figure 6.1: An artificial data set with 80 objects, designed to have 4 clusters.

Using classical K-means to cluster the data, as shown in Fig. 6.2, different initializations of the centroids leads to different clustering results. The number of clusters, K, is arbitrarily set to three, an attempt to imitate the case where the intrinsic number of cluster contained in the data is not known. As one can see, the objects contained in G_2, as shown in Fig. 6.2 (a) in the first clustering run are clustered into two groups, G_1 and G_3, as shown in Fig. 6.2 (b) in the second clustering run. Given $K = 4$, the objects contained in G_2 in the first clustering run are clustered into G_2 and G_4 in the second clustering run, while the objects contained in G_3 and G_4 in the first clustering run are clustered into G_3 in the second clustering run, according to Fig. 6.2 (c) and (d). Similarly, given $K = 5$, the clustering results also vary between different clustering runs, as shown in Fig. 6.2 (e) and (f), respectively. Thus, the outcome of clustering is determined by the initial centroid selection, no matter what value is chosen for K, be it less than, equal to or greater than the intrinsic number of clusters.

Due to the variations introduced by initialization, the value of $TIC(\mathsf{G}, \mathsf{C})$ is not unique for a given K. One solution is to run K-means repeatedly (100 times in our experiment) and to select the minimum of $TIC(\mathsf{G}, \mathsf{C})$ as the clustering performance for a given K. Fig. 6.3 shows the resulting values of $TIC(\mathsf{G}, \mathsf{C})$ as a function of K. According to our *prior* knowledge of the data, $K = 4$ should be the solution and this corresponds to the 'elbow point' shown in Fig. 6.3. The so-called 'elbow point' in fact

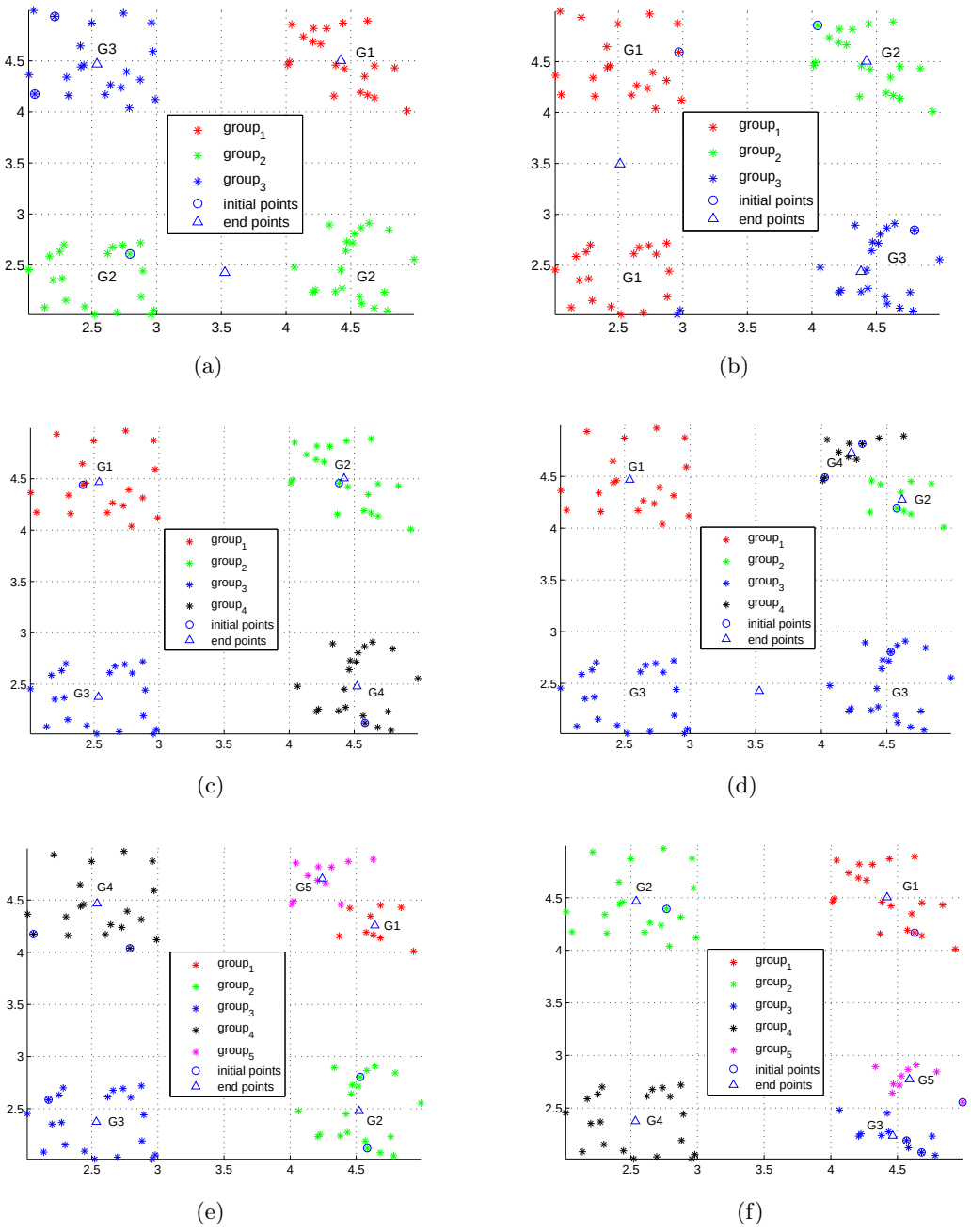
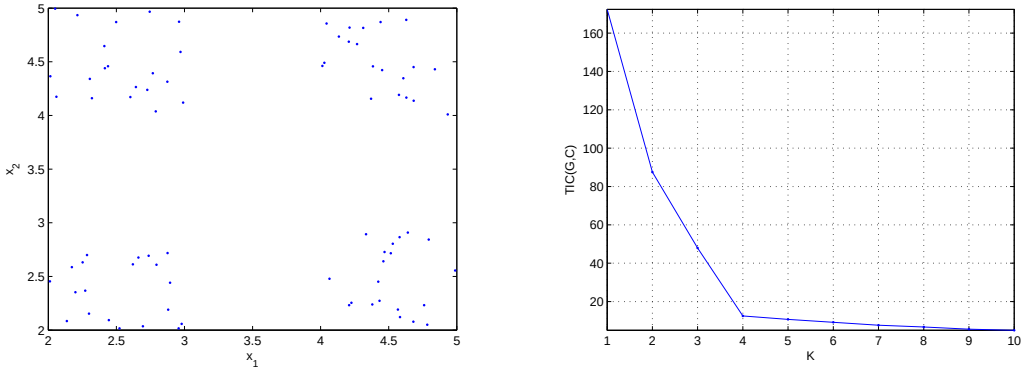
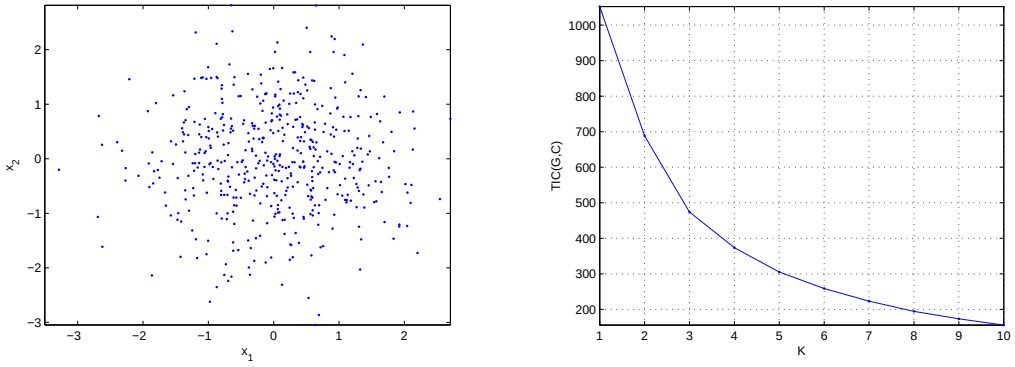


Figure 6.2: Clustering results using classical K-means on artificial data: (a) First clustering run, $K = 3$; (b) Second clustering run, $K = 3$; (c) First clustering run, $K = 4$; (d) Second clustering run, $K = 4$; (e) First clustering run, $K = 5$; (f) Second clustering run, $K = 5$.



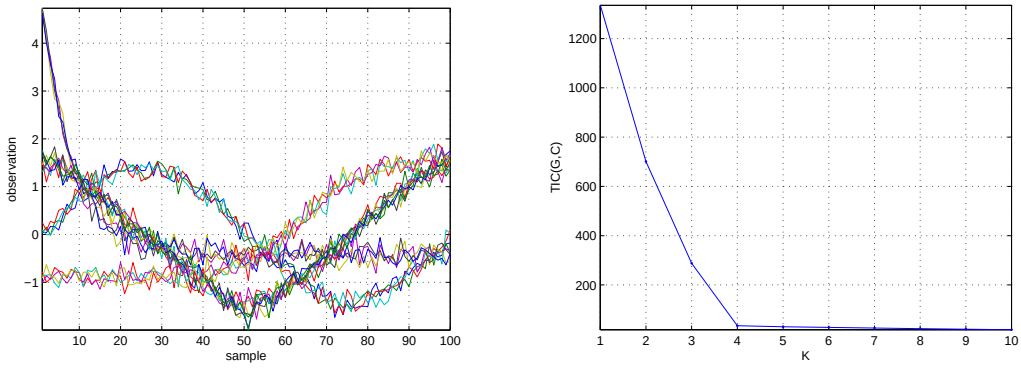
(a) (b)

Figure 6.3: $TIC(\mathsf{G},\mathsf{C})$ changes as a function of K for classical K-means clustering on the artificial data. (a) Artificial data (b) Changes in $TIC(\mathsf{G},\mathsf{C})$ as a function of K



(a) (b)

Figure 6.4: $TIC(\mathsf{G},\mathsf{C})$ changes as a function of K for classical K-means clustering on random data. (a) Random data (b) Changes in $TIC(\mathsf{G},\mathsf{C})$ as a function of K



(a) (b)

Figure 6.5: $TIC(\mathsf{G},\mathsf{C})$ changes as a function of K for classical K-means clustering on SDS1. (a) SDS1 data (standardized) (b) Changes in $TIC(\mathsf{G},\mathsf{C})$ as a function of K

has its theoretical origin in factor analysis, where it is termed as the scree test criterion, proposed by Cattell [15]. A scree test is a visual method that is used to look for the disjunctions in the patterns of eigenvalues to determine the number of dominant factors.

To illustrate the validity of using the 'elbow point', we can use a special case when there are no intrinsic clusters in the data, *e.g.* random data. If the 'elbow point' method is effective, a clear 'elbow point' should not exist in the curve of $TIC(\mathsf{G}, \mathsf{C})$ for this data. This is confirmed in Fig. 6.4, where the random data is displayed in Fig. 6.4 (a) and the changes in $TIC(\mathsf{G}, \mathsf{C})$ as a function of K is shown in Fig. 6.4 (b). It can be seen that there is no such sharp change in the $TIC(\mathsf{G}, \mathsf{C})$ curve in Fig. 6.4 (b). As a final example, the performance of the 'elbow point' method is illustrated for SDS1 in Fig. 6.5. The data is standardised to zero mean and unit variance for K-means to focus on shape difference rather than the amplitude. As can be seen, it correctly predicts 4 as the number of clusters. Setting $K = 4$, the obtained clusters are displayed in Fig. 6.6 and Fig. 6.7. Obviously, different clustering results are obtained for different clustering runs.

6.2.3 Application of K-Means to OES Data

The experimental results of applying the 'elbow point' method to IDS1 and IDS1Filt are shown in Fig. 6.8 (a) and (b), respectively. The estimated number of clusters are 3 for IDS1 and 2 for IDS1Filt. Correspondingly, the obtained clusters represented by the centroids are displayed in Fig. 6.9 and Fig. 6.10, respectively. Standard deviation is employed to measure the variation in the objects contained in each cluster. As can be seen in Fig. 6.10, the variation increases towards the end of each etch run (between sample 70 to 90) indicating some inconsistency in the patterns of the objects contained in the cluster. Based on the patterns extracted by MSC, discussed later in the chapter, it can be concluded that classical K-means clustering will generally only extract the most dominant patterns.

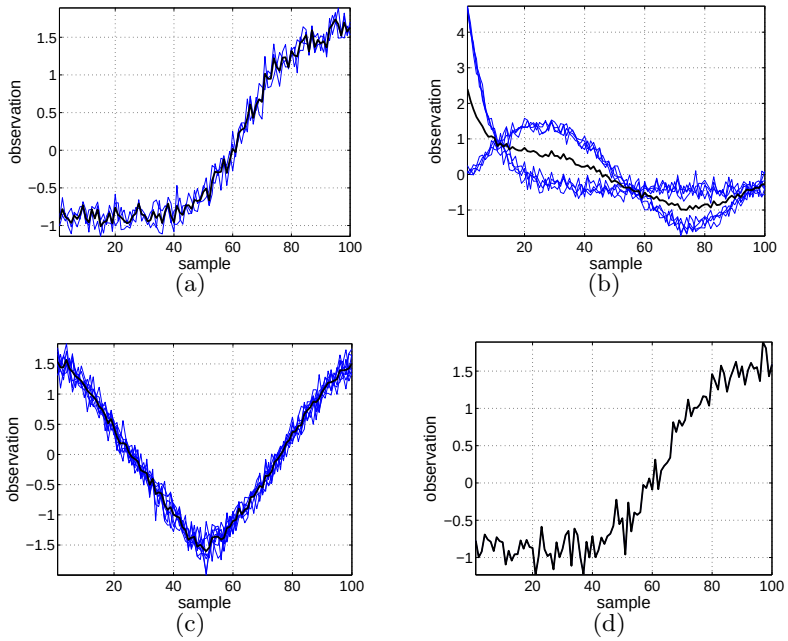


Figure 6.6: Data distribution in each cluster for the first clustering run (thicker line represents the centroid): (a) Cluster 1; (b) Cluster 2; (c) Cluster 3; (d) Cluster 4.

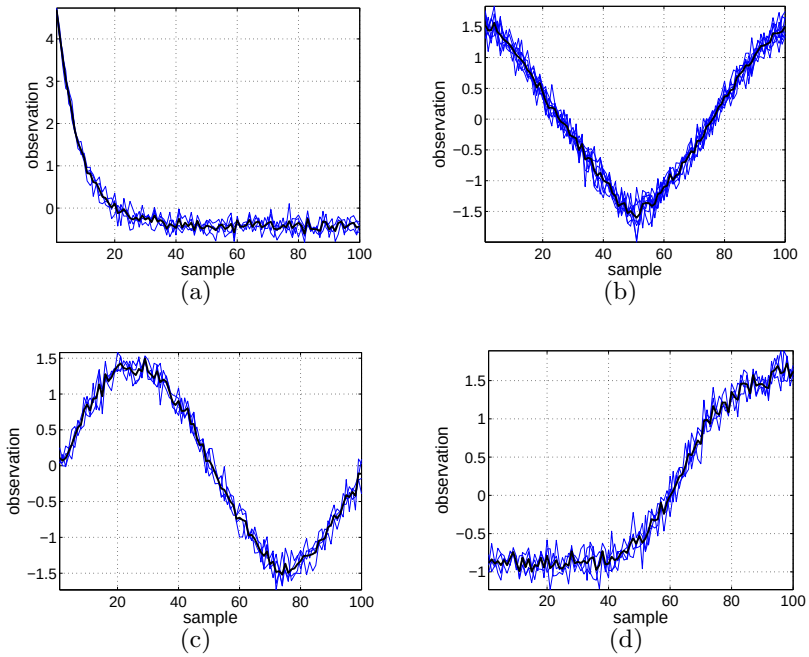


Figure 6.7: Data distribution in each cluster for the second clustering run (thicker line represents the centroid): (a) Cluster 1; (b) Cluster 2; (c) Cluster 3; (d) Cluster 4.

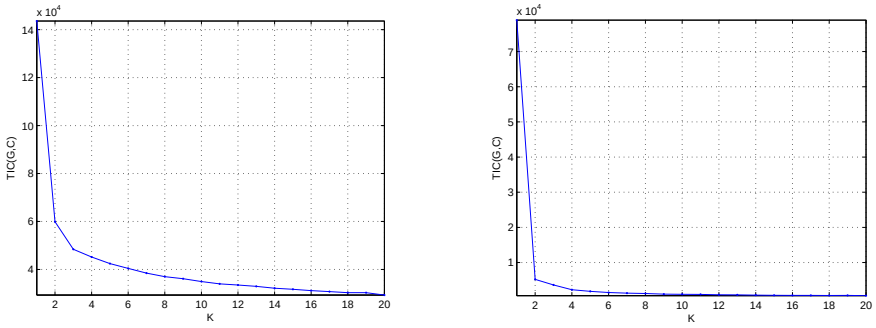


Figure 6.8: $TIC(\mathsf{G},\mathsf{C})$ changes as a function of K for classical K-means clustering. (a) IDS1 data (b) IDS1Filt data

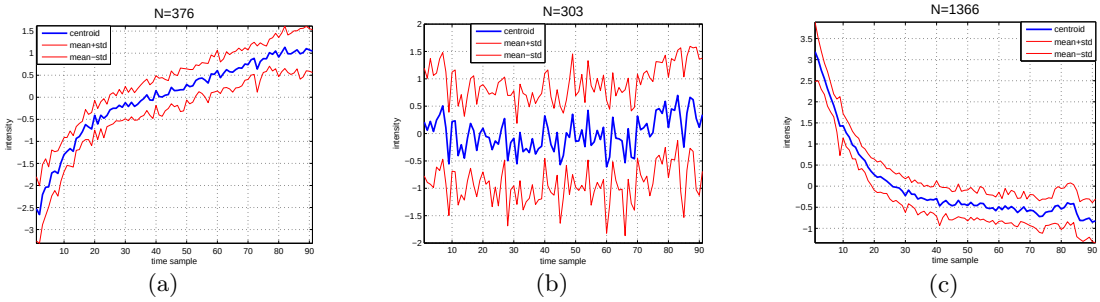


Figure 6.9: Centroid of each cluster for K-means clustering on IDS1 (thinner lines = the standard deviation measures of the intensity changes for the objects contained in each cluster, N = the number of objects): (a) Centroid 1; (b) Centroid 2; (c) Centroid 3.

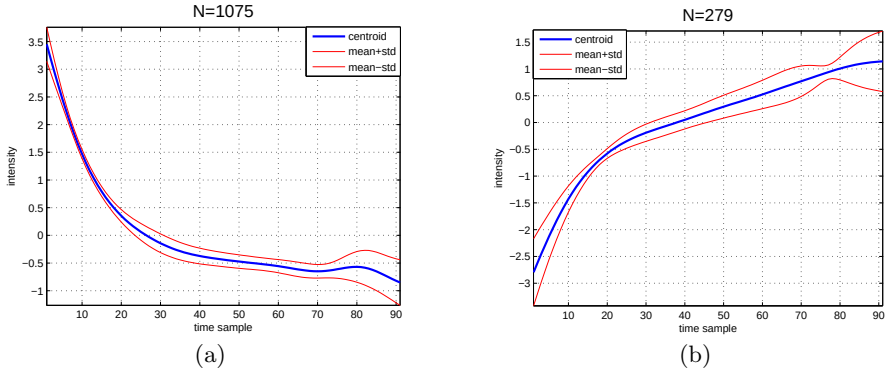


Figure 6.10: Centroid of each cluster for K-means clustering on IDS1Filt (thinner lines = the standard deviation measures of the intensity changes for the objects contained in each cluster, N = the number of objects): (a) Centroid 1; (b) Centroid 2.

6.3 Self Organizing Maps

A self organizing map (SOM) [90] is a class of artificial neural network that is trained using competitive learning to produce a low dimensional representation of high dimensional data. Competitive learning employs a winner-takes-all policy [56]: for each input, the output neurons of the network compete among themselves to be activated. The output neurons are placed at the nodes of a lattice (usually one or two dimensions). During the competitive learning process, the locations of the output neurons continue to be ordered according to the various patterns of the inputs until a self organizing map is formed.

The most important applications of the SOM are in the visualization of complex processes and systems, which are otherwise difficult or even impossible to be detected by direct human observations [72]. The SOM algorithm is unsupervised, *i.e.* the training process is entirely data-driven, without the requirement of any *prior* knowledge. As compared to artificial neural networks based on supervised learning (*e.g.* back propagation neural networks), this is a clear advantage. Typical processes for achieving a self organizing map involve competition, cooperation and synaptic adaptation [56].

6.3.1 Algorithm Description and Basic Operation

Competitive Process

For each input, the output neurons in the network compute their value of a discrimination function and only the output neuron achieving the maximum value is activated. Let $\mathbf{x}$ denote an input vector

$$\mathbf{x} = [x_1, x_2, \ldots, x_m]^{\mathrm{T}} \tag{6.5}$$

and $\mathbf{w}_j$ denote the weight vector of output neuron j

$$\mathbf{w}_j = [w_{j1}, w_{j2}, \ldots, w_{jm}]^{\mathrm{T}}, \text{ for } j = 1, 2, \ldots, l, \tag{6.6}$$

where l is the total number of output neurons in the network. The output neuron, i, that best matches the input vector $\mathbf{x}$ can be described as

$$i(\mathbf{x}) = \arg\min_j \|\mathbf{x} - \mathbf{w}_j\|_2, \text{ for } j = 1, 2, \ldots, l. \tag{6.7}$$

Cooperative Process

SOMs are different from other neural networks in that SOMs use neighbourhood functions to preserve the topological neighbourhood centered on winning neuron i [56]. Suppose h_{ij} denotes the topological neighbourhood of i,

$$h_{ij} = \exp(-\frac{d_{ij}^2}{2\sigma(n)^2}), \tag{6.8}$$

where d_{ij} denotes the L_2-norm distance between winning neuron i and its neighbourhood neuron j in the output space,

$$d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|_2, \tag{6.9}$$

$\mathbf{r}_i$ and $\mathbf{r}_j$ define the discrete position of winning neuron i and excited neuron j in the lattice, respectively. The parameter $\sigma(n)$ defines the neighbourhood radius [135],

$$\sigma(n) = \sigma_0 \exp(-\frac{n}{\tau_1}), \tag{6.10}$$

where n denotes the number of iterations. The neighbourhood radius shrinks or decreases with the number of iterations, leading to the decrease of h_{ij} for a given d_{ij}. By definition, h_{ij} attains its maximum when $d_{ij} = 0$. h_{ij} decreases monotically with increasing d_{ij} and tends to zero as $d_{ij} \to \infty$.

Synaptic Adaptation

Using SOM, the synaptic weight vector $\mathbf{w}_j$ of neuron j is required to be updated according to the input vector $\mathbf{x}$ from iteration n to $n+1$, according to

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n) h_{ij}(n)(\mathbf{x} - \mathbf{w}_j(n)), \tag{6.11}$$

where $\eta(n)$ is called the learning rate,

$$\eta(n) = \eta_0 \exp(-\frac{n}{\sigma_2}),\ n = 0, 1, 2, \ldots. \tag{6.12}$$

The adaptation of synaptic weights can be decomposed into two phases: ordering phase and convergence phase. In the first phase, the learning rate and neighborhood radius are large resulting in a fast training. The values of these two parameters decay with the number of iterations and slow down the tuning process, so the second phase is also

called a fine tuning phase. A detailed discussion of how these two phases are achieved by the synaptic weighting update is given in [91].

As training proceeds, patterns of output neurons around activated neuron i are adapted to become more like the pattern of neuron i. At the end of training, every object is mapped with an output neuron and the neighbour output neurons tend to have similar patterns.

Here, the SOM toolbox developed by Vesanto *et al* is employed [162]. The basic steps of a SOM algorithm can thus be summarised as follows [56]:

Step 1: Initialization. Choose random values for $\mathbf{w}_j,\ j = 1,\ \cdots,\ l$.

Step 2: Sampling. A training sample is randomly selected from the set of input vectors.

Step 3: Similarity Matching. Find the best matching (winning) neuron at iteration n for an input neuron, $\mathbf{x}$, according to Eq. (6.7).

Step 4: Updating. Update the synaptic weight vectors for all output neurons, according to Eq. (6.11).

Step 5: Continuation. Go back to Step 2 until no noticeable changes are observed in the self organizing map. In practice, the algorithm is terminated when a predefined maximum number of clusters have taken place.

When applying the SOM for clustering, the output neurons are equivalent to the cluster centroids in K-means and the samples for which a neuron is the winner are the objects contained in the corresponding cluster.

6.3.2 Estimating Algorithm Effectiveness on Simulated Data

The data is standardised to have zero mean and unit variance to focus on shape difference. The experimental results of applying the SOM to clustering of the SDS1

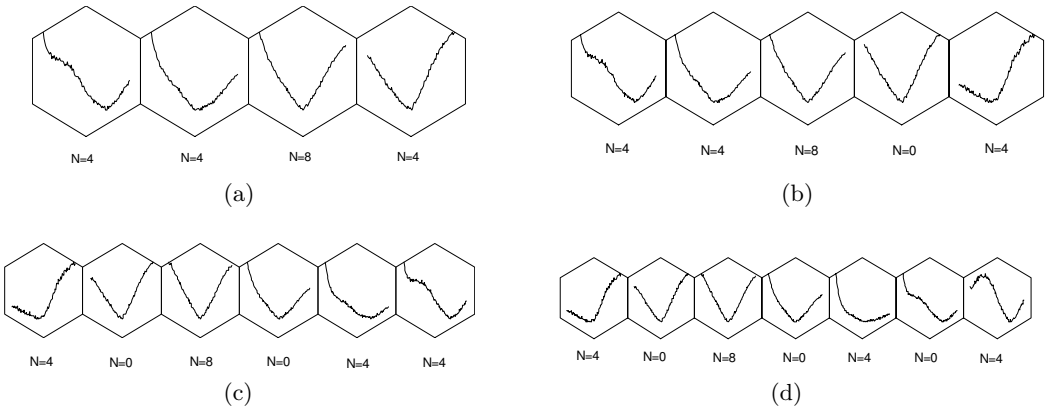


Figure 6.11: Experimental results of using SOM on the SDS1 data: (a) Synaptic weight vectors for 4 output neurons; (b) Synaptic weight vectors for 5 output neurons; (c) Synaptic weight vectors for 6 output neurons; (d) Synaptic weight vectors for 7 output neurons (N denotes the number of samples for which a neuron is a winner.)

benchmark demonstrate that the more output neurons used, the more patterns can be explored. When the number of output neurons is arbitrarily set to 4, as Fig. 6.11 (a) shows, there are 2 different patterns shown in the output neurons (represented by the synaptic weight vectors). Increasing the number of output neurons to 5, one more pattern is included (as shown in the last cell in Fig. 6.11 (b)). When the number reaches to 7, all patterns contained in the SDS1 are recognized.

One intrinsic feature of the SOM refers to the so-called topological-ordering property [56], which shows that the output neurons of the SOM feature map are ordered topologically based on the similarity between neurons patterns. This is in fact, a direct consequence of the method used to update the synaptic weighting as defined in Eq. (6.11). In order to include all the patterns in output neurons, the number of neurons needs to be set bigger than the number of distinctive patterns contained in the data to allow for the gradual transition between patterns.

6.3.3 Application of SOM to OES Data

IDS1 and IDS1Filt are used to test the effectiveness of the SOM in clustering high-dimensional OES data. Without any *prior* knowledge, the selection of the structure

of the feature map is arbitrary. For ease of visualisation and taking into account the desire to achieve data reduction, the number of output neurons is selected to be 16 and 25, respectively. Iteration times for training and fine tuning processes are set to 1000 and 100, respectively, as suggested in [56] for dealing with large data sets.

Fig. 6.12 shows the synaptic weight vectors (patterns) of the output neurons arranged in a lattice size of 1×16 for IDS1 and IDS1Filt, respectively. As one can see, output neurons with similar patterns are located near each other and the neurons with the most distinctive patterns are the furthest apart. For IDS1Filt, the patterns of some output neurons are destroyed by the noise (as in Fig. 6.12 (a)), making it hard to estimate the number of different patterns contained in the feature map.

The patterns obtained when the lattice geometry is changed to a 4×4 structure are shown in Fig. 6.13 (a) and (b) for the SOM on IDS1 and IDS1Filt, respectively. As can be observed, roughly 3 different patterns are extracted for IDS1 (Fig. 6.13 (a)) and IDS1Filt (Fig. 6.13 (b)). Comparing Fig. 6.12 (a) with Fig. 6.13 (a) and Fig. 6.12 (b) with Fig. 6.13 (b), one can find that the number of objects that the output neurons with similar patterns can represent varies if the neurons are arranged in the lattice of different geometry. Because the selection of the lattice geometry is generally arbitrary, this drawback really reduces the reliability of the results obtained by the SOM. As one more example, more output neurons are employed. Fig. 6.14 show the patterns of 25 output neurons arranged in a lattice size of 5×5 for IDS1 and IDS1Filt, respectively. As compared to Fig. 6.12, the patterns of output neurons and the times that each neuron gets hit are quite different even for the same data set. Thus the SOM is sensitive to the setting of lattice geometry (lattice structure and size).

Based on the analysis of using the SOM for classifying the SDS1, IDS1 and IDS1Filt data sets, it can be concluded that the SOM is not an appropriate method for recognising the distinctive patterns in the data. Moreover, for dealing with data sets with a large number of objects, say 2000, the number of output neurons may be required to be at least a few hundreds due to the topological-ordering property, which is computationally impractical.

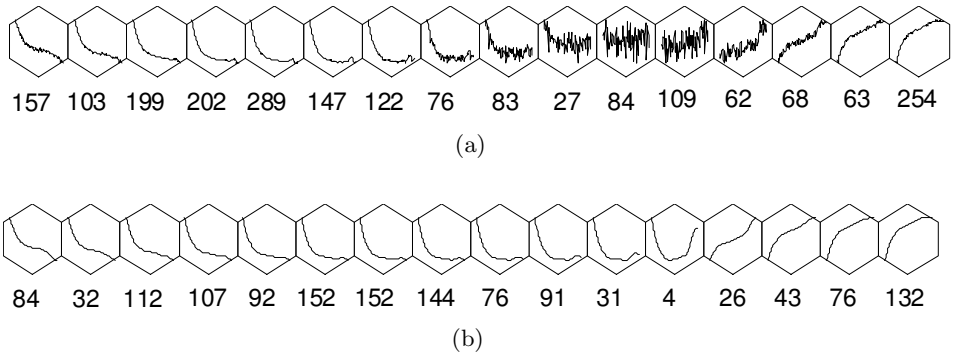


Figure 6.12: Synaptic weight vectors for 16 output neurons in 1-D lattice: (a) SOM on IDS1; (b) SOM on IDS1Filt

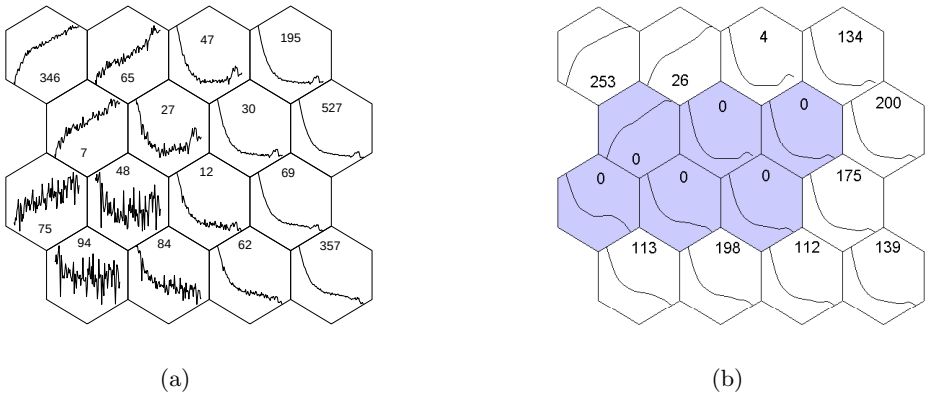


Figure 6.13: Synaptic weight vectors for 16 output neurons in 2-D lattice size of 4×4: (a) SOM on IDS1; (b) SOM on IDS1Filt.

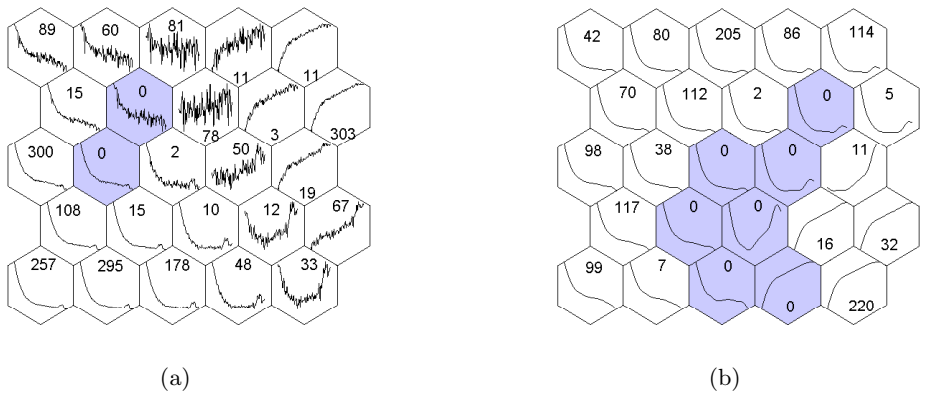


Figure 6.14: Synaptic weight vectors for 25 output neurons arranged in 2-D lattice size of 5×5: (a) SOM on IDS1; (b) SOM on IDS1Filt.

6.4 Quality Threshold Clustering Algorithm

A common problem with the K-means clustering and SOM is that the number of clusters needs to be set in advance. Moreover, for the SOM, the lattice geometry is also required and for the K-means clustering, the clustering result is unstable. Quality threshold (QT) clustering [58] has been developed to avoid many of these problems, but is specialised for clustering gene expression data.

6.4.1 Basic Operation

The focus of QT clustering is to find large clusters that have a so-called quality guarantee [58], which is defined in terms of a similarity measure for the objects in the cluster. A cluster is said to have a quality value or diameter of ϵ, if for every object in the cluster, there is at least one other object such that the similarity between them is less than ϵ. The quality of cluster, G, can thus be expressed as

$$Q(G) = 1 - \min_{\mathbf{x}_i \in G} \{ \max_{\mathbf{x}_j \in G, \mathbf{x}_i \neq \mathbf{x}_j} [jcorr(\mathbf{x}_i, \mathbf{x}_j)] \}, \tag{6.13}$$

where $jcorr(\cdot)$ denotes the jackknife correlation coefficient [58]. The QT algorithm works as follows:

Step 1: Each object is taken in turn as a cluster starting point, and a cluster is built up to contain all those objects for which the cluster quality is satisfied ($Q \leq \epsilon$).

Step 2: The cluster with the largest number of objects is selected and the corresponding objects are removed from the data set.

Step 3: The process is repeated from Step 1 on the remaining data, until no objects are left.

The pseudocode for the algorithm is given as follows [58].

while $card(\mathbf{X}) > 0$ **do**
  for $i = 1$ to $card(\mathbf{X})$ **do**
    $G_i = \mathbf{x}_i$, (object $\mathbf{x}_i$ is taken as a starting object for candidate cluster G_i.)
    $flag = true$
    while $flag$ **do**
      $\mathbf{x}_p = \arg\max_{\mathbf{x}_j \in \mathbf{X}/\mathbf{x}_i}[Q(G_i \cup \mathbf{x}_j)]$
      if $Q(G_i \cup \mathbf{x}_p) \leq \epsilon$ **then**
        $G_i = G_i \cup \mathbf{x}_p$
      else
        $flag = false$;
      end if
    end while
  end for
  $G_i^* = \arg\max_{G_i}\{card(G_i)\}$. (find the cluster with maximum cardinality)
  $\mathbf{X} = \mathbf{X}/G_i^*$. (remove selected cluster from $\mathbf{X}$)
end while

QT clustering is customised in this work with Pearson's correlation coefficient used as the dissimilarity measure. In comparison to existing clustering algorithms such as the classical K-means and SOM, the QT clustering has a few advantages. First, as is defined in the algorithm, each object initiates a candidate cluster, so the clustering is not affected by the order in which the similarity data appears. Secondly, the number of clusters is not required at the start of the algorithm. Moreover, the algorithm always returns the same result no matter how many times it is run and finally, the solution obtained is the global solution [58].

One key issue with QT clustering is how to select the threshold. Here, the total intra-cluster sum squared error, $TIC(\mathsf{G}, \mathsf{C})$ (as defined in Eq. (6.1)), is employed as a metric to compare threshold values. As QT does not produce representative objects for each cluster, as a post-processing step, the object having the biggest correlation with the other intra-cluster objects is selected, used for calculating $TIC(\mathsf{G}, \mathsf{C})$. Fig. 6.15 shows

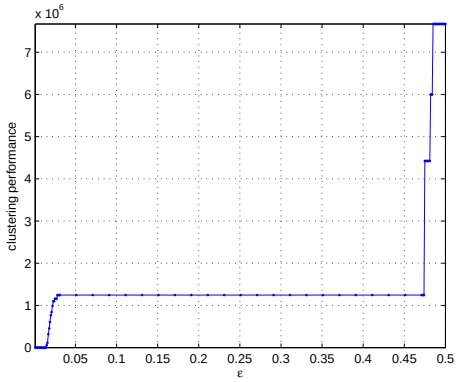


Figure 6.15: $TIC(\mathsf{G}, \mathsf{C})$ changes as a function of ϵ for QT clustering on SDS1.

$TIC(\mathsf{G}, \mathsf{C})$ changes as a function of ϵ for SDS1. The minimum of $TIC(\mathsf{G}, \mathsf{C})$ is obtained when $\epsilon = 0.01$, which corresponds to the case where every single object is assigned to a cluster. The 'elbow point' occurs when $\epsilon = 0.03$ (any $TIC(\mathsf{G}, \mathsf{C})$ values for $\epsilon \in$ [0.03 0.47] are equal). Fig. 6.16 shows the corresponding clustering results. 4 clusters are obtained and the objects with similar patterns are correctly separated in different clusters.

6.4.2 Application of QT to OES Data

To further explore the operation of QT clustering, it is applied to the high-dimensional IDS1 and IDS1Filt data sets. $TIC(\mathsf{G}, \mathsf{C})$ changes as a function of ϵ for IDS1 and IDS1Filt are shown in Fig. 6.17. The 'elbow point' occurs at $\epsilon = 0.04$ and $\epsilon = 0.03$ for IDS1 and IDS1Filt, respectively.

Using $\epsilon = 0.04$ for QT clustering on IDS1, 320 clusters are obtained, among which, 275 clusters are single-object. It is not feasible to visualise the patterns in each cluster, given the number involves, hence only the big clusters are considered. Here, the first 8 clusters are selected with each containing no less than 9 objects. The patterns of the objects contained in these 8 clusters (with data standardised) are shown in Fig. 6.19. As can be seen from Fig. 6.19, the first cluster contains 1464 objects, while the patterns included are distinct. Because of the existence of noise, the clusters obtained are not effective in summarising the patterns contained in the data set.

Applying QT to IDS1Filt with $\epsilon = 0.03$ returns 4 clusters. The patterns of objects

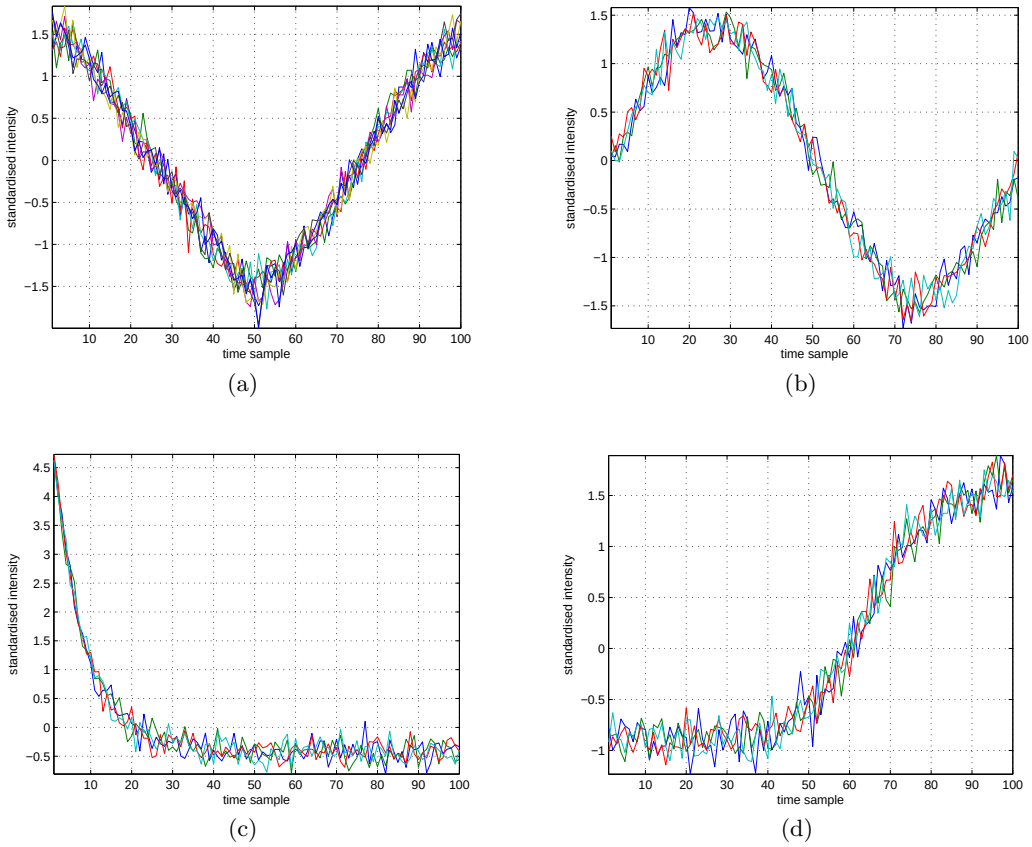


Figure 6.16: Patterns of the 4 clusters obtained by applying the QT clustering to SDS1 (the threshold set to 0.1): (a) Cluster 1 (object 9 to 16); (b) Cluster 2 (object 1 to 4); (c) Cluster 3 (object 5 to 8); (d) Cluster 4 (object 17 to 20).

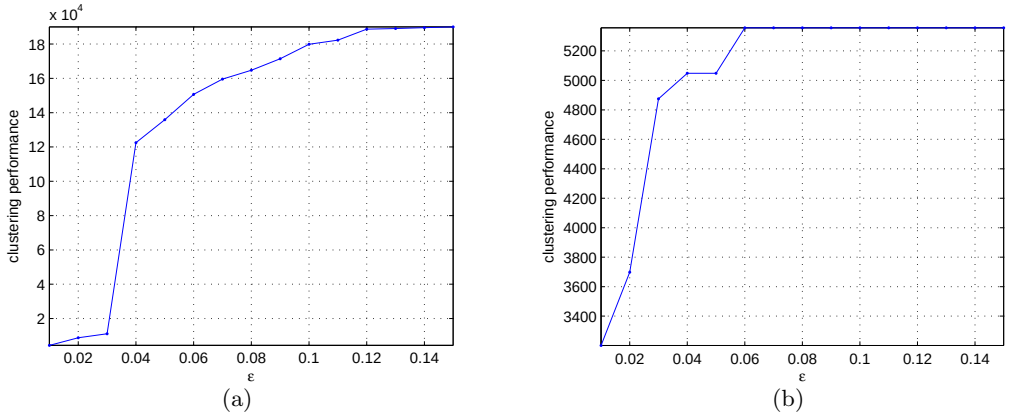


Figure 6.17: $TIC(\mathsf{G}, \mathsf{C})$ changes as a function of ϵ for QT clustering: (a) IDS1; (b) IDS1Filt.

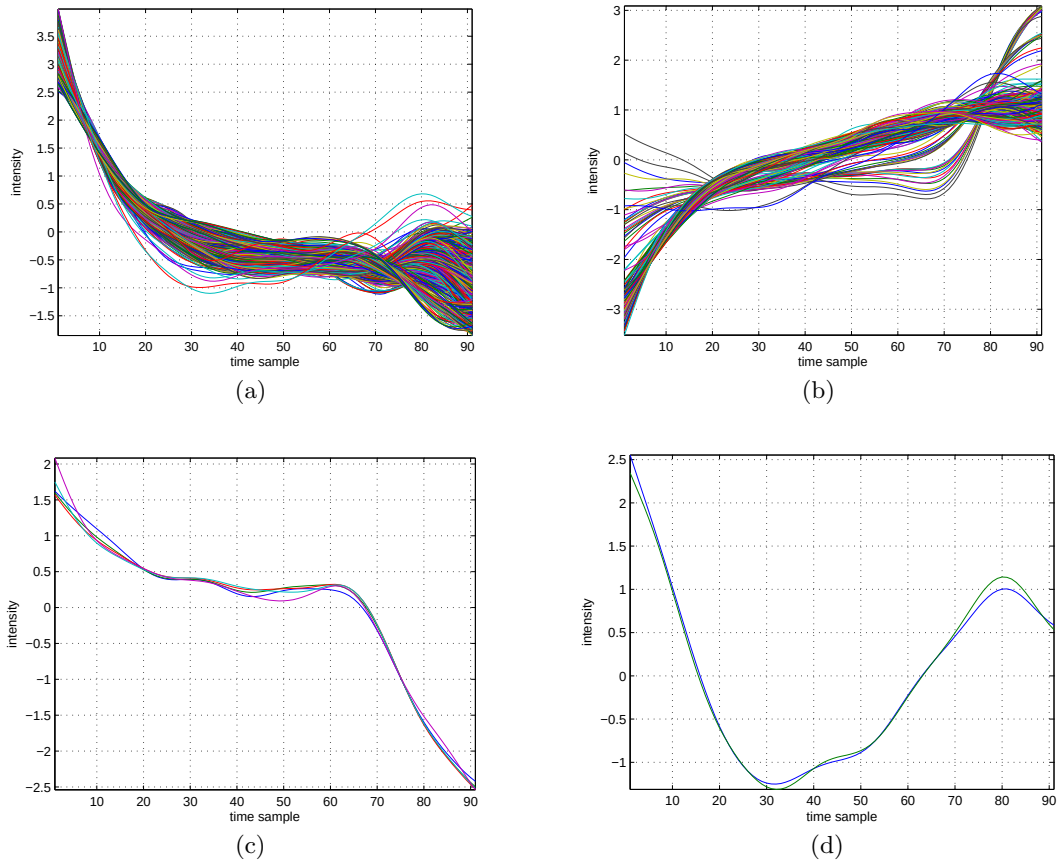


Figure 6.18: The 4 clusters obtained by QT clustering on IDS1Filt ($\epsilon = 0.03$): (a) Cluster 1, size 1068; (b) Cluster 2, size 279; (c) Cluster 3, size 5; (d) Cluster 4, size 2.

contained in these 4 clusters (with data standardised) are shown in Fig. 6.18). As can be observed, the patterns of intra-cluster objects contained in Cluster 1 and Cluster 2 are distinct. The reason is that the objective of QT clustering is to seek large clusters, rather than the distinctive patterns. Thus as long as the pattern of a new object is similar to that of an assigned object, the new object can be included in the cluster that contains this assigned object. Therefore, for pattern recognition, the QT clustering is not effective.

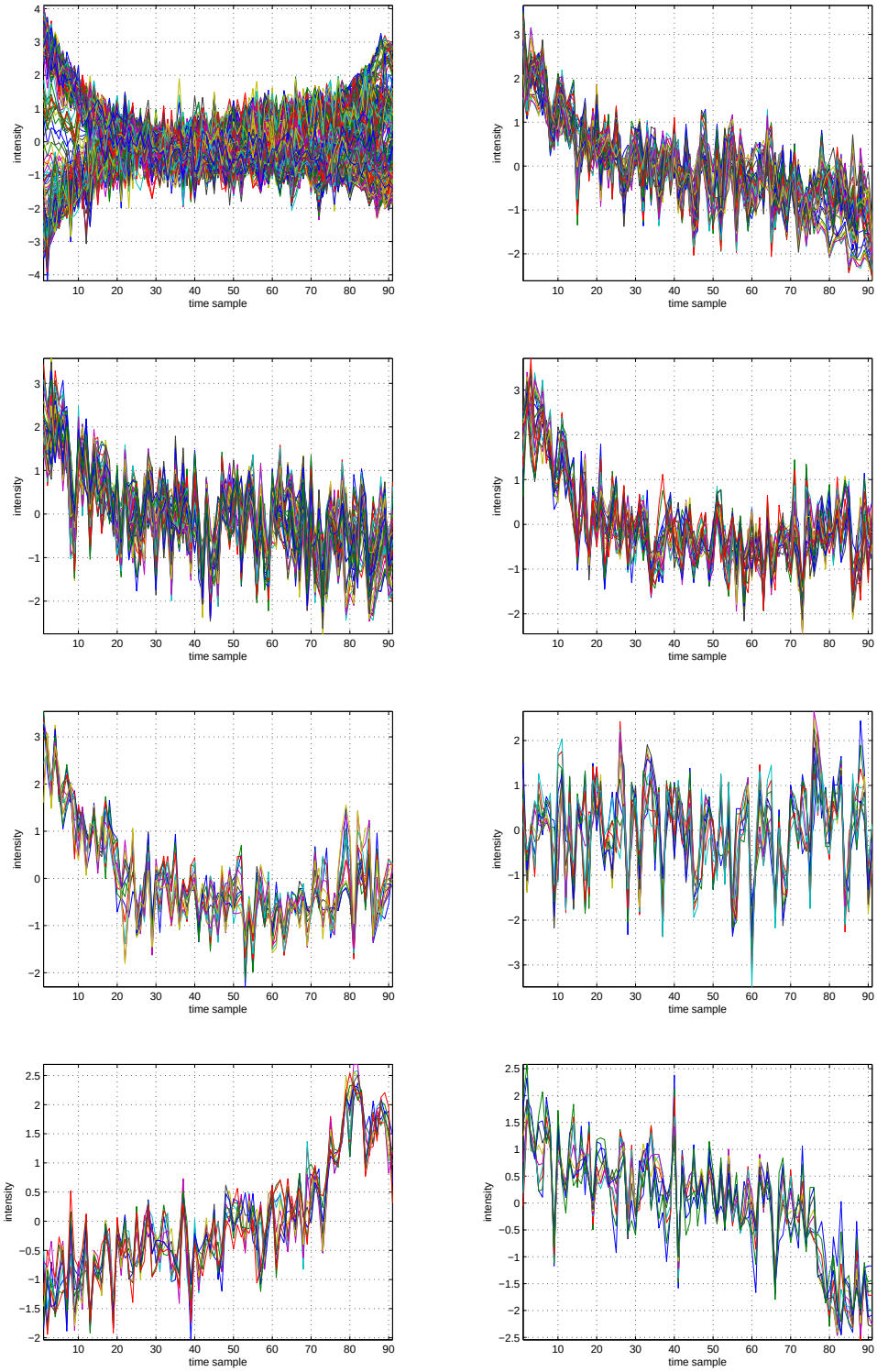


Figure 6.19: The patterns of the 8 largest clusters obtained by applying QT clustering to IDS1 ($\epsilon = 0.04$): (a) Cluster 1, size 1464; (b) Cluster 2, size 49; (c) Cluster 3, size 44; (d) Cluster 4, size 24; (e) Cluster 5, size 13; (f) Cluster 6, size 11; (g) Cluster 7, size 10; (h) Cluster 8, size 9.

6.5 Max Separation Clustering Algorithm

When dealing with a data set with over 2000 dimensions and without complete knowledge of the data patterns, clustering is a candidate method for exploring and defining the data patterns. Ideally, the clustering should be designed to find the most distinctive set of features so that each cluster is maximally different from all previous clusters.

Our proposed algorithm, the Max Separation Clustering (MSC), is in principle a correlation-based clustering algorithm. Clustering by employing correlation as the similarity function can group the OES signals that evolve similarly over time into the same cluster and avoids the issues with the scale ambiguity/uncertainty associated with OES signals. A key point for this algorithm is how to select the representative object for each cluster. Here, a joint use of the MaxMin criterion [120] and the single LINK-age method [7] is employed.

The MaxMin criterion stresses that a new centroid should be the farthest object from the existing clusters, when compared with the other unassigned objects and the single LINK-age method gives a way of defining the distance between two clusters (the shortest distance between any two elements in the two clusters). Details of the joint use of these two methods are described in Step 5 of the MSC algorithm. Because the representative object is not a centroid (average of all the objects in a cluster), nor a so-called medoid [120] (the closest object to the centroid), the object is named as maxoid in our algorithm. The maxoid is the object in a given cluster which is the furthest from all objects in the existing clusters.

A complete description of the algorithm follows

Step 1: Identify the maxoid candidates.

For a given set of objects $\mathbf{X}$, identify the objects $\mathbf{x}_i^*$ and $\mathbf{x}_j^*$ which are farthest apart, *i.e.*

$$(\mathbf{x}_i^*, \mathbf{x}_j^*) = \arg\max_{\mathbf{x}_i, \mathbf{x}_j} \{dist(\mathbf{x}_i, \mathbf{x}_j)\}, \tag{6.14}$$

where $dist(\cdot)$ is the L_2 norm.

Step 2: Select $\mathbf{x}_i^*$ as the new maxoid, where by definition $pow(\mathbf{x}_i) > pow(\mathbf{x}_j)$ (where $pow(\cdot)$ denotes the signal power) and

$$\mathbf{m}_{\text{new}} = \mathbf{x}_i^*. \tag{6.15}$$

The set of maxoids, M, is initiated as $\mathsf{M} = \{\mathbf{m}_{\text{new}}\}$.

Step 3: Assign all objects $\mathbf{x}_i \in \mathbf{X}$ to a new cluster, G_{new}, according to the condition

$$G_{\text{new}} = \{\mathbf{x}_i \in \mathbf{X} \mid corr(\mathbf{x}_i, \mathbf{m}_{\text{new}}) \geq \xi, \forall\ \mathbf{x}_i \in \mathbf{X}\}. \tag{6.16}$$

Note that $corr\{\cdot\}$ is a function that measures the similarity between $\mathbf{x}_i$ and $\mathbf{m}_{\text{new}}$ and ξ is the similarity threshold, which defines the desired similarity level. Here, $corr(\cdot)$ is chosen as the Pearson product-moment correlation coefficient [26],

$$corr(\mathbf{x}, \mathbf{y}) = \frac{1}{m-1} \sum_{i=1}^{m} (\frac{x_i - \overline{x}}{\sigma_x})(\frac{y_i - \overline{y}}{\sigma_y}) \tag{6.17}$$

where $\mathbf{x} = \{x_1, \ldots, x_m\}$, $\mathbf{y} = \{y_1, \ldots, y_m\}$, the means of the x-value and the y-value are $\overline{x}$ and $\overline{y}$ and their standard deviations are σ_x and σ_y.

Add this new cluster into G, $\mathsf{G} = \{G_1, \ldots, G_{\text{new}}\}$.

Step 4: Check the stop condition.

(a) If there are no objects left, the algorithm has finished.

(b) Otherwise, allocate the unassigned objects to $\widetilde{\mathbf{X}}$ and continue.

Step 5: Find the next new maxoid.

Select the new maxoid as the object that is the furthest away from the existing set of clusters (G). Thus,

$$\mathbf{m}_{\text{new}} = \arg \max_{\mathbf{x}_i \in \widetilde{\mathbf{X}}} \{dist(\mathbf{x}_i, \mathsf{G})\}, \tag{6.18}$$

where $dist(\mathbf{x}_i, \mathsf{G})$ is defined as the distance from the object $\mathbf{x}_i$ to the closest object in G:

$$dist(\mathbf{x}_i, \mathsf{G}) = \min_{G_j \in \mathsf{G}} \min_{\mathbf{q} \in G_j} \{dist(\mathbf{x}_i, \mathbf{q})\}. \tag{6.19}$$

Add this new maxoid into M, $\mathsf{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_{\text{new}}\}$.

Step 6: Let $\mathbf{X} = \widetilde{\mathbf{X}}$ and return to Step 3.

The main operations in this algorithm can be summarized as follows. At each iteration, one unassigned object which is the furthest away from the objects in all existing clusters is selected to be the maxoid of a new cluster. Then, all unassigned objects that are similar to the new maxoid are allocated to this new cluster. The iteration is terminated when there are no unassigned objects left.

One thing to be stressed is that the threshold ξ, as defined in Eq. (6.16) is the only parameter that needs to be specified in this algorithm. A larger value of ξ imposes a stronger requirement for similarity between the objects in each cluster and as a result, leads to the generation of more clusters, *i.e.* as $\xi \to 1$, the number of clusters $K \to n$ (if the n objects are distinct).

An important property of MSC is that it is robust to outliers. In terms of data patterns, the outliers are distinct from the 'normal' objects, so most likely the outliers are discriminated and grouped in separate clusters. Thus MSC is more robust to outliers than the classical K-means and the SOM.

By design of SDS1, the similarity level between objects is known, making the selection of $\xi = 0.8$ applicable. In practice, such *priori* knowledge is rarely known, leading to the proposal of the threshold-selection method discussed in the next section. The application of MSC to SDS1 shows that there are 4 clusters in the data. The details about the number of objects, channel distributions, channel index of each maxoid, pattern of each maxoid and the range of power changes for the intra-cluster objects are given in Table 6.1. As one can see, MSC is effective at extracting the patterns contained in SDS1 and clustering the data having similar patterns. Moreover, the maxoid is an effective representative of the patterns contained in each cluster.

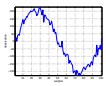
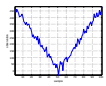
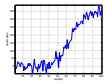
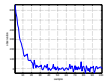
Channel index	Number of objects	Object distribution	Object index of maxoid	Range of power changes
1	4	1,2,3,4	2 	19686-21380
2	8	9,10,11,12, 13,14,15,16	16 	18807-21362
3	4	17,18,19,20	17 	19250-19992
4	4	5,6,7,8	5 	15590-16354

Table 6.1: Clustering results of MSC on SDS1.

6.6 Experiments on OES Data

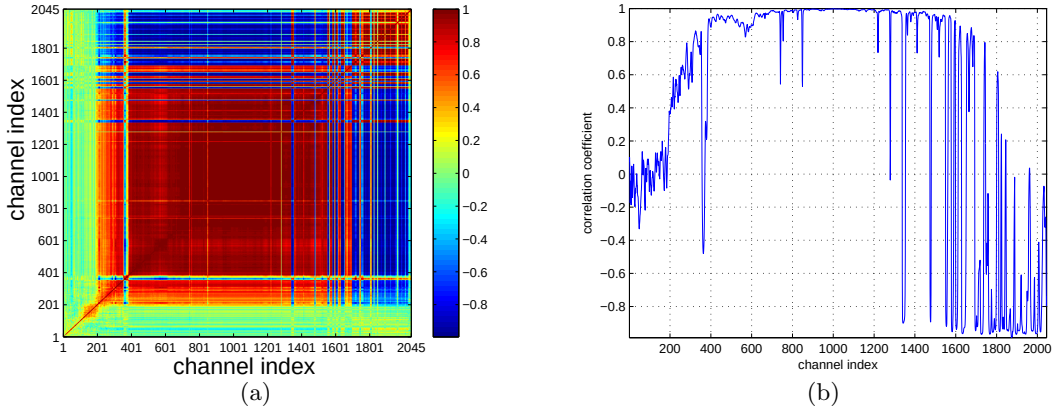
6.6.1 Selecting the Clustering Threshold

A simple correlation analysis of our OES data, as shown in Fig. 6.20, reveals that a high level of correlation exists among the OES channels, and is as high as 0.99 for some channels. The goal of clustering is to group the highly correlated channels together. In our algorithm, the task of allocating the channels to different clusters reduces to one of selecting the similarity threshold. A simple solution is to try different threshold values and then to pick one with good clustering performance.

Here, the total intra-cluster sum squared error, $TIC(\mathsf{G}, \mathsf{M})$, as defined in Eq. (6.1), is employed to measure the clustering performance, that is,

$$TIC(\mathsf{G}, \mathsf{M}) = \sum_{k=1}^{K} IC(G_k, \mathbf{m}_k). \tag{6.20}$$

Fig. 6.21 (a) shows the changes in $TIC(\mathsf{G}, \mathsf{M})$ as a function of the similarity threshold. As can be observed, the 'elbow point' occurs when the similarity threshold equals to 0.9. Fig. 6.21 (b) shows the changes in the number of clusters as a function of the similarity



(a) (b)

Figure 6.20: Correlation between the OES channels for IDS1: (a) The color at each point represent the correlation coefficient between the signals recorded at two channels; (b) The correlation between channel 1000 and the other OES channels.

threshold and the 'elbow point' occurs when the similarity threshold is 0.91. Switching the axis of the similarity threshold to the number of clusters, the rapid decrease in the clustering performance (occurring when the similarity threshold equals to 0.9) is offset by the fast increase in the number of clusters, leading to the resulting linear relationship as shown in Fig. 6.21 (c). The result demonstrates that the higher the value of similarity threshold, the lower $TIC(\mathsf{G}, \mathsf{M})$, but one should be aware that the lowest cost (zero) is obtained when every single channel is assigned to its own cluster, in which case the clustering is in fact meaningless. Therefore, the 'elbow point' criterion is employed to select the approximate similarity threshold of 0.9.

6.6.2 Clustering Results Using the Selected Threshold

As we have discussed, a value of 0.9 is selected as the similarity threshold for differentiating the different features in the OES data. Applying our proposed clustering algorithm to the OES data shows that the 2045 channels can be divided into 264 clusters (Fig. 6.22).

However, many details cannot easily be seen from Fig. 6.22, such as the number of channels in each cluster and the strength of the signals in each cluster. While it is not practical to show the details about the channel distribution for all the 264 clusters,

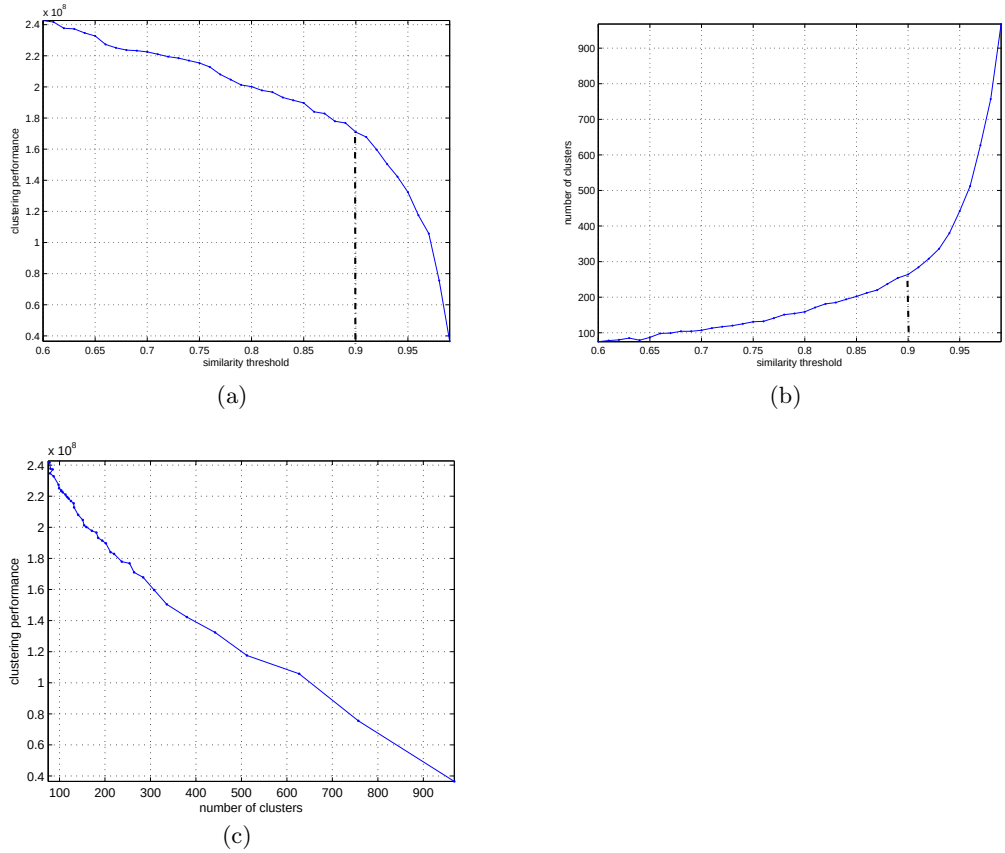


Figure 6.21: Selecting the similarity threshold: (a) Clustering performance changes as a function of the similarity threshold; (b) Number of clusters changes as a function of the similarity threshold; (c) Clustering performance changes over the number of clusters.

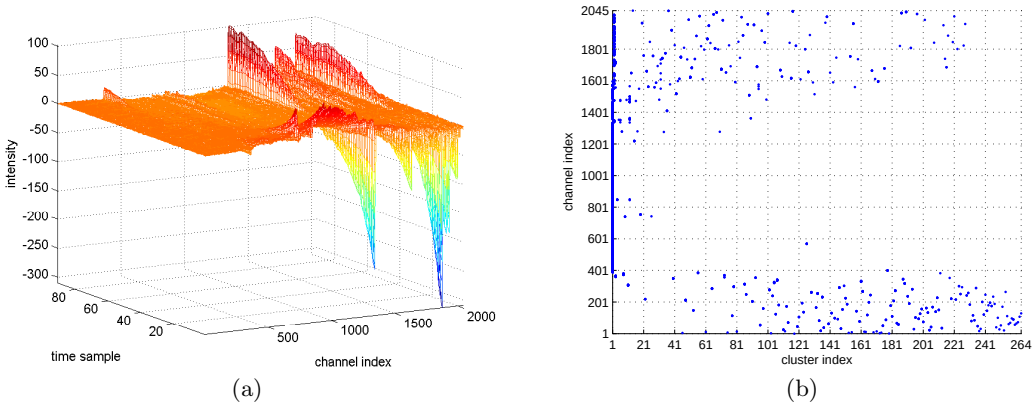


Figure 6.22: MSC of IDS1 with $\xi = 0.9$: (a) Mean centered IDS1; (b) Channel distribution in each cluster.

NO. of channels contained in each cluster	Average power of the channels in each cluster	No. of clusters
1081	88.1	1
261	475.8	1
21	132.6, 13.4	2
11	3.4	1
8	1.9, 1.5	2
7	5.3,2.7,2.4,1.5	4
6	3.5,1.6	2
5	0.52-23.46	21
4	0.48-21.42	21
3	0.17-5.77	78
2	0.16-7.58	40
1	0.16-10.28	91

Table 6.2: Simple statistics of the channel distribution in each cluster

some simple statistics on the channel distribution in each cluster are shown in Table 6.2. As an example, the 3rd entry shows that two of the clusters contain 21 channels and the average power of the channels in each cluster is 132.6 and 13.4, respectively (The power is unitless because the measure of the optical density is unitless). When there are a large number of clusters, *e.g.* 21, 78, 40 and 91 with the same number of channels, it is more useful to use plots (Fig. 6.23) to describe the power distribution across clusters, while showing the range of powers in the table.

6.6.3 Further Analysis of the Main Clusters

The term main clusters, is used here to refer to the clusters that contain a large number of channels and in our experiment, the large numbers are 1081, 261 and 21. To present an analysis for the main clusters, we firstly start with the cluster containing 1081 channels. A 3-D visualization of the data is displayed in Fig. 6.24 (a), while a 2-D visualisation of all channels is given in Fig. 6.24 (b), followed by the displaying of the maxoid in Fig. 6.24 (c).

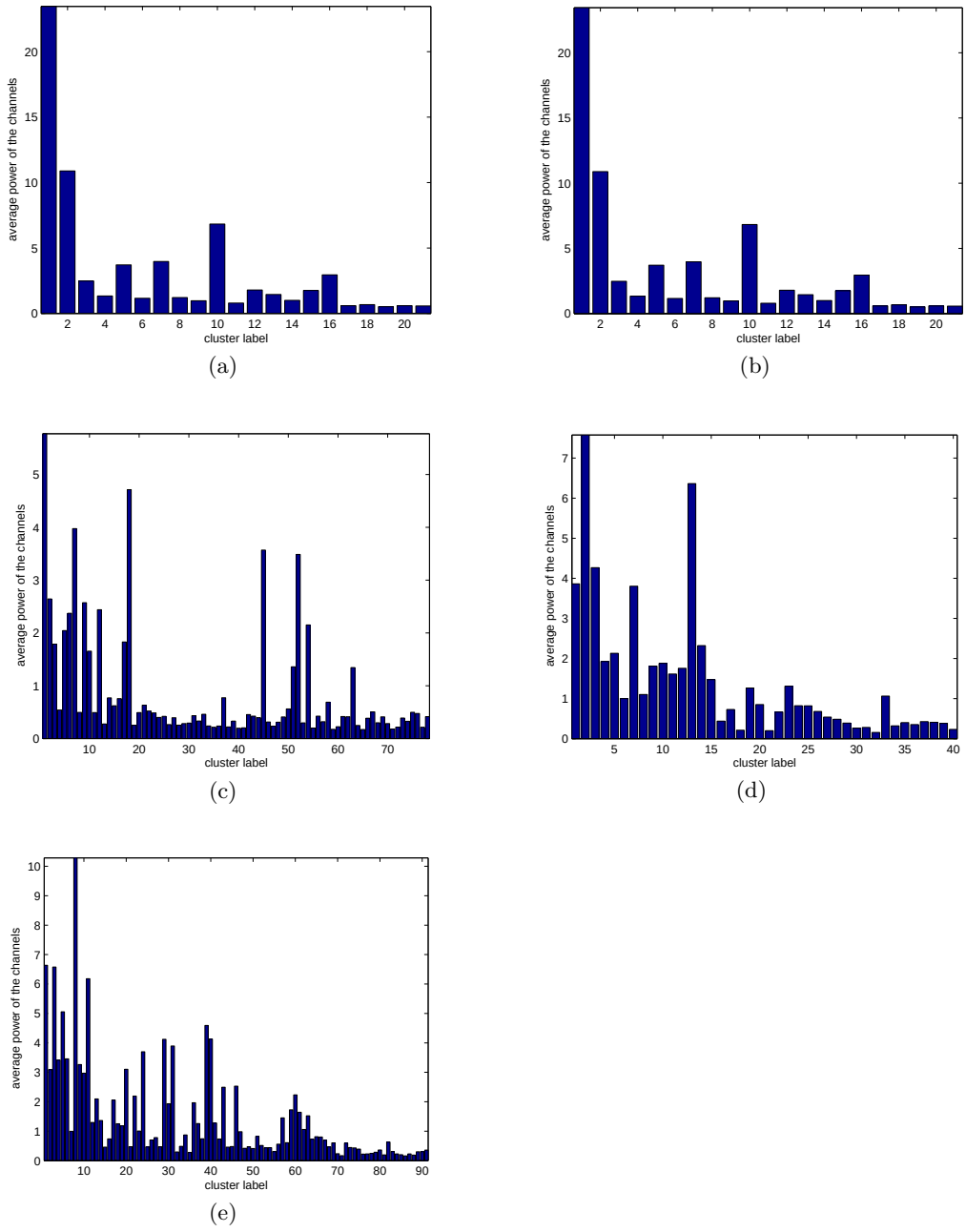


Figure 6.23: Average power distribution for the clusters having the same number of channels: (a) Clusters containing 5 channels in each one; (b) Clusters containing 4 channels in each one; (c) Clusters containing 3 channels in each one; (d) Clusters containing 2 channels in each one; (e) Clusters only containing 1 channel.

Since the similarity threshold is set to 0.9, the 1081 selected channels are highly correlated. Nevertheless, the high correlation may not be sufficient to guarantee that the intensity changes of the channels over time are similar or even the same. Hence, it is necessary to further check if the signal patterns in each cluster are similar.

Given that the different signals having different strength, as shown in Fig. 6.24 (d), we will first employ the normalization method (mean-centered and amplitude scaled) to eliminate the effect of signal strength. A 3D and 2D display of the normalized channels can be seen in Fig. 6.25 (a) and Fig. 6.25 (b), respectively. Fig. 6.25 (c) shows mean $\pm$ one standard deviation of the intensity changes over the 1081 channels. The results confirm that the signal patterns within the same cluster are quite similar, despite different signal strengths. Note that the mean profile is essentially the centroid of the cluster and the standard deviation gives an indication of the spread. A plot of the maxoid, centroid and medoid is shown in Fig. 6.25 (d).

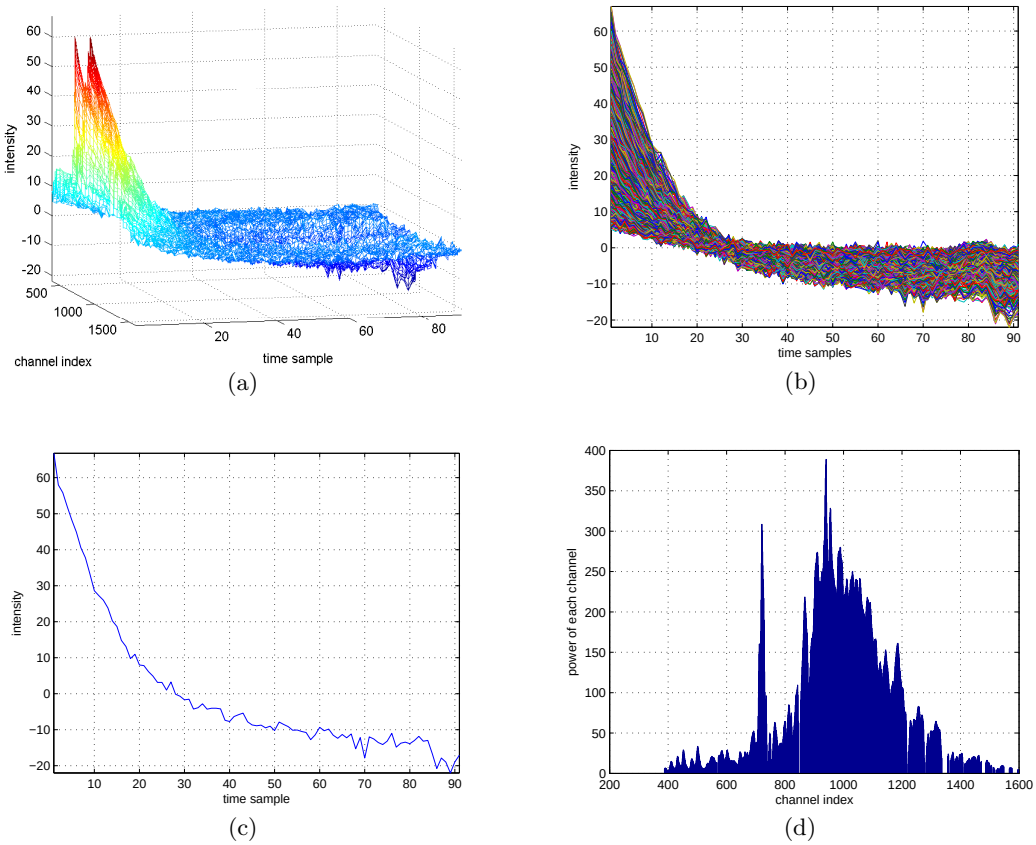


Figure 6.24: Cluster containing 1081 channels: (a) 3-D visualisation of the channels (b) 2-D visualisation of the channels; (c) The maxoid; (d) Channel power.

So far, the data patterns of the cluster containing 1081 channels have been examined. In what follows, we will apply the same method to examine the data patterns of the other main clusters. Fig. 6.26 summarises the data patterns for the cluster containing 261 channels, while Fig. 6.27 shows the two 21-channel clusters.

According to all the above analysis, it can be concluded that all the inter-cluster channel patterns are similar and different patterns can be preserved in different clusters. As such, the idea of using the correlation function as the similarity measurement function to differentiate different clusters is correct and the use of 0.9 as the threshold is also justified.

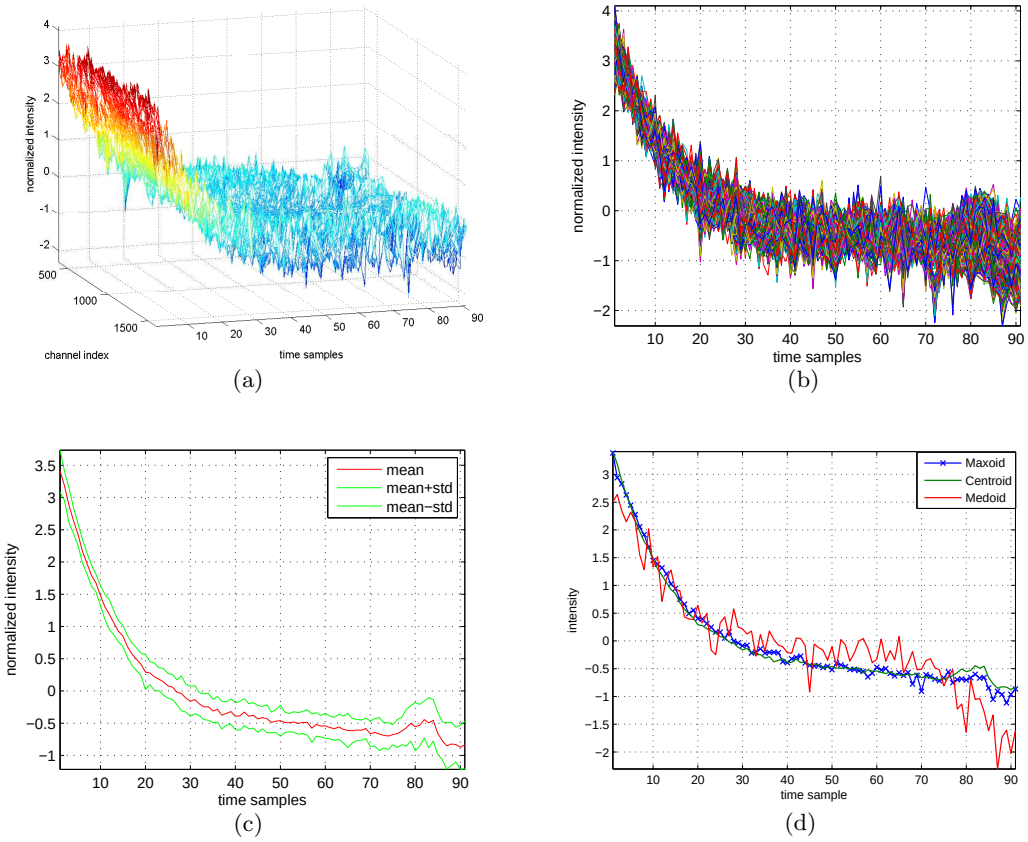


Figure 6.25: Normalized channels in the 1081-channel cluster: (a) 3D visualisation; (b) 2-D visualisation (c) Mean and standard deviation (std) measuring the range of the normalised intensity changes; (d) A plot of the maxoid, centroid and medoid.

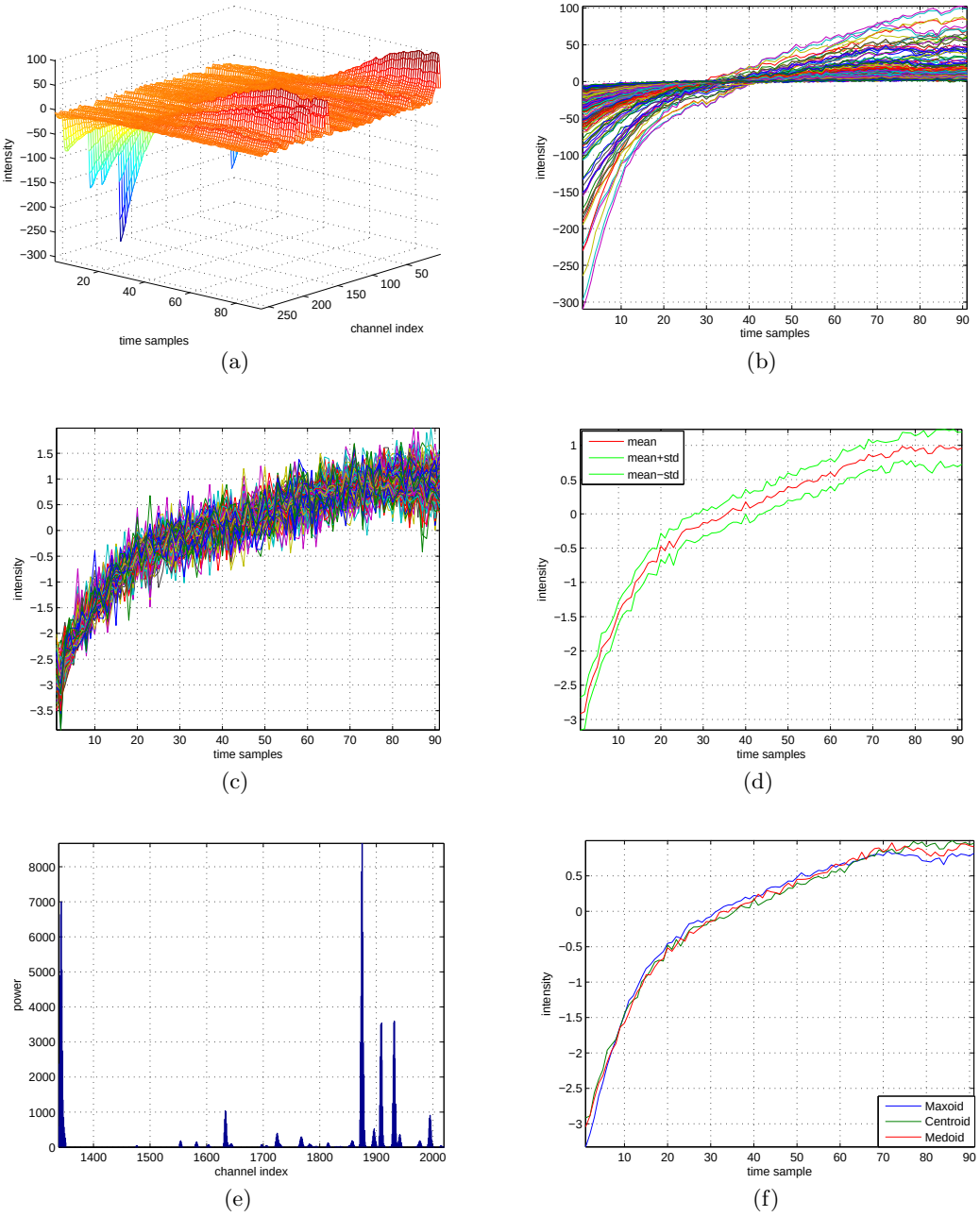


Figure 6.26: Cluster containing 261 channels: (a) 3-D visualisation of the channels (b) 2-D visualisation of the channels; (c) 2-D visualisation of the normalized channels; (d) Mean and standard deviation (std) measuring the range of the normalised intensity changes; (e) Channel power; (f) A plot of the maxoid, centroid and medoid (data normalised).

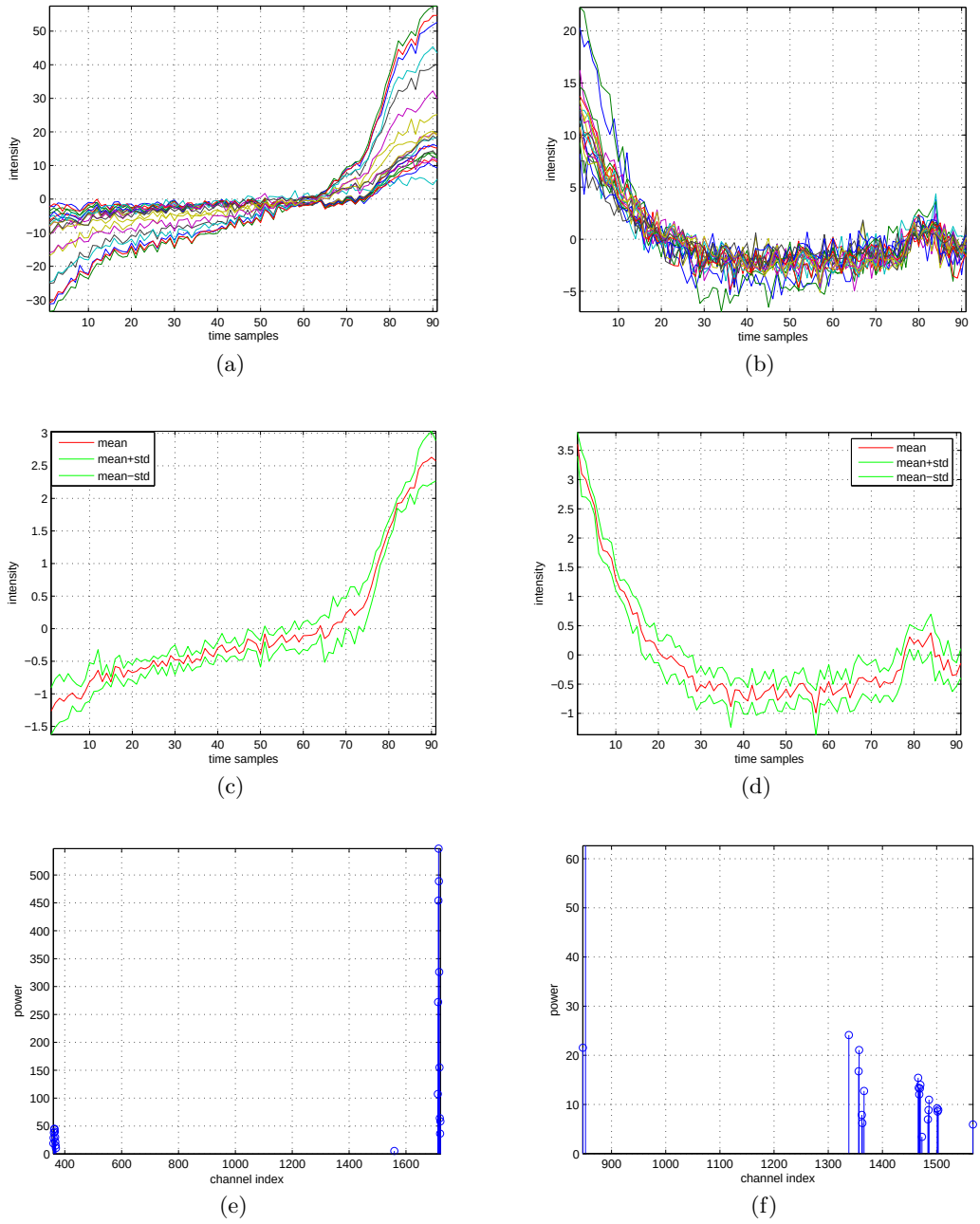


Figure 6.27: (a) 2D visualisation of the first 21-channel cluster; (b) 2D visualisation of the second 21-channel cluster; (c) Mean and standard deviation (std) measuring the range of the normalised intensity changes in the first 21-channel cluster; (d) Mean and standard deviation (std) measuring the range of the normalised intensity changes in the second 21-channel cluster; (e) Channel power for the first 21-channel cluster; (f) Channel power for the second 21-channel cluster.

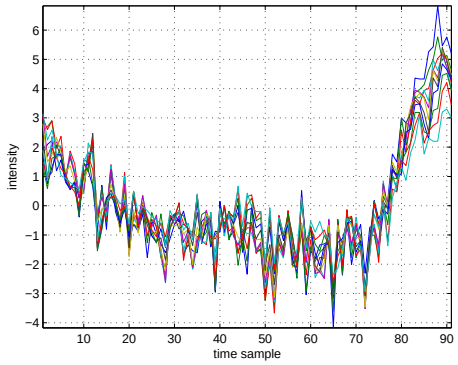


Figure 6.28: 2D display of the 11 channels in the same cluster

6.6.4 Further Analysis of the Sub-Clusters

As we have demonstrated in Table 6.2, in addition to the main clusters, there are still some other clusters containing a relatively smaller number of channels, *e.g.* from 2 to 11. We refer to these clusters as sub-clusters. There are 169 sub-clusters, which represents 64% of the total cluster count. Hence, it is important to analyze these sub-clusters as well. The clusters containing only one channel will be discussed separately.

We start with the analysis of the sub-clusters from the cluster containing 11 channels. The intensity changes of these 11 channels with time are displayed in Fig. 6.28. One probably has noticed that the changes in these channels happen nearly at the same time, which leads us to wonder if these channels are adjacent in frequency, since due to its finite resolution, the OES sensor tends to detect one wavelength emission over a number of adjacent channels, leading to redundancy in the OES measurements. To get the answer, we check the channel index and the result shows that the channel indices for these 11 channels are 372-382. To check if this is the case for all the other sub-clusters, we give the list of channels in each cluster in Table 6.3.

In fact, although there are some exceptions (marked by '+' in Fig 6.29), most of the channels that belong to one cluster are adjacent to each other. Another finding is that nearly all of these sub-cluster channels are located at channels below 400 or above 1400, which are actually the two ends of the OES sensor spectra.

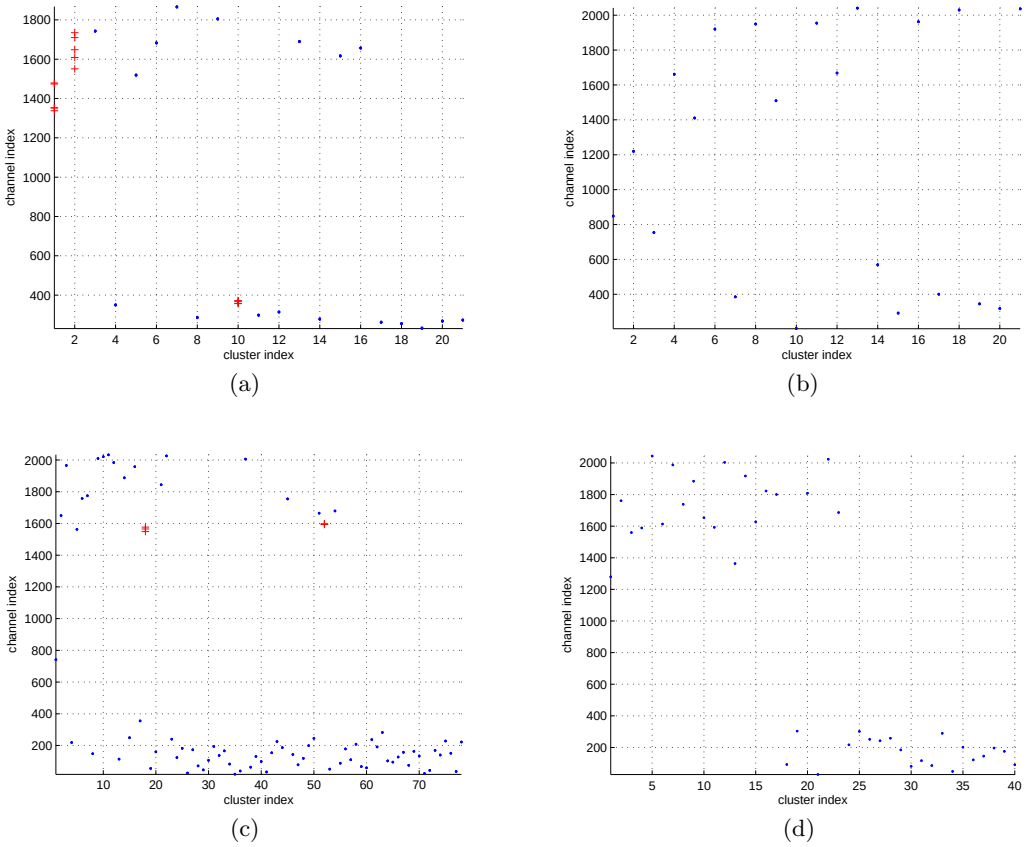


Figure 6.29: Channel index of the channels in each cluster, if the channels are adjacent to each other, then they are represented by '.'; otherwise they are represented by '+': (a) Cluster containing 5 channels; (b) Cluster containing 4 channels; (c) Cluster containing 3 channels; (d) Cluster containing 2 channels.

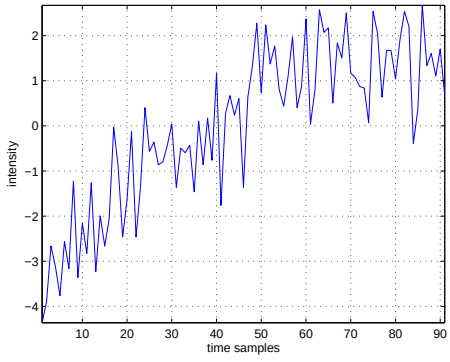


Figure 6.30: Strongest one-channel cluster

Number of channels contained in each cluster	Number of clusters	Channel index
11	1	372,373,374,375,376,377,378,379,380,381,382
8	2	321,322,323,324,325,326,327,328
		336,337,338,339,340,341,342,343
7	4	305,306,307,308,309,310,311
		1620,1621,1622,1623,1624,1625,1626
		1671,1672,1673,1674,1675,1676,1677
		329,330,331,332,333,334,335
6	2	1748,1749,1750,1751,1752,1753
		210,211,212,213,214,215
5	21	Fig. 6.29 (a)
4	21	Fig. 6.29 (b)
3	78	Fig. 6.29 (c)
2	40	Fig. 6.29 (d)

Table 6.3: Channel index of the channels in each of the sub-clusters and Fig. 6.29 shows the case when the 'number of clusters' is too big.

6.6.5 Further Analysis of the Single-Channel Clusters

According to Table 6.2, there are 91 single-channel clusters in total, containing one channel in each cluster. The strongest one-channel cluster is shown in Fig. 6.30, where one can see that there is a pattern in the signal, but it has been corrupted by noise. When the signal to noise ratio (SNR) is very low, it is very difficult to match signal patterns as they are swamped by the noise signals, which are uncorrelated. This could explain why there are so many single-channel clusters existing in our data. Thus, this suggests that an analysis of how noise impacts the MSC results is needed.

6.6.6 Effect of Noise on MSC

The issue of noise on OES signals was discussed in Section 3.7 and a method of estimating the signal to noise ratio (SNR) defined. In fact, there exists a strong relation between SNR and correlation coefficient (Pearson product-moment). Consider two mean-centered signals, denoted by x and y respectively. Let $\widetilde{x}$ and $\widetilde{y}$ be the two signals

corrupted by independent noise n_1 and n_2:

$$\widetilde{x} = x + n_1,\ \widetilde{y} = y + n_2, \tag{6.21}$$

where $E(n_1) = E(n_2) = 0$ and $E(n_1^2) = E(n_2^2) = \sigma^2$. Then, it follows that

$$\frac{corr(\widetilde{x}, \widetilde{y})}{corr(x, y)} = \sqrt{\frac{SNR_x}{1 + SNR_x} \frac{SNR_y}{1 + SNR_y}}. \tag{6.22}$$

A complete derivation is given in the Appendix A.2. If the two signals are identical (*i.e.* $x = y$ and $corr(x, y) = 1$), then Eq. A.7 reduces to

$$corr(\widetilde{x}_1, \widetilde{x}_2) = \frac{SNR_x}{1 + SNR_x}, \tag{6.23}$$

where $\widetilde{x}_1$ and $\widetilde{x}_2$ are used to differentiate the noise effect on the same signal.

Eq. 6.23 shows that even if two signals are perfectly correlated (identical), the presence of noise can make them appear less correlated. This has a significant impact on the performance of the MSC algorithm. For example if the similarity threshold is set to 0.9, then identical channels will be misclassified as belonging to different clusters if their $SNR < 9$. Thus, an important pre-processing step for MSC is to remove channels where the

$$SNR < \frac{\xi}{1 - \xi}. \tag{6.24}$$

Applying this step to IDS1 leads to 691 OES channels being discarded, leaving 1354 for further analysis.

6.6.7 Clustering on the Filtered OES Data

Applying max separation clustering to the IDS1Filt data set shows that the 1354 OES channels can be divided into 8 clusters, as shown in Fig. 6.31. Here, Cluster zero is used to represent all the discarded low SNR channels. A detailed channel distribution in each cluster is shown in Table 6.4, where one can see that there are two major clusters, each containing more than 200 OES channels. A detailed power distribution for the two major clusters is shown in Fig. 6.32.

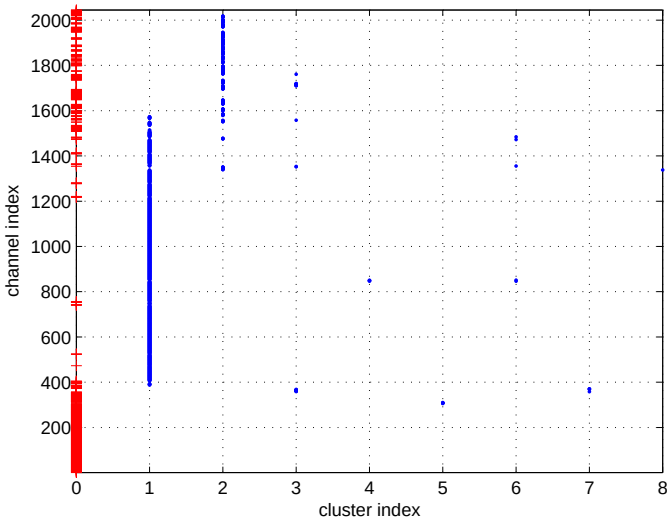


Figure 6.31: Clustering on the preprocessed OES data

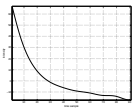
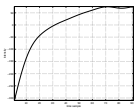
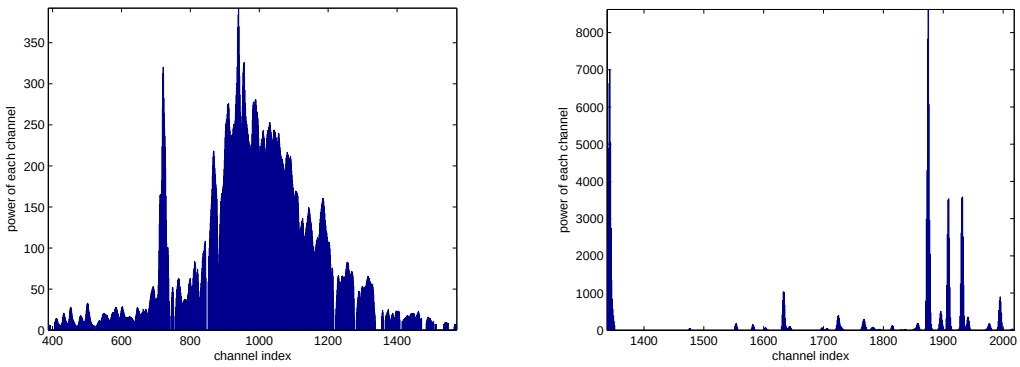
CI	Number of	Channel distribution	Channel index	RPC
1	1061	388-393,406-472,474-522,526-739,744-752, 757-845,852-1216,1222-1276,1282-1337,1356-1360, 1366-1408,1415-1471,1485-1507,1513-1514,1536-1548, 1566-1573,	940 	4.08-392.17
2	249	1339-1351,1475-1478,1551-1557,1579-1586,1600-1608, 1629-1647,1695-1709,1721-1734,1762-1773,1777-1797, 1812-1818,1828-1831,1834-1839,1849-1862,1870-1882, 1891-1915,1923-1946,1970-1982,1989-2002,2012-2018,	1875 	4.3-8622
3	25	359-368,1352-1353,1558,1710-1720,1761	1715	8.55-545.15
4	4	847-850,	849	16.91-23.97
5	5	306-310,	308	4.26-7.39
6	5	846,851,1355,1472,1484,	851	6.69-40
7	4	358,369-371,	371	4.88-9.7
8	1	1338	1338	23.16

Table 6.4: Channel distribution in each of the 8 clusters (CI = cluster index; RPC = range of power changes for the intra-cluster channels).



(a) (b)

Figure 6.32: Channel power: (a) Channels in Cluster 1; (b) Channels in Cluster 2.

Visualization of the two major clusters is provided in Fig. 6.33 and Fig. 6.34, respectively. In each figure, the three-dimensional display of the intra-cluster channel distributions, two-dimensional display of the standardized channels and mean and standard deviation analysis of the intensity changes of the standardized channels are provided in sequence. The mean and standard deviation analysis provides an indication of the similarity of the intra-cluster channels.

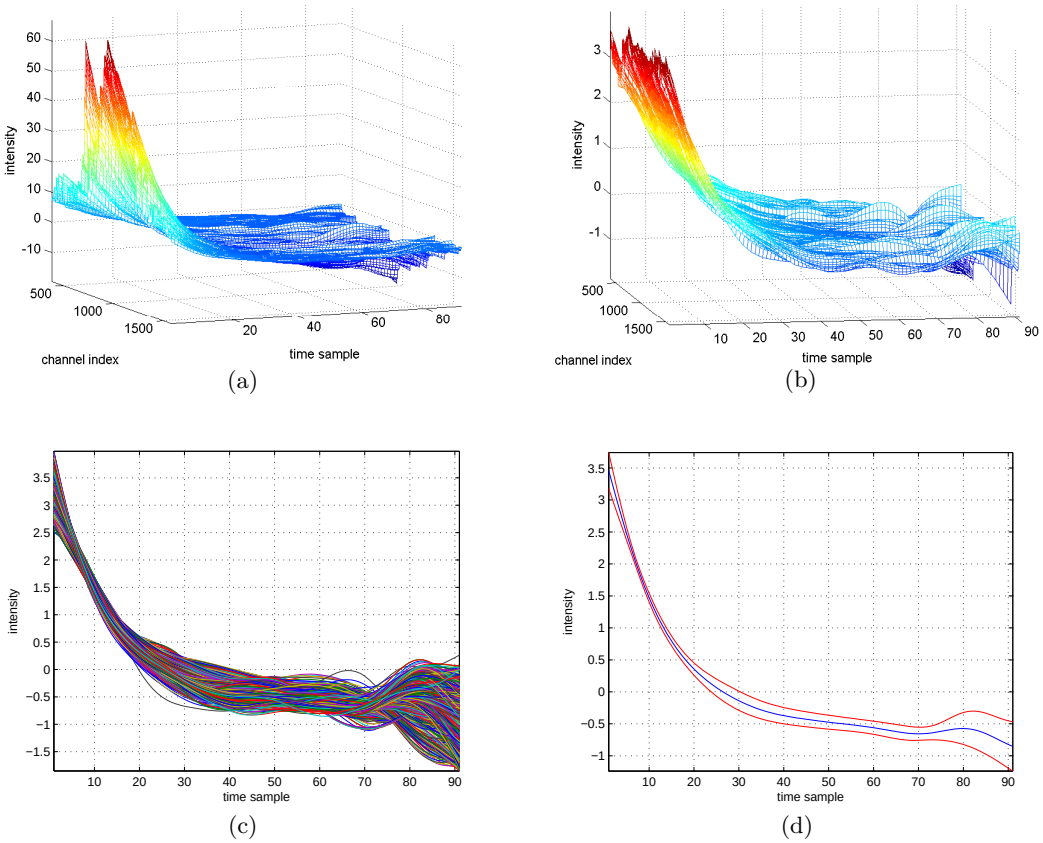


Figure 6.33: Data distribution in Cluster 1: (a) 3D display of Cluster 1 channels; (b) 3D display of the standardized channels in Cluster 1; (c) 2D display of the standardized Cluster 1 channels; (d) Mean and standard deviation of the standardized channels in Cluster 1.

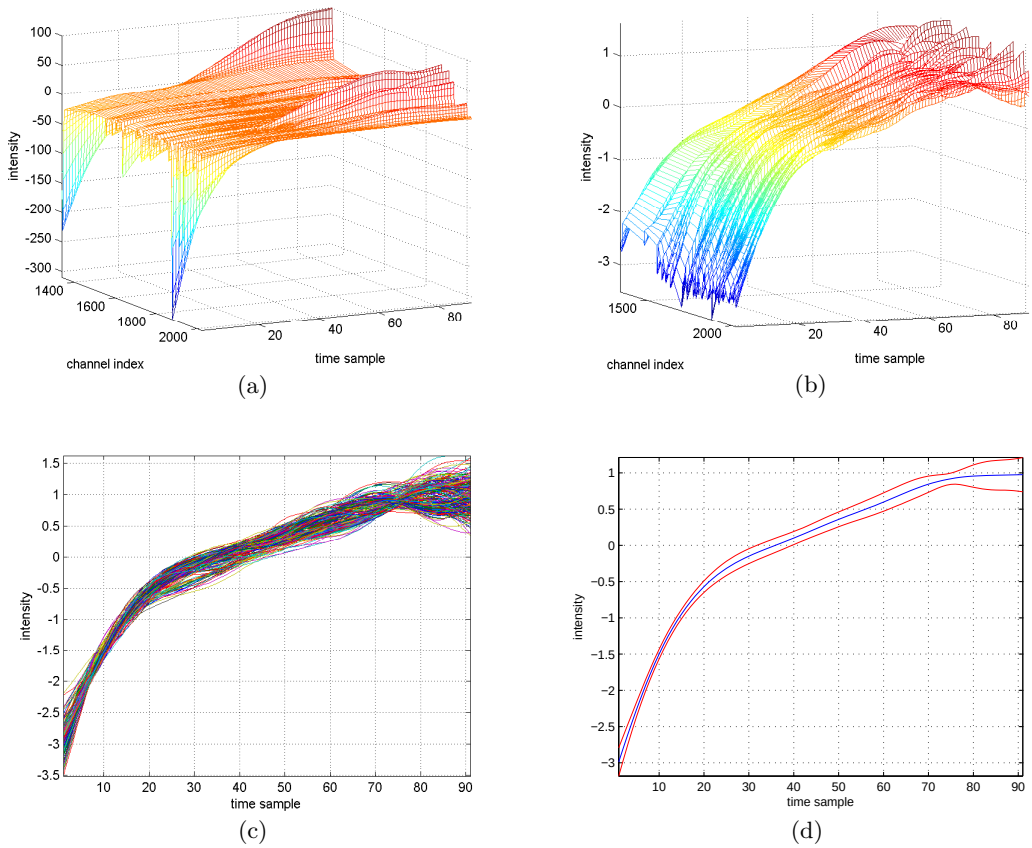


Figure 6.34: Data distribution in Cluster 2: (a) 3D display of Cluster 2 channels; (b) 3D display of the standardized channels in Cluster 2; (c) 2D display of the standardized Cluster 2 channels; (d) Mean and standard deviation of the standardized channels in Cluster 2.

The maxoids for all eight clusters are shown in Fig. 6.35, where both the mean-centered data and filtered data are shown. As one can see, the maxoids representing each cluster can capture quite different patterns in the data, which reflects the design of the algorithm.

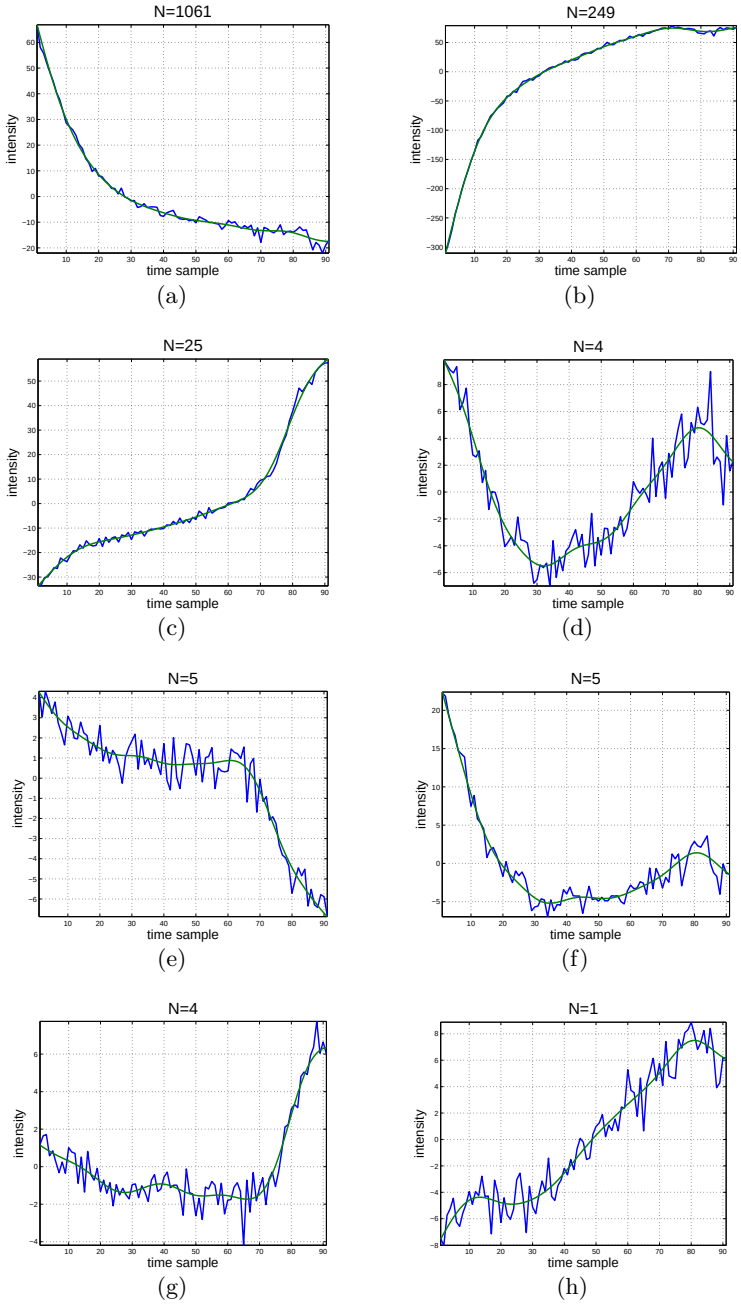


Figure 6.35: Intensity changes of the maxoids for all eight clusters (rough line: raw channel data with mean removed; smooth line: filtered channel): (a) Maxoid 1; (b) Maxoid 2; (c) Maxoid 3; (d) Maxoid 4; (e) Maxoid 5; (f) Maxoid 6; (g) Maxoid 7; (h) Maxoid 8.

6.7 Discussion and Conclusions

K-means, SOM and QT, three of the most powerful and widely used non-hierarchical clustering methods have been introduced in this chapter. With the aid of simulated data sets, the characteristics and properties of each method have been discussed. The results of applying K-means, SOM and QT on the raw and filtered OES benchmark data sets show the insufficiency of these methods in distinguishing and summarizing the patterns in OES data.

In this chapter, a new clustering algorithm, Max Separation Clustering (MSC), has been developed. MSC does not require a *priori* specification of the number of clusters and is not subject to inter-run variability. The application of MSC to clustering of OES data sets has been explored in detail. The results confirm that MSC is able to extract and summarise the different patterns contained in OES data and that the newly proposed maxoid in MSC is an effective representation of the patterns in each cluster. Another contribution in this chapter is the analysis of the noise effect on MSC. This analysis highlights the relationship between the similarity threshold and SNR and the need to omit low SNR signals to improve MSC performance. With low SNR signal removal, the number of clusters obtained by MSC is greatly reduced, leading to more effective summarization of the dominant patterns in the data.

Chapter 7

Hierarchical Clustering

As discussed in Chapter 6, there are two basic ways of searching for clusters, categorized as hierarchical and nonhierarchical clustering. Nonhierarchical clustering has been covered in Chapter 6 and a novel MSC algorithm has been developed. This chapter focuses on the hierarchical clustering approach and develops a custom single linkage hierarchical clustering (SLHC) implementation for OES data analysis.

Hierarchical clustering accomplishes classification by a series of merging (for agglomerative hierarchical clustering) or divisions (for divisive hierarchical clustering) of the clusters. In agglomerative methods, the clustering starts by assigning every single object as a separate cluster. Then at each step the two most similar clusters are merged until all objects are merged into a single cluster. For this reason, agglomerative methods are sometimes referred to as bottom-up methods. In divisive methods the clustering starts with all objects assigned to a single cluster. Then at each step a cluster is split in two. Divisive methods proceed in the direction opposite to agglomerative methods and thus, are sometimes referred to as top-down methods. Since agglomerative methods are the most widely discussed in literature and used in computer packages, and divisive methods can generally be viewed as agglomerative methods in reverse [68], further discussion of hierarchical clustering in this chapter is focused on agglomerative methods.

The chapter is organized as follows. First, an overview of the hierarchical clustering approaches is given. This is followed by a description of the customized SLHC algo-

rithm. Experimental results illustrating the operation of SLHC on both simulated and OES data sets are presented and compared with the results obtained by MSC. Then, a novel method for determining the number of clusters is developed. This is followed by an algorithm evaluation and comparison with other cluster number selection methods for SDS1 and IDS1 data sets.

7.1 An Overview of Hierarchical Clustering

Different hierarchical clustering approaches arise as a result of different choices of dissimilarity measure and methods for linking clusters. An introduction to dissimilarity measures and linkage methods is given in this section, with an emphasis on discussing and comparing the resulting algorithm properties.

7.1.1 Dissimilarity Measures

Dissimilarity measures, as the name implies, measure the dissimilarity/distance between objects. Equivalently, one can also use similarity measures or so-called proximity measures [76] to quantify the similarity between objects.

A number of dissimilarity measures have been proposed and by a more generalized definition, these measures can be divided into distance measures and correlation measures. A list of dissimilarity measures is given in Table 7.1, where the first four can be regarded as distance measures and the rest as correlation measures. A good discussion of some of these measures is given in [37]. In Table 7.1, x_{ik} and x_{jk} denote the k^{th} observation value of the m-dimensional objects i and j, respectively, and w_k denotes the weighting for the k^{th} observation.

Among distance measures, Euclidean distance, defined as

$$dist(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||_2, \tag{7.1}$$

is possibly the most widely used. The Euclidean distance (also known as the L_2 norm) can be interpreted as the physical distance between two objects. The city block distance (L_1 norm) was originally used to measure the distance between city blocks when

Measure	$dis(\mathbf{x}_i, \mathbf{x}_j)$
Weighted Euclidean distance	$(\sum_{k=1}^{m} w_k^2(x_{ik} - x_{jk})^2)^{\frac{1}{2}}$
Weighted city block distance	$\sum_{k=1}^{m} w_k\|x_{ik} - x_{jk}\|$
Weighted Minkowski distance	$(\sum_{k=1}^{m} w_k^p\|x_{ik} - x_{jk}\|^p)^{\frac{1}{p}}$ $(p \geq 1)$
Canberra distance	$\begin{cases} 0 & \text{for } x_{ik} = x_{jk} = 0 \\ \sum_{k=1}^{m} \|x_{ik} - x_{jk}\|/(\|x_{ik}\| + \|x_{jk}\|) & \text{for } x_{ik} \neq 0 \text{ or } x_{jk} \neq 0 \end{cases}$
Pearson's correlation based measure	$(1 - corr(\mathbf{x}_i, \mathbf{x}_j))/2$ where $corr(\cdot)$ denotes the Pearson's correlation function.
Cosine function	$cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j}{\|\|\mathbf{x}_i\|\|_2\|\|\mathbf{x}_j\|\|_2}$
Jackknife correlation	$1 - min\{\phi_{ij}^1, \ldots, \phi_{ij}^l, \ldots, \phi_{ij}^p\}$ where ϕ_{ij}^l denotes the Jackknife correlation coefficient with the l^{th} observation left out.
Spearman's correlation	$1 - 6\sum_{k=1}^{m} d_k^2/(m(m^2-1))$ where $d_k = r_{ik} - r_{jk}$ with r_{ik} denoting the ranking order of x_{ik} in the series of x_{ik} for $k = 1, \cdots, m$.

Table 7.1: A list of dissimilarity measures ($dis(\mathbf{x}_i, \mathbf{x}_j)$ denotes the dissimilarity between $\mathbf{x}_i$ and $\mathbf{x}_j$.)

following the street layout. It is also referred to as taxicab distance [96], rectilinear distance [8] and Manhattan distance [99]. Both the Euclidean ($p = 2$) and city block ($p = 1$) distances are in fact special cases of the Minkowski distance (L_p norm $= ||\mathbf{x} - \mathbf{y}||_p = (\sum_i |x_i - y_i|^p)^{\frac{1}{p}}$).

The Canberra distance [97] is designed to measure the sum of an array of scaled differences between coordinates of a pair of objects. The Canberra distance is very sensitive to small changes when both coordinates are near zero. According to [37], when the object coordinates are binary, the Canberra distance can better reflect the dissimilarity between objects than other dissimilarity measures.

Despite the fact that distance measures are more often employed in clustering as dissimilarity measures, correlation measures are also widely used [153, 23, 191].

Pearson's correlation is a typical correlation measure that aims at exploring the linear

relationship between two objects. By definition, the correlation coefficient can change only in the interval $[-1\ 1]$, where the value 1 represents the strongest possible positive relationship between two objects and the value -1 the strongest possible negative or reverse relationship. The effectiveness of using Pearson's correlation as a dissimilarity measure has been proven in [75, 156, 157, 158] for gene expression data analysis.

One drawback of Pearson's correlation is that the correlation between two objects can be dominated by outliers in the object observations [58]. Solving the problem led to the development of so-called Jackknife correlation [58]. Jackknife correlation can be regarded as a leave-one-out Pearson's correlation coefficient, that is, it is computed for the data with one observation at a time omitted. Thus there are m values computed for a m-dimensional object. The minimum value is then selected as the Jackknife correlation coefficient between two objects. By doing so, the Jackknife correlation avoids the effect of single outliers. However, it is computationally expensive and is rarely used.

Another drawback of Pearson's correlation coefficient is that it assumes an approximately Gaussian distribution of the observations of an object, and may not be robust for non-Gaussian distribution [189]. Addressing this problem, Charles Spearman proposed the Spearman's rank correlation [149]. By definition, the observation values of an object are replaced by the ranking orders. For example, if x_{ik} is the third highest value in x_{iq} for $1 \leq q \leq m$, then r_{ik} equals to 3. Spearman's correlation does not make any assumptions about the data distribution, but the ordering of values cannot represent the complete information contained in the observations. Jiang *et at* [76] confirmed that Spearman's correlation is a worse performer on average than Pearson's correlation in measuring the relationship between objects.

Correlation measures are often used in situations where the clustering is based on object profiles rather than object amplitudes/scales. Similar situations can also be measured using the cosine function [191]. The smaller the value of the cosine function the less similar the two objects are.

In fact, Euclidean distance, cosine function and Pearson's correlation are intrinsically

equivalent under certain conditions. If the data is mean centered,

$$cos(\mathbf{x}_i, \mathbf{x}_j) = corr(\mathbf{x}_i, \mathbf{x}_j). \tag{7.2}$$

i.e., the cosine function equals to the Pearson's correlation. If the data is standardized (mean centered and scaled to unit variance), the relationship between Pearson's correlation coefficient and the Euclidean distance can be expressed as [76]

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{2m(1 - corr(\mathbf{x}_i, \mathbf{x}_j))}, \tag{7.3}$$

where $dist(\cdot)$ denotes the Euclidean distance and m denotes the object dimension. Note that to prove Eq. (7.3), the sample standard deviation is calculated using the biased format.

7.1.2 Different Linkage Methods

Dissimilarity measures are used to quantify the differences between objects. To merge the most similar clusters at each step, it is also vital to define the dissimilarity between clusters, which is achieved by defining the dissimilarity between objects from different clusters. In this way, the relationships between clusters can be identified and used to determine at each step which clusters should be linked and hence, decide the resulting cluster structures. Some typical linkage methods are discussed as follows.

Single linkage clustering [146], the most widely known agglomerative method and also known as the nearest neighbor method, measures the distance between clusters as the distance between their two closest member objects. The drawback of single linkage is that it tends to produce clusters with unbalanced sizes and ignores the cluster structure at each merging step.

Complete linkage, also known as the furthest neighbour method [148] calculates the distance between clusters as the distance between their two furthest member objects. As a result, complete linkage tends to produce clusters with equal diameters (maximum dissimilarity between intra-cluster objects).

Average linkage [147] calculates the averaged distance between all possible pairs of intra-cluster objects. As a result, it tends to join the clusters with small variances. The advantage of average linkage is that the objects contained in the same cluster are more likely to be similar.

Centroid linkage [147] calculates the inter-cluster distance as the distance between cluster centroids, while median linkage [50] calculates the inter-cluster distance as the distance between cluster medians. Centroid and median linkages are both robust to the cluster outliers.

A more complex linkage method is the so-called Ward method [166]. Ward distance defines the inter-cluster distance, $dist(G_i, G_j)$, as

$$dist(G_i, G_j) = \frac{N_i N_j}{N_i + N_j} dist(\mathbf{c}_i, \mathbf{c}_j) \tag{7.4}$$

where G_i and G_j denote the i^{th} and j^{th} clusters, respectively, $dist(\mathbf{c}_i, \mathbf{c}_j)$ denotes the Euclidean distance between cluster centroid $\mathbf{c}_i$ and $\mathbf{c}_j$ and N_i and N_j denote the cardinalities of G_i and G_j, respectively. Given G_i and G_j merged into a new cluster (G_{new}), the new cluster centroid $\mathbf{c}_{new}$ is defined as

$$\mathbf{c}_{new} = \frac{N_i \mathbf{c}_i + N_j \mathbf{c}_j}{N_i + N_j}. \tag{7.5}$$

Given the intra-cluster distance, $IC(G_i, \mathbf{c}_i)$, as defined in Eq. (6.2), then it can be shown that

$$IC(G_{new}, \mathbf{c}_{new}) = IC(G_i, \mathbf{c}_i) + IC(G_j, \mathbf{c}_j)) + dist(G_i, G_j). \tag{7.6}$$

According to Eq. (7.6), $IC(G_{new}, \mathbf{c}_{new})$ will be greater than either $IC(G_i, \mathbf{c}_i)$ or $IC(G_j, \mathbf{c}_j))$, since the terms on the right-hand side of Eq. (7.6) are all positive. Thus, using Ward's definition, the intra-cluster distance is a monotonically increasing function. One feature of Ward's method is that it tends to join small clusters, since small clusters in general, have small intra-cluster distances.

The flexible beta method, proposed by Lance and Williams [98], is often regarded as a generalized distance measure in which each of the above measures are special cases.

Cluster Method	α_1	α_2	β	γ
Single linkage	$1/2$	$1/2$	0	$-1/2$
Complete linkage	$1/2$	$1/2$	0	$1/2$
Average linkage	N_i/N_i+N_j	N_j/N_i+N_j	0	0
Centroid	N_i/N_i+N_j	N_j/N_i+N_j	$-N_iN_j/(N_i+N_j)^2$	0
Median	$1/2$	$1/2$	$-1/4$	0
Ward's Method	$\frac{N_i+N_{new}}{N_i+N_j+N_{new}}$	$\frac{N_j+N_{new}}{N_i+N_j+N_{new}}$	$-\frac{N_{new}}{N_i+N_j+N_{new}}$	0

Table 7.2: The relationship between the flexible beta method and different linkages

Suppose the cluster G_i and G_j are merged into a new cluster G_{new} (cardinality N_{new}), the inter-cluster distance between G_{new} and G_k, $dist(G_k, G_{new})$, is defined as

$$\begin{aligned} dist(G_k, G_{new}) = \alpha_1 dist(G_k, G_i) + \alpha_2 dist(G_k, G_j) \\ + \beta dist(G_i, G_j) + \gamma |dist(G_k, G_i) - dist(G_k, G_j)|. \end{aligned} \tag{7.7}$$

Taking $\theta \equiv \{\alpha_1, \alpha_2, \beta, \gamma\}$ as the parameter set, different specifications of θ correspond to different linkage methods [130, 49]. Simplifying the definition given in Eq. (7.7), Lance and Williams [98] suggested a constraint condition:

$$\begin{aligned} &\alpha_1 = \alpha_2 = (1-\beta)/2, \\ &\gamma = 0, \\ &\beta < 1. \end{aligned} \tag{7.8}$$

Within this framework, all the linkage methods we have discussed before can be considered as special cases of the flexible beta method. More details of the parameter specifications are given in Table (7.2) [130].

7.2 Custom Single Linkage Hierarchial Algorithm

In this section, a custom single linkage hierarchical clustering (SLHC) algorithm is developed, aiming at providing detailed information on the relationship between clusters and intra-cluster objects that cannot be obtained using MSC. For example, given a cluster generated by MSC for a similarity threshold of 0.9, MSC does not provide information (at least not directly) on the correlation/similarity structure within the cluster. The ability to examine the data at different similarity levels is important,

because the patterns of objects contained in the same cluster can still vary from each other, although possibly in a subtle way. A detailed look at the distributions of these objects can help to discriminate these patterns and thus, help to translate the objects into corresponding chemicals or at least limit the selection of the possible chemicals.

Moreover, according to Johnson [77], exploring the natural clusters in the data by only one clustering method is in general not reliable, so several methods should be employed to verify the correctness of results. If different methods produce similar clustering results, then we can have more confidence in their validity.

Detailed Algorithm Description

Single linkage is one of the simplest agglomerative hierarchical clustering methods. By definition, single linkage merges the two closest clusters at each stage, where closeness is measured as the shortest distance between their member objects. This method matches our target of classifying similar OES channels into the same cluster. The other reason for using single linkage is that similarity can be measured based on the raw data, rather than on summary statistics, such as average, centroid, *etc.*, which makes estimation of the relationship between objects more reliable.

A complete description of the algorithm is as follows:

Step 1: Cluster initialization.

For a given set of objects $\mathbf{X}$ ($\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\},\ \mathbf{x}_i \in \mathbb{R}^{m\times 1}$), assign each individual object to a cluster:

$$G_i = \mathbf{x}_i,\ , i = 1, \ldots, n. \tag{7.9}$$

Set the cluster count $\widehat{n} = n$.

Step 2: Find the two clusters from the set of existing clusters that have the least dissimilarity.

Firstly, define the dissimilarity between object $\mathbf{x}_i$ and object $\mathbf{x}_j$ as

$$dis(\mathbf{x}_i, \mathbf{x}_j) = 1 - |corr(\mathbf{x}_i, \mathbf{x}_j)|, \tag{7.10}$$

where $corr(\cdot)$ denotes the Pearson's correlation coefficient. The dissimilarity between Cluster m and Cluster n is then defined as:

$$dis(G_i, G_j) = \min_{\mathbf{x}_i \in G_i, \mathbf{x}_j \in G_j} dis(\mathbf{x}_i, \mathbf{x}_j), \tag{7.11}$$

Step 3: Merge the two most similar clusters, G_p and G_q into one.

$$(G_p, G_q) = \arg\min_{G_i, G_j} dis(G_i, G_j), \tag{7.12}$$

Merging G_p and G_q results in a new cluster G_{new}, that is

$$G_{new} = \{G_p \cup G_q\} \tag{7.13}$$

and the cluster count becomes

$$\widehat{n} = \widehat{n} - 1. \tag{7.14}$$

Thus, all the objects belonging to G_p and G_q are assigned to G_{new} and G_p and G_q are deleted.

Step 4: Check the stop condition.

(a) If $\widehat{n} = 1$, the clustering is complete.

(b) Otherwise, return to Step 2.

In our version of single linkage, the similarity between objects is measured as the absolute value of the Pearson's correlation coefficient, which shows that even if two objects are strongly negatively correlated, the similarity between these two objects is still considered as high. This choice reflects the fact that the intensity decrease of one chemical in plasma etch process leads to a corresponding intensity increase of its by-products. In OES measurements, each channel (object) has a unique correspondence to an optical wavelength that can be taken as the fingerprint of a certain chemical. Therefore, even if the objects show reverse trends, in essence, they probably represent the fingerprints of the same chemicals.

SLHC clusters objects with reverse trends in the same group, but it is also important to know when such groupings have taken place. Thus to establish if positive and negative

objects are contained in the same group, additional checks have to be performed on each cluster. The algorithm for doing this can be expressed as:

---

for $i = 1$ to k **do**
  select $\mathbf{x}_i \in G_i$ at random, $G_i^+ \leftarrow \mathbf{x}_i$
  for $j = 1$ to N_i **do**
    if $corr(\mathbf{x}_i, \mathbf{x}_j) > 0$ **then**
      $G_i^+ \leftarrow \mathbf{x}_j$
    else
      $G_i^- \leftarrow \mathbf{x}_j$
    end if
  end for
end for

As shown in Fig. 7.1 (a), clustering is therefore a two-step process. The first step is to cluster the objects using SLHC, followed by separation of each cluster into positive/negative object trends. In contrast the MSC implementation described in Chapter 6 can achieve the separation directly (Fig. 7.1 (b)). However, with MSC further checks are needed to determine if any of the clusters are inversely related. Given the MSC clusters $G_1, \cdots, G_k$ with maxoids $\mathbf{m}_1, \cdots, \mathbf{m}_k$, the checking procedure can be expressed as:

---

for $i = 1$ to $k - 1$ **do**
  for $j = i + 1$ to k **do**
    if $corr(\mathbf{m}_i, \mathbf{m}_j) < -0.9$, G_i, G_j are inversely related **then**
      set $G_i^+ = G_i, G_i^- = G_j$
    end if
  end for
end for

In fact, MSC and SLHC can both be set up to either combine objects with reverse trends or allocate them to separate clusters. In the former, an additional step is required to split the resulting clusters into the positive/negative components, but in doing so the

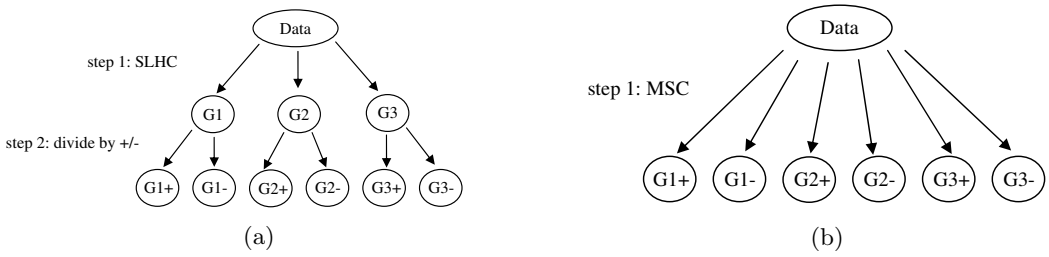


Figure 7.1: A comparison between SLHC and MSC for clustering of objects with reverse patterns: (a) SLHC; (b) MSC.

companion clusters are automatically obtained.

7.3 Experimental Results

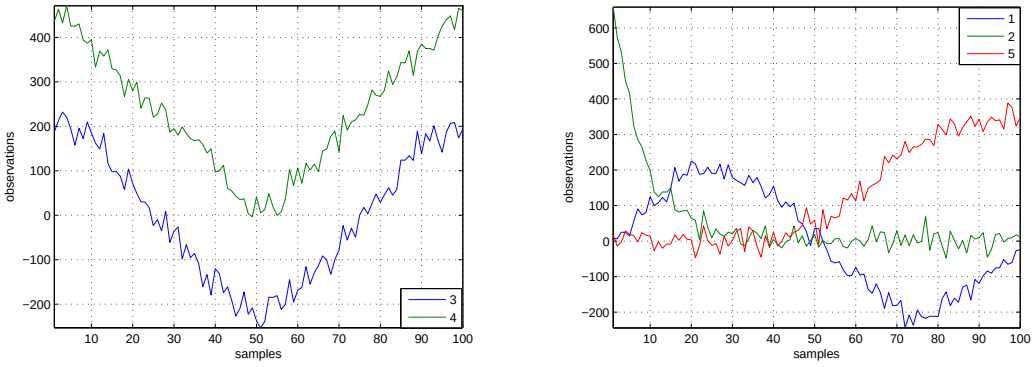
In this section, experimental results are presented for the SLHC algorithm applied to the SDS1, IDS1 and IDS1Filt data sets. The SDS1 allows the operation of the customized single linkage to be illustrated, aiming at providing further understanding of clustering on the IDS1 and IDS1Filt.

7.3.1 SLHC Applied to Simulated Data

To illustrate how SLHC works, a simplified version of SDS1 will be considered. This consists of only 5 objects one for each of the patterns in the data set. A plot of the 5 features numbered with 1 to 5 is shown in Fig. 7.2.
For the initial clustering, each individual object is assigned to a cluster, which gives the initial set of clusters, G, as

$$\mathsf{G}(0) = \{[1], [2], [3], [4], [5]\}. \tag{7.15}$$

Applying the dissimilarity function defined in Eq. 7.10, the dissimilarity matrix, a matrix recording the dissimilarity between clusters, can be computed as:



(a) (b)

Figure 7.2: Plot of the 5 features.

	1	2	3	4	5
1		0.8031	0.9827	0.9903	0.1748
2			0.5013	0.4996	0.6485
3				0.0230	0.7119
4					0.7068
5					

The smallest entry in the matrix is 0.0230, so a new cluster is generated by the merging of object 3 and 4. The G becomes

$$\mathsf{G}(1) = \{[1], [2], [3, 4], [5]\}. \tag{7.16}$$

Then, the dissimilarity between $[3, 4]$ and the other clusters changes to

$$\begin{aligned} d_{(34)1} &= \min[d_{13}, d_{14}] = 0.9827 \\ d_{(34)2} &= \min[d_{23}, d_{24}] = 0.4996 \\ d_{(34)5} &= \min[d_{35}, d_{45}] = 0.7068. \end{aligned} \tag{7.17}$$

As such, the dissimilarity matrix for G changes to

	1	2	$[3, 4]$	5
1		0.8031	0.9827	0.1748
2			0.4996	0.6485
$[3, 4]$				0.7068
5				

Then, the least dissimilarity is between [1] and [5], so [1] and [5] are joined together to form G and the updated set of clusters becomes

$$\mathsf{G}(2) = \{[1,5],[2],[3,4]\}. \tag{7.18}$$

Correspondingly, the dissimilarity matrix changes to

	$[1,5]$	2	$[3,4]$
$[1,5]$		0.6485	0.7068
2			0.4996
$[3,4]$			

Using the same principle, [3,4] and [2] are joined, which produces the following clustering:

$$\mathsf{G}(3) = \{[1,5],[2,3,4]\}, \tag{7.19}$$

and dissimilarity matrix

	$[1,5]$	$[2,3,4]$
$[1,5]$		0.6485
$[2,3,4]$		

In the final iteration, all the objects are joined together to form a single cluster.

$$\mathsf{G}(4) = \{[1,2,3,4,5]\}. \tag{7.20}$$

The complete clustering procedure can be illustrated using a dendrogram plot, or the so-called tree diagram, which is a widely used pictorial representation of the clustering procedure [37]. The dendrogram for the example considered above is shown in Fig. 7.3 (a). The number shown at the bottom of the figure represents the object indexes in the designed data set. The heights of the stems represent the dissimilarity at which clusters are joined. Taking object [1] and [5] as an example, since the dissimilarity between them equals to 0.1703, they are connected at this height in the dendrogram. A connection between the stems shows the generation of a new cluster. A complete dendrogram represents the whole clustering process and allows direct visualisation of the relationship between objects in the data set.

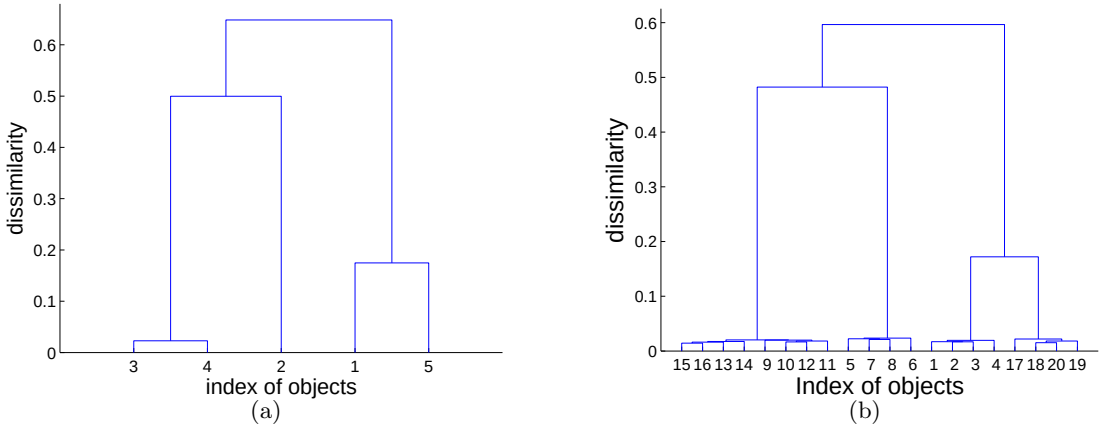
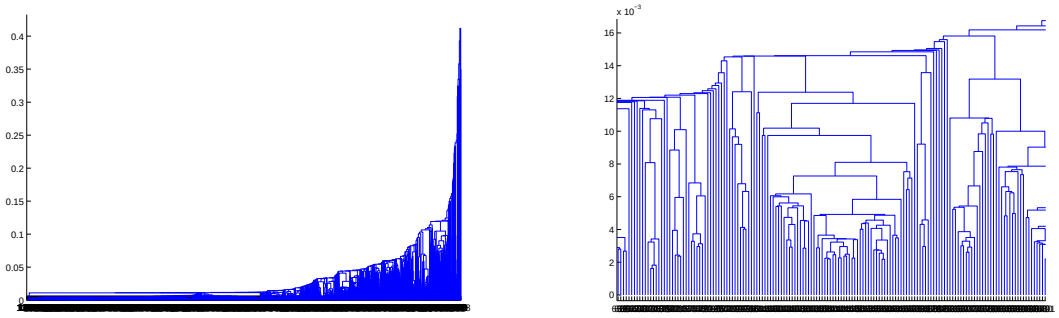


Figure 7.3: Dendrogram of applying SLHC on the simulated data: (a) 5 objects from the SDS1; (b) The whole SDS1.

The clustering results obtained when SLHC is applied to the full SDS1 data set is shown in Fig. 7.3 (b). The additional objects are simply noisy copies of the original 5 patterns (4 copies for each pattern) and this can be seen by the clusters formed.

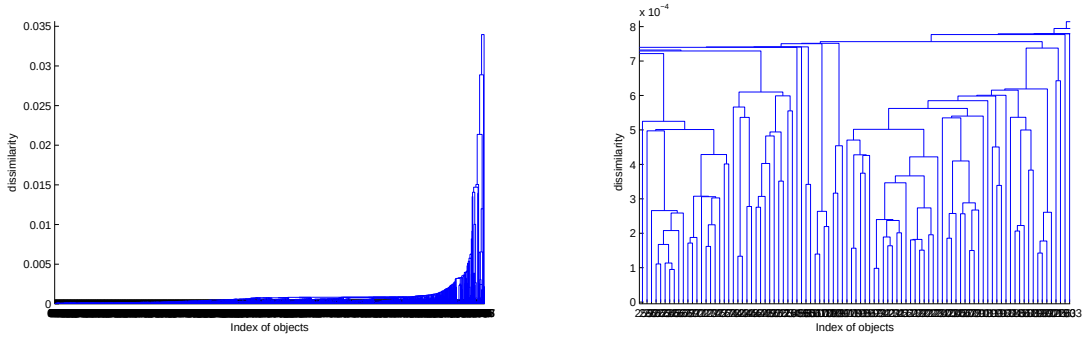
7.3.2 SLHC Applied to OES Data

The SLHC dendrogram for the IDS1 and IDS1Filt data sets are shown in Fig. 7.4 and Fig. 7.5, respectively. The x-axis in these figures is meant to show the indices of objects, but because of the large number of objects, it ends up as a solid black block lying under the figure. In addition, the cluster structure is lost as well. Even in the case of IDS1Filt, where the number of objects has been reduced due to the discarding of the noise signals, the number of residual objects (1354) is still too big to be presented visually. This is the main disadvantage of the dendrogram plot.



(a) (b)

Figure 7.4: Dendrogram of applying SLHC on the IDS1: (a) Full dendrogram; (b) Zoomed dendrogram around the middle.



(a) (b)

Figure 7.5: Dendrogram of applying SLHC on the IDS1Filt: (a) Full dendrogram; (b) Zoomed dendrogram around the middle.

To explore how SLHC extracts the patterns contained in the OES data, the IDS1Filt data set will be used as an example. Based on the MSC analysis in Chapter 6, the number of clusters is set to 8 for the SLHC. Fig. 7.6 presents all the objects contained in each of the eight clusters (data is standardised). As can be seen, two of the clusters, 6 and 8, have reverse patterns. The clusters obtained from splitting these reverse patterns are shown in Fig. 7.7.

Fig. 7.8 shows the range of intensity changes of the objects contained in different

Cluster index	Number of channels	Channel distribution	RSNR	RC
1	1	1338	17.47	1338
2	1	1353	16.85	1353
3	1	1355	9.8	1355
4	2	848-849	10.16,10.67	848
5	2	847,850	15.17, 16.2	847
6	15	306-310,1711-1720	10.38-295.57	1716
7	14	358-371	11.1-80.14	364
8	1318	388-393,406-472,474-522,526-739 744-752,757-846, 851-1216 1222-1276, 1282-1337, 1339-1352 1356-1360, 1366-1408, 1415-1472 1475-1478 , 1484-1507,1513-1514 1536-1548, 1551-1558,1566-1573 1579-1586, 1600-1608, 1629-1647 1695-1710, 1721-1734, 1761-1773 1777-1797, 1812-1818,1828-1831 1834-1839 , 1849-1862,1870-1882 1891-1915 , 1923-1946,1970-1982 1989-2002, 2012-2018	9.05-1575.2	1343

Table 7.3: Channel distribution in each of the 8 clusters obtained by SLHC (RSNR=range of SNR; RC=representative channel).

clusters (reverse patterns are adjusted to have the same trends) by the mean and standard deviation. The channel distributions of all clusters obtained by SLHC are listed in Table 7.3.

7.4 Comparison with Max Separation Clustering

Both the max separation clustering (MSC) and single linkage hierarchical clustering (SLHC) methods aim at grouping the objects with similar patterns into the same cluster, while keeping distinctive patterns in different clusters. Thus, clustering results obtained by these two methods on the same data set are expected to share some similarity. Otherwise, the obtained clustering results by either method are unreliable.

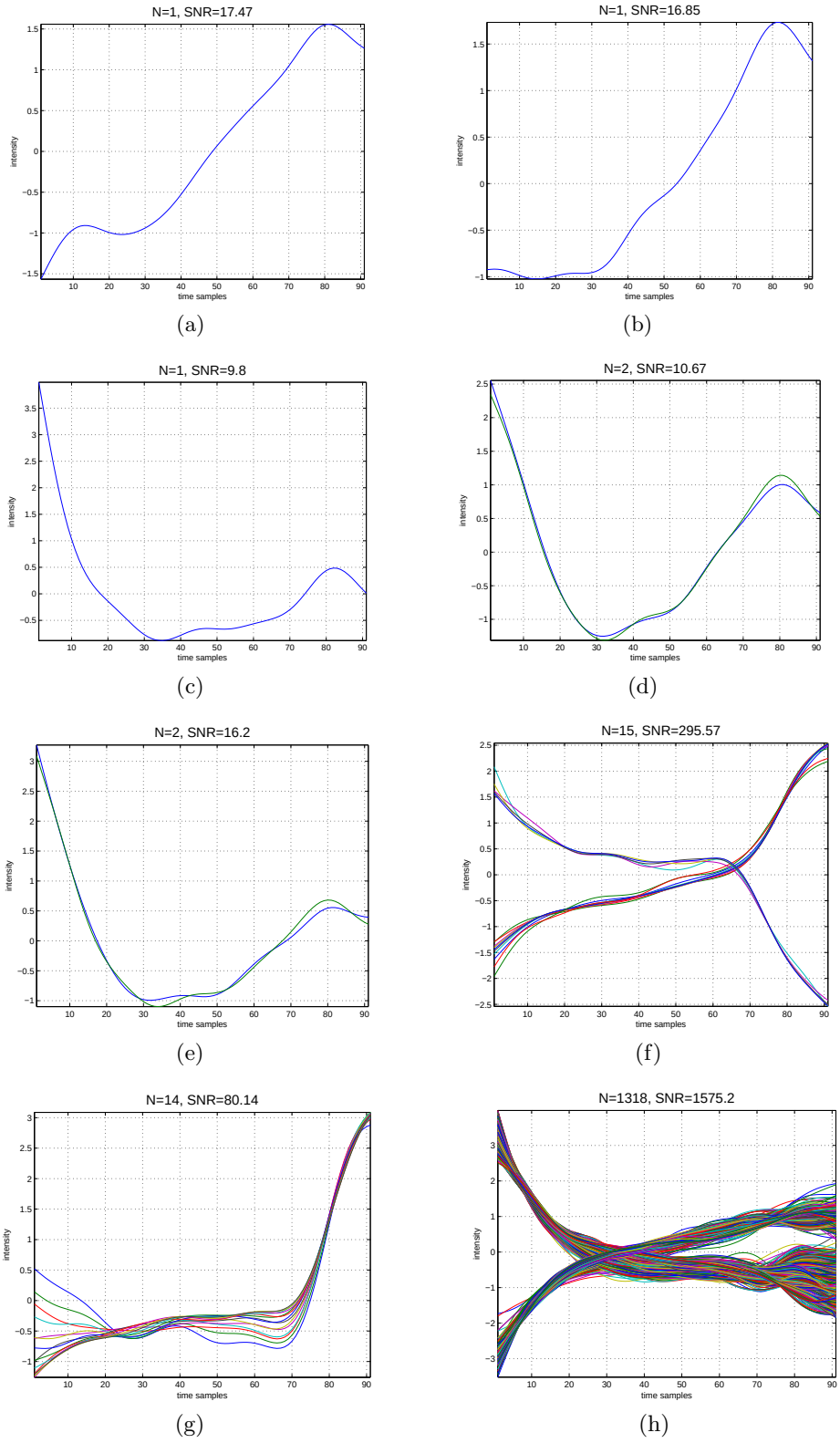


Figure 7.6: Objects contained in each of the eight OES clusters (N=the number of objects; SNR=SNR of the object having the strongest power): (a) Cluster 1; (b) Cluster 2; (c) Cluster 3; (d) Cluster 4; (e) Cluster 5; (f) Cluster 6; (g) Cluster 7; (h) Cluster 8.

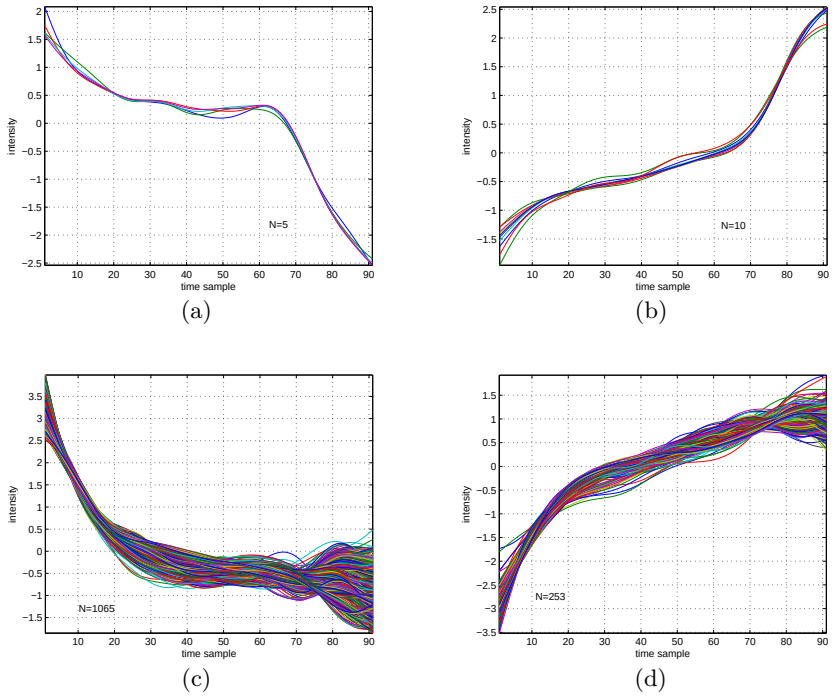


Figure 7.7: Splitting the intra-cluster objects with reverse trends: (a) Cluster 6^+; (b) Cluster 6^-; (c) Cluster 8^+; (d) Cluster 8^-

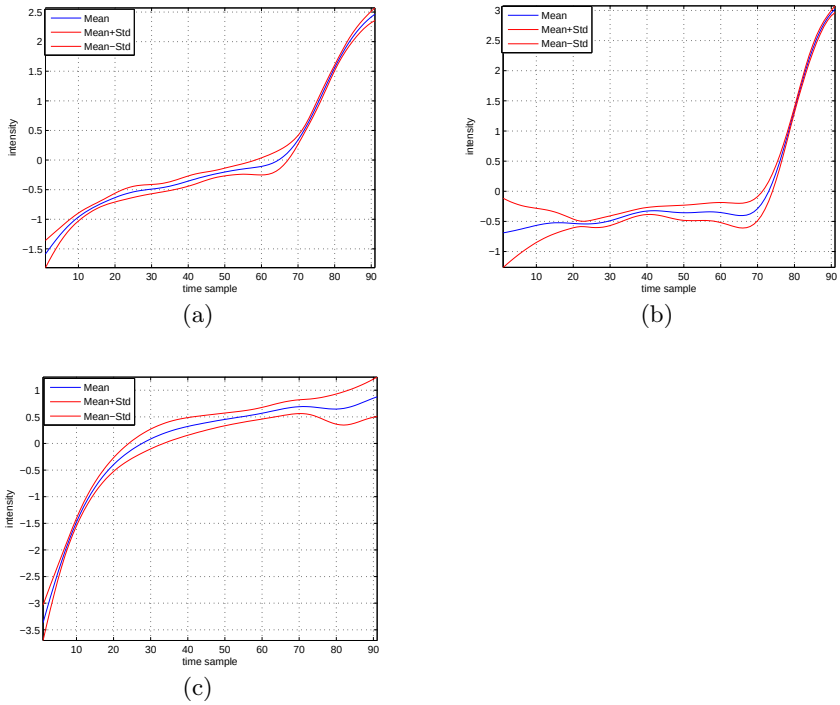


Figure 7.8: Summary of the range of intensity changes of the objects contained in different clusters using mean and standard deviation: (a) Cluster 6; (b) Cluster 7; (c) Cluster 8.

Method	SLHC									
Original cluster index	8^+	8^-	7	6^+	6^-	5	4	3	2	1
New cluster index	1	2	3	4	5	6	7	8	9	10
Cluster size	1065	253	14	10	5	2	2	1	1	1
Method	MSC									
Original cluster index	1	2	3	5	6	4	7	8		
New cluster index	1	2	3	4	5	6	7	8		
Cluster size	1061	249	25	5	5	4	4	1		

Table 7.4: Reordering the clusters according to cluster size for SLHC and MSC.

For ease of comparison of the cluster patterns across methods, the clusters obtained are reordered according to cluster size. The corresponding relationship between the original cluster index and the new index is listed in Table 7.4. To represent each cluster, the object with the highest power in each cluster is used. Fig. 7.9 and Fig. 7.10 show the patterns of the representative objects for all clusters obtained by the SLHC and MSC, respectively. As can be observed, the data patterns explored by both MSC and SLHC are quite similar, which confirms the effectiveness of the clustering results by either method. Note that Fig. 7.9 (c) and (d) have the similar patterns to that in Fig. 7.10 (c), and Fig. 7.9 (g) and (h) have the similar patterns to Fig. 7.10 (f). The only pattern that is captured by MSC which is not evident in the SLHC clusters is the one shown in Fig. 7.10 (g). This in fact has been included in cluster 7 (original index) obtained by SLHC (Fig. 7.6 (g)). The reason is that SLHC compares the similarity between unassigned object and any assigned objects contained in a cluster, giving more chance for the unassigned object to be included in this cluster. For completeness, the channel distribution for each cluster obtained by MSC and SLHC are provided in Table 7.5.

Recall that one of the motivations for exploring SLHC is that MSC cannot provide direct information on how similar clusters are to each other. This information is available with SLHC and can be easily visualized using a dendrogram plot, as shown in Fig. 7.11. Plot (a) is a zoomed version focusing on the top end of the dendrogram. The

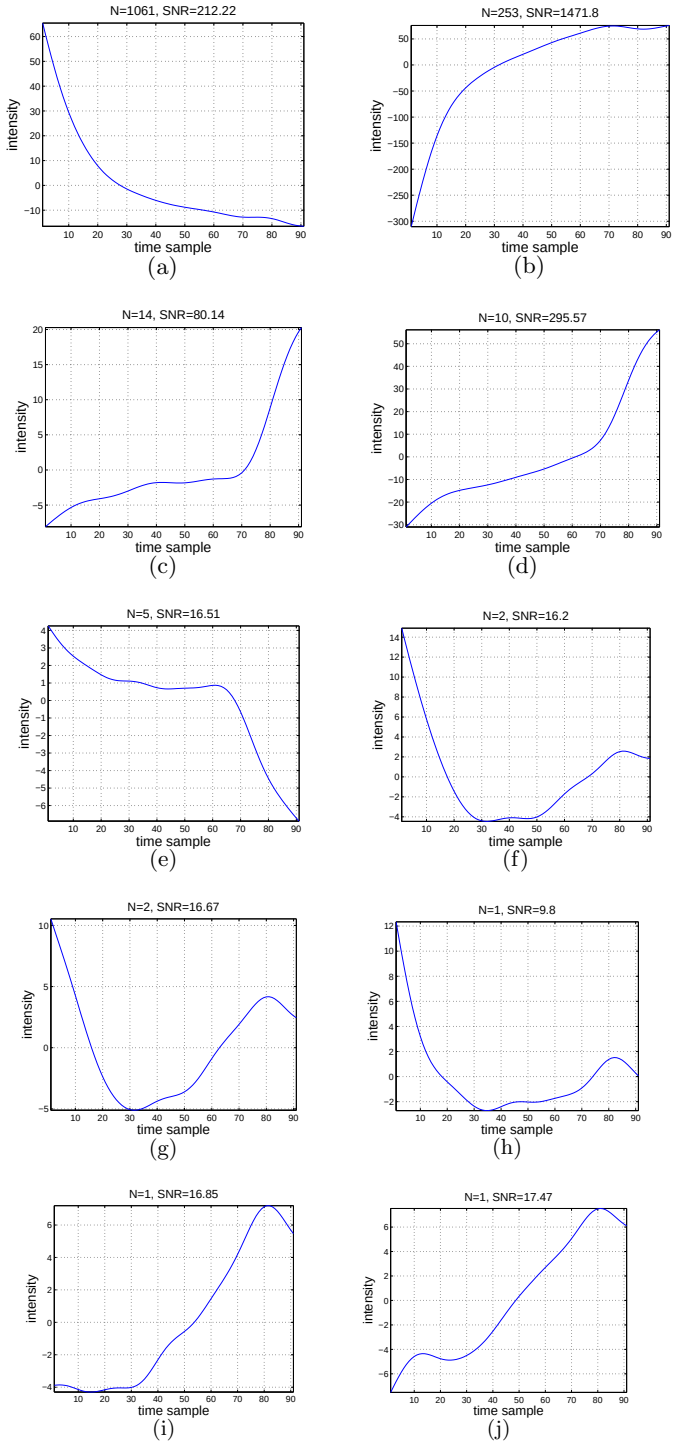


Figure 7.9: Object with the highest power in each of the 10 clusters, obtained by the SLHC (new cluster index is used): (a) Cluster 1; (b) Cluster 2; (c) Cluster 3; (d) Cluster 4; (e) Cluster 5; (f) Cluster 6; (g) Cluster 7; (h) Cluster 8; (i) Cluster 9; (j) Cluster 10.

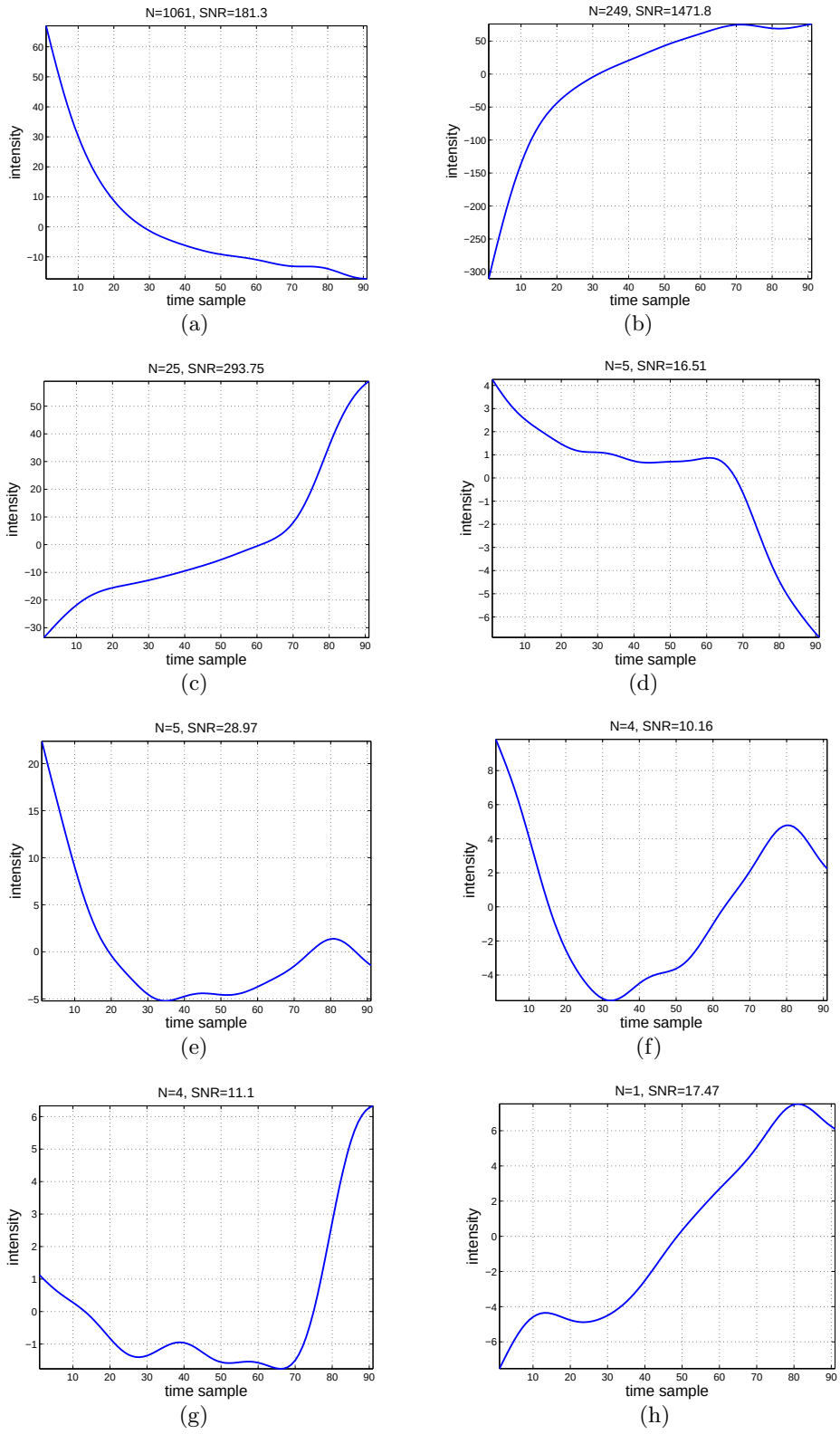


Figure 7.10: Cluster maxoid of each of the 8 clusters, obtained by the MSC (new cluster index is used): (a) Cluster 1; (b) Cluster 2; (c) Cluster 3; (d) Cluster 4; (e) Cluster 5; (f) Cluster 6; (g) Cluster 7; (h) Cluster 8.

Method	Cluster index	NC	Channel distribution
SLHC	1	1	1338
	2	1	1353
	3	1	1355
	4	2	848-849
	5	2	847,850
	6	15	306-310,1711-1720
	7	14	358-371
	8	1318	388-393,406-472,474-522,526-739,744-752,757-846,851-1216 1222-1276, 1282-1337,1339-1352,1356-1360,1366-1408,1415-1472 1475-1478, 1484-1507,1513-1514,1536-1548,1551-1558,1566-1573 1579-1586, 1600-1608,1629-1647,1695-1710,1721-1734,1761-1773 1777-1797, 1812-1818,1828-1831,1834-1839,1849-1862,1870-1882 1891-1915, 1923-1946,1970-1982,1989-2002,2012-2018
MSC	1	1061	388-393,406-472,474-522,526-739,744-752,757-845,852-1216 1222-1276,1282-1337,1356-1360,1366-1408,1415-1471,1485-1507 1513-1514,1536-1548,1566-1573
	2	249	1339-1351,1475-1478,1551-1557,1579-1586,1600-1608,1629-1647 1695-1709,1721-1734,1762-1773,1777-1797,1812-1818,1828-1831 1834-1839,1849-1862,1870-1882,1891-1915,1923-1946,1970-1982 1989-2002,2012-2018
	3	25	359-368,1352-1353,1558,1710-1720,1761
	4	4	847-850
	5	5	306-310
	6	5	846,851,1355,1472,1484
	7	4	358,369-371
	8	1	1338

Table 7.5: Channel distribution for each cluster obtained by MSC and SLHC (NC: the number of channels)

resulting merged clusters are highlighted by unique colors, as shown in plots (b), (c) and (d).

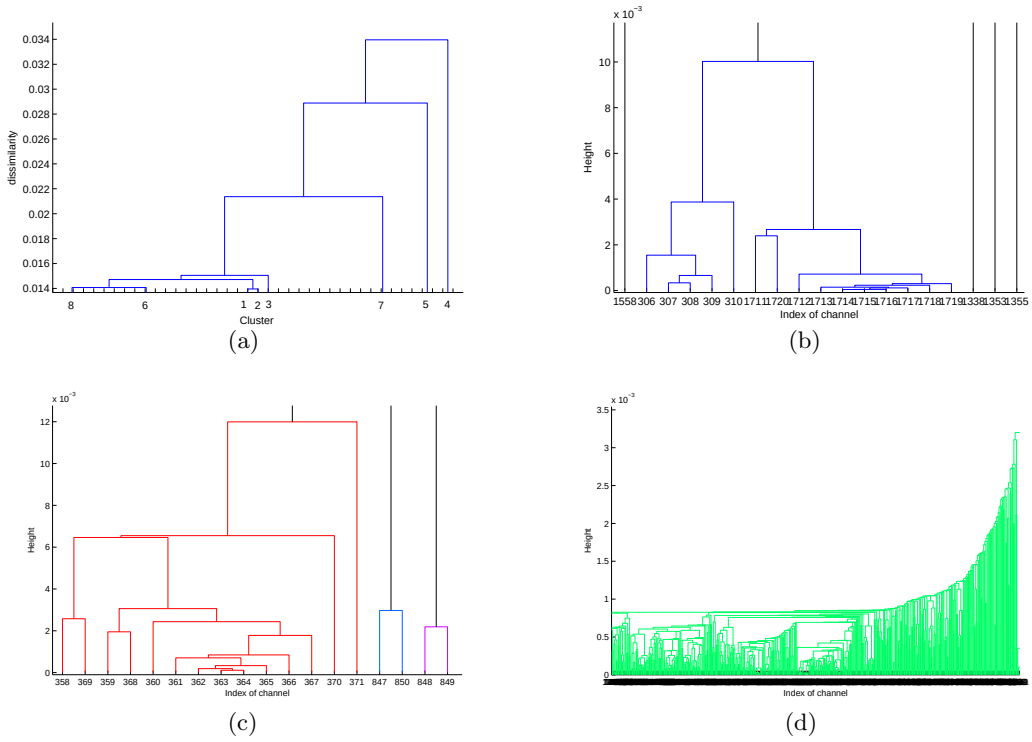


Figure 7.11: Zoomed version of dendrogram plot showing the inter-cluster relationship and the intra-cluster similarity between objects: (a) On the top end level showing the inter-cluster relationship; (b), (c) and (d) showing the dissimilarity level between intra-cluster objects for (cluster 1,2,3,6), (cluster 4,5,7) and (cluster 8), respectively.

Note that in Fig. 7.11 the object index is converted to the index of OES channels for all the sub-figures to allow comparison with the channel distribution shown in Table 7.5. According to these sub-figures, for the smaller clusters, one can easily see the channels in each cluster and the level of similarity between channels. Taking channel 307 as an example, according to Fig. 7.11 (b), the most related channel to channel 307 is channel 308. Channel 309 is the most related channel to both Channel 307 and 308, followed by channel 306 and channel 310. At a slightly higher level, it can be seen that the group of channels most related to the cluster 306-310 is the group 1711-1720. Hence, if we know that channel x represents an important wavelength, for a particular species, we

might be interested in identifying which other channels show similar behavior. Using SLHC, such information can be obtained. For the cluster that contains a large number of channels (Fig. 7.11 (d)), visualisation of patterns is more difficult, but by interactive zooming on the dendrogram the detailed relationships can be explored.

One issue that has yet to be discussed is how to determine the appropriate number of clusters when using SLHC, *i.e.* the appropriate cut-off level in the dendrogram. In the following section, a review of the existing methods for cluster-number selection is provided first and then a novel method for automatic selection of the number of clusters is presented.

7.5 Selecting the Number of Clusters

Selecting the number of clusters is an important consideration in hierarchical clustering. The reason is that hierarchical clustering itself does not generate a set of clusters. What hierarchical clustering can do is to provide an ordering of the relationship between objects, typically represented in the form of cluster trees, starting with each object in separate clusters and ending with all objects in a single cluster. Thus, determining the appropriate number of clusters (the clustering resolution) is a post-processing step. However, cluster-number selection methods, also referred to as stopping rules, are heuristic and *ad hoc* procedures, so one must be critical about using these rules.

Comprehensive experiments on estimation of the effectiveness of over 30 existing cluster-number selection measures were carried out by Milligan and Cooper [119] on a series of simulated data sets. The results show that one method can be better than another for certain data structures, but no single method outperforms the others across all data structures. Thus, there are no methods which are in general good at selecting the correct number of clusters. However, the study is still very valuable, as it identifies which methods are unreliable for certain types of data. In this sub-section, five typical methods are introduced, namely the Calinski-Harabasz index, the Duda and Hart Index, the Beal's F-type Index, the Index I and the silhouette index.

7.5.1 Calinski-Harabasz Index

The Calinski-Harabasz (CH) index, one of the best performers in Milligan and Cooper's study, is designed to measure the clustering performance as a function of the number of clusters (k). When the function, $CH(k)$, obtains its maximum, the optimum clustering resolution is achieved. $CH(k)$ is defined as [12]

$$CH(k) = \frac{B}{k-1} / \frac{W}{n-k}, \tag{7.21}$$

where B and W denote the inter-cluster dispersion and intra-cluster dispersion, respectively, that is

$$\begin{aligned} B &= \sum_{i=1}^{k} N_i (\mathbf{c}_i - \overline{\mathbf{x}})^{\mathsf{T}} (\mathbf{c}_i - \overline{\mathbf{x}}) \\ W &= \sum_{i=1}^{k} \sum_{l=1}^{N_i} (\mathbf{x}_l^i - \mathbf{c}_i)^{\mathsf{T}} (\mathbf{x}_l^i - \mathbf{c}_i), \end{aligned} \tag{7.22}$$

where $\mathbf{x}_l^i$ denotes the l^{th} object in cluster i, $\mathbf{c}_i$ is the centroid of cluster i, $\overline{\mathbf{x}}$ denotes the sample mean for all objects and N_i denotes the number of objects contained in cluster i. Given that the total number of objects is n, it follows that

$$n = \sum_{i=1}^{k} N_i. \tag{7.23}$$

In principle, the CH index tries to select a situation where the inter-cluster performance is most different from the intra-cluster performance. Similar ideas are also employed in the design of the Davies-Bouldin index [28], Dunn's Index [32] and the Xie-Beni index [179].

7.5.2 Duda and Hart Index

Duda and Hart proposed an index that can be used to judge if a cluster should be divided into two sub-clusters for divisive clustering. Let IC denote the clustering performance measured as the intra-cluster sum of squared distances between the objects and the centroid. Then the DH index, another best performer in Milligan and Cooper's study, is defined as [31] :

$$DH = \{1 - \frac{IC_2^2}{IC_1^2} - \frac{2}{\pi m}\} \{\frac{nm}{2[1 - 8/(\pi^2 m)]}\}^{1/2}, \tag{7.24}$$

where IC_1 denotes the clustering performance for one cluster, say cluster A, IC_2 denotes the summed clustering performance when cluster A is separated into cluster A1 and

cluster A2. m denotes the dimensionality of the object and n denotes the number of objects in cluster A. The null hypothesis of one cluster is rejected if the ratio exceeds a certain significance level, specified according to the standard normal distribution function. Whereas CH can be thought as an absolute or global performance measure computed for each value of k, DH is a relative or local measure which looks at the change in performance as the number of clusters change from k to $k+1$. The DH index employs a local criterion. As highlighted in [119], the DH index performs well in most cases, but it cannot provide an overall measure of the clustering performance as a function of the number of clusters. Thus, as a cluster-number selection method, the DH index has its limitation.

7.5.3 Beal's F-type Index

Another popular index measure is the so-called Beale's F-type index. Beale proposed using a pseudo F-distribution statistic to test whether the existing clustering G_2 (consisting of k_2 clusters) is better than clustering G_1 (containing k_1 clusters with $k_2 > k_1$). Beale's F-tpye index is defined as [4, 77]:

$$f = \frac{(w_1 - w_2)}{w_2} \frac{1}{[\frac{n-k_1}{n-k_2}][\frac{k_2}{k_1}]^{2/\rho} - 1}, \tag{7.25}$$

where ρ is set to 2, in general, n is the number of objects and

$$w_1 = \sum_{i=1}^{k_1} \sum_{l=1}^{N_i} (\mathbf{x}_l^i - \mathbf{c}_i)^{\mathsf{T}} (\mathbf{x}_l^i - \mathbf{c}_i), \tag{7.26}$$

where N_i denotes the cardinality of cluster i, $\mathbf{c}_i$ is the centroid of cluster i and $\mathbf{x}_l^i$ is the l^{th} object contained in cluster i. Similarly, w_2 is defined in the case where the number of clusters is k_2. Beale argued that when f exceeds a certain significance level, clustering G_1 is better than clustering G_2. Beale's F-type index tries to combine the local and global criteria into one index, but it does not define how to calculate the differences in clustering performance between two successive iterations, so it cannot be used to indicate the natural structures of clusters [70]. Moreover, the drawback of Beale's index, as well as the DH index is that the significance value needs to be specified for each individual experiment.

7.5.4 Index I

Index I [116] is a clustering resolution method developed recently that defines the appropriate number of clusters as the number which maximises the I-index, defined as

$$I(k) = (\frac{1}{k} \times \frac{TIC_1(\mathsf{G},\mathsf{C})}{TIC_k(\mathsf{G},\mathsf{C})} \times D_k)^2. \tag{7.27}$$

Here, k denotes the number of clusters, $TIC_1(\mathsf{G},\mathsf{C})$ is the $TIC(\mathsf{G},\mathsf{C})$ (defined in Eq. (6.1)) for the clustering where all objects are in a single cluster, and $TIC_k(\mathsf{G},\mathsf{C})$ is the $TIC(\mathsf{G},\mathsf{C})$ when there are k clusters. Scaling factor D_k is defined as

$$D_k = \max_{i,j=1}^{k} \|\mathbf{c}_i - \mathbf{c}_j\|_2, \tag{7.28}$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ are the centroid of cluster i and j, respectively. For a given data set, $TIC_1(\mathsf{G},\mathsf{C})$ is constant, hence $I(k)$ varies as a result of k, $TIC_k(\mathsf{G},\mathsf{C})$ and D_k competing and balancing with each other.

7.5.5 Silhouette Index

Silhouette index (SI) [81] is a widely used method for interpreting and validating the clusters in data. For object $\mathbf{x}_l^i$ (object l in cluster i), the average similarity of object $\mathbf{x}_l^i$ to all objects in cluster j, G_j, can be defined as

$$a_j(\mathbf{x}_l^i) = \frac{\sum_{\mathbf{x}\in G_j} corr(\mathbf{x}, \mathbf{x}_l^i)}{N_j}. \tag{7.29}$$

Let $a_w(\mathbf{x}_l^i)$ denote the average similarity between $\mathbf{x}_l^i$ and the other objects within its own cluster, G_i, and let $a_n(\mathbf{x}_l^i)$ denote the average similarity between $\mathbf{x}_l^i$ and the other objects contained in its nearest cluster, G_n, that is

$$w = i \text{ and } n = \arg\max_{j\neq i} a_j(\mathbf{x}_l^i). \tag{7.30}$$

The silhouette value for object $\mathbf{x}_l^i$ is then defined as

$$s(\mathbf{x}_l^i) = \frac{a_w(\mathbf{x}_l^i) - a_n(\mathbf{x}_l^i)}{\max\{a_n(\mathbf{x}_l^i), a_w(\mathbf{x}_l^i)\}}. \tag{7.31}$$

A value of $s(\mathbf{x}_l^i)$ close to 1 shows that object $\mathbf{x}_l^i$ is more similar to the objects contained in its own cluster than the objects contained in its nearest cluster, so the classification of $\mathbf{x}_l^i$ to G_i is justified. Accordingly, if $s(\mathbf{x}_l^i)$ is close to -1, object $\mathbf{x}_l^i$ is more similar

to the objects contained in G_n, so the classification is wrong. However, a value near 0 cannot be used to estimate whether the classification of $\mathbf{x}_l^i$ is right or wrong. This is the drawback of the silhouette index.

The silhouette value for cluster i is then defined as

$$S_i = \frac{1}{N_i} \sum_{l=1}^{N_i} s(\mathbf{x}_l^i), \tag{7.32}$$

where N_i is the cardinality of G_i and the overall or global clustering performance across all k clusters can be defined as [73]

$$GS_k = \frac{1}{k} \sum_{i=1}^{k} S_i. \tag{7.33}$$

Hence, GS_k is a measure of the clustering performance for a given number of clusters, and will be referred to as the silhouette index.

7.6 B-Index

Hierarchical clustering usually does not require any parameters to be specified before the clustering process is completed. However, to make the clustering meaningful, it is necessary to determine the appropriate level of separation *i.e.* to estimate the intrinsic number of clusters in the data. In the previous section, a number of approaches for estimating the appropriate clustering level have been presented. Here, a new B-index is proposed.

7.6.1 Theoretical Description

The index is designed to explore the differences in clustering performance between two successive iterations, while at the same time taking into account the clustering performance for the single cluster and all clusters. The index, referred to as the B-index, is defined as follows:

$$B(k) = |\frac{N_{k+1}^* - N_k^*}{N_k*} \times \frac{D}{D_k} \times \triangle D_k|, \tag{7.34}$$

where

$$D = \max_{\mathbf{x}_i, \mathbf{x}_j} dis(\mathbf{x}_i, \mathbf{x}_j), \tag{7.35}$$

$$D_k = \max_{\mathbf{x}_i, \mathbf{x}_j \in G_k^*} dis(\mathbf{x}_i, \mathbf{x}_j), \tag{7.36}$$

$$D_{k+1} = \max_{\mathbf{x}_i, \mathbf{x}_j \in G_{k+1}^*} dis(\mathbf{x}_i, \mathbf{x}_j), \tag{7.37}$$

$$\triangle D_k = \frac{w_k D_k - w_{k+1} D_{k+1}}{\max(w_k, w_{k+1})}, \tag{7.38}$$

$$w_k = std_{\mathbf{x}_i, \mathbf{x}_j \in G_k^*}(dis(\mathbf{x}_i, \mathbf{x}_j)), \tag{7.39}$$

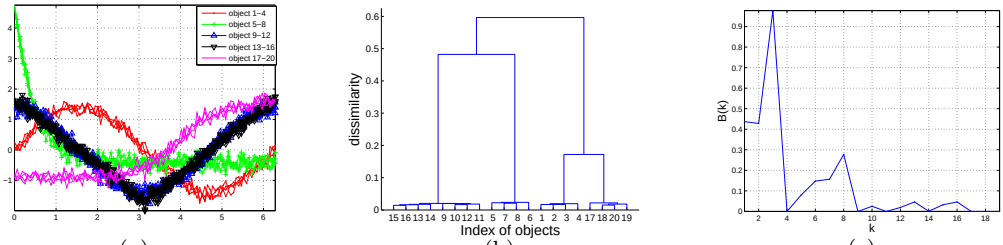
$$w_{k+1} = std_{\mathbf{x}_i, \mathbf{x}_j \in G_{k+1}^*}(dis(\mathbf{x}_i, \mathbf{x}_j)). \tag{7.40}$$

$std(\cdot)$ is the standard deviation function and $dis(\cdot)$ is the dissimilarity function, as defined in Eq. 7.10. G_k^* is the new cluster formed in the creation of G_k with k clusters, $\mathsf{G}_k = \{G_1, \cdots, G_k^*\}$ (*i.e.* where two of the clusters in G_{k+1} are combined to produce a single cluster, reducing the cluster count by 1). Similarly, G_{k+1}^* is the new cluster formed in the creation of G_{k+1} with $k+1$ clusters, $\mathsf{G}_{k+1} = \{G_1, \cdots, G_{k+1}^*\}$. In agglomerative hierarchical clustering, SLHC for example, G_{k+1} is the clustering prior to G_k. N_k^* and N_{k+1}^* denote the cardinality of G_k^* and G_{k+1}^*, respectively.

D measures the greatest dissimilarity between objects (global criterion). For a given data set, D is constant. D_k measures the intra-cluster dissimilarity in G_k (local criterion) and decreases with increasing k. $\triangle D_k$ measures the weighted difference of the clustering performance for the two clusters generated in the two successive iterations (between iterations). The basic idea is if the clusters generated in the two successive iterations are similar, the value of $\triangle D_k$ is small and hence, as a clustering performance measure, $B(k)$ is small. Otherwise, $B(k)$ is big. The clustering performance is weighted by the dispersion level of the intra-cluster objects (measured by w_k and w_{k+1}). This is included to deal with the case where outliers are merged into the newly generated cluster.

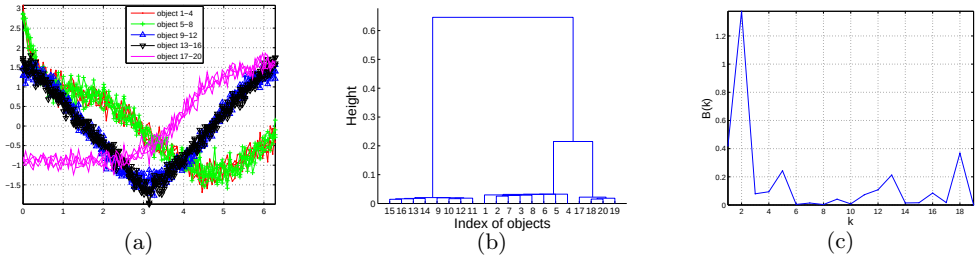
$B(k)$ measures the changes in $\frac{N_{k+1}^* - N_k^*}{N_k^*}$, $\frac{D}{D_k}$ and $\triangle D_k$. The maximum of $B(k)$ corresponds to a case where the two clusters with the most distinctive features are merged together, so the clustering should be stopped in the **previous** iteration. Thus the selected number of clusters, k^s, is given by

$$k^s = \arg\max_k B(k) + 1. \tag{7.41}$$



(a) (b) (c)

Figure 7.12: Selecting the number of clusters using the B-index for SDS1 ($\alpha = 1$): (a) Plot of SDS1; (b) Clustering Dendrogram; (c) B-index.



(a) (b) (c)

Figure 7.13: Selecting the number of clusters using the B-index for SDS1 ($\alpha = 0.5$): (a) Plot of SDS1; (b) Clustering Dendrogram; (c) B-index.

7.6.2 B-Index Applied to Simulated Data

The operation of the B-index will be illustrated using the SDS1 data set, which consists of 20 objects with 4 distinct patterns ($\alpha = 1$).

A plot of the standardised SDS1 data is shown in Fig. 7.12 (a). The SLHC dendrogram is shown in Fig. 7.12 (b), from which one can see that the 20 objects can be clustered into 4 groups. In the B-index curve (shown in Fig. 7.12 (c)), when $k = 3$, $B(k)$ achieves its maximum, which indicates that the merged two clusters have the most distinctive patterns. Hence, the clustering should be stopped in the previous iteration, *i.e.* when number of clusters is 4 (Eq. (7.41)).

Using the SDS1 data set with $\alpha = 0.5$, the distinct number of patterns is reduced to

3. The plot of the standardised data and the corresponding clustering dendrogram are shown in Fig. 7.13 (a) and (b), respectively. The changes in $B(k)$ as a function of the number of clusters (k) is shown in Fig. 7.13 (c), where one can see that when $k = 2$, $B(k)$ reaches the maximum. Hence, the number of clusters obtained is 3 (Eq. (7.41)).

These two case studies show that the B-index is effective at revealing the intrinsic structures in the data. When dealing with data containing a large number of objects, for example, greater than 100, methods such as direct observation of the distributions of objects in the dendrogram are not feasible. It is in these situations that the use of the B-index is advantageous.

7.6.3 B-Index Applied to OES Data

IDS1 and IDS1Filt, the two OES data sets, used as benchmarks, each contains more than 1000 objects. Direct observation of the dendrogram is not feasible as a means of determining a solution for the number of clusters. Hence, the B-index is employed. Fig. 7.14 shows the result of using the B-index on IDS1. When $k = 86$, $B(k)$ reaches its maximum, so the number of clusters is predicted as 87. Fig. 7.15 (a) shows the channel distribution in each cluster. A more detailed description of each cluster is presented in Table 7.6. The average power analysis shows that weak signals are contained in most of the clusters. Visual observation of each cluster shows that the weak signals are more likely to correspond to noise signals. As an example, the channels contained in cluster 7 (with 10 channels) are presented in Fig. 7.15 (b).

The application of the B-index to IDS1Filt is shown in Fig. 7.14 (c). Fig. 7.14 (d) shows the interval $k \in [1, 20]$ of Fig. 7.14 (c) in more detail. It can be seen that when $k = 6$, $B(k)$ reaches its maximum, suggesting 7 as the intrinsic number of clusters. As discussed in sub-section 7.3.2 where the number of clusters was set to 8, the result obtained by B-index simply corresponds to the result obtained when cluster 1 and 2 (Fig. 7.6 and Table 7.3) are merged into a single cluster. As such, the detailed comparison with MSC is given in section 7.4. However, rather than achieving data clustering in one step as MSC does, SLHC has to be used jointly with the B-index,

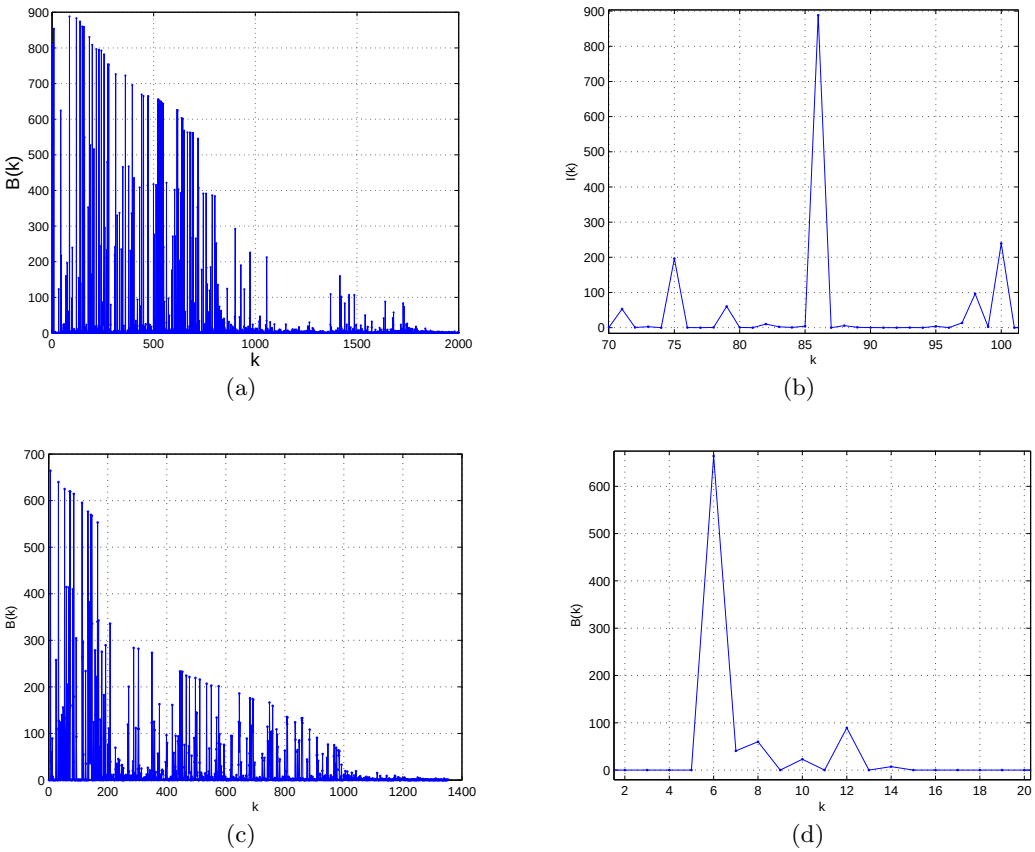
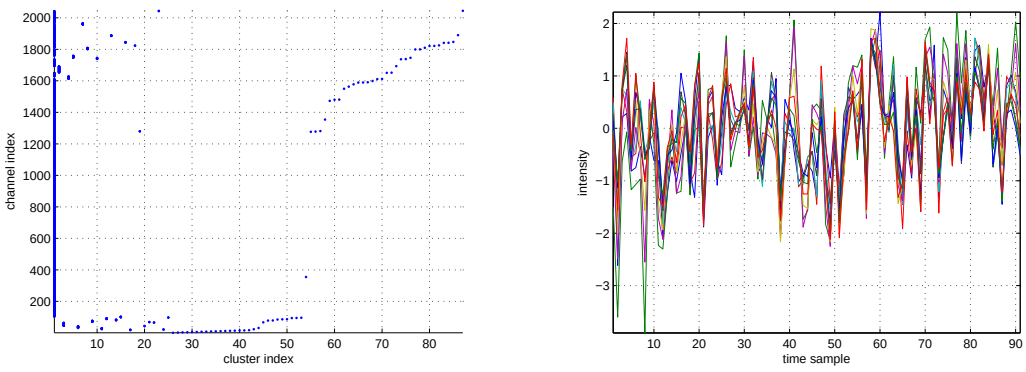


Figure 7.14: B-index applied to OES data: (a) $B(k)$ for the IDS1 (b) Zoomed version of Fig. 7.14 (a); (c) $B(k)$ for the IDS1Filt; (d) Zoomed version of Fig. 7.14 (c).

with SLHC providing the linkage between objects and the B-index determines the appropriate level of separation.

7.7 Comparison Between B-index and Other Cluster-Number Selection Methods

In this section, the B-index is compared with a number of competing indices, namely the I-index, CH index and silhouette index. The I-index was selected for comparison as it provides the inspiration for the design of the B-index, while the CH and silhouette indices were selected as they are the best performers and most widely used clustering performance measures [119, 81].



(a) (b)

Figure 7.15: (a) Channel distribution in each cluster; (b) Channels in cluster 7.

7.7.1 Performance On Simulated Data

Here the performance of the selected indices for the SDS1 data set with $\alpha = 1$ and $\alpha = 0.5$ is presented. Fig. 7.16 shows a comparison of the indices on the SDS1 data set with $\alpha = 1$. The changes in $I(k)$ as a function of the number of clusters is shown in Fig. 7.16 (b), where the maximum value of $I(k)$ is obtained when $k = 19$. In fact, it is not surprising, because according to Eq. (7.27), $TIC_k(G, C)$ decreases as k increases, while D_k, which measures the inter-cluster dissimilarity, may have very small values, especially when each object is assigned to a single cluster. As such, the ratio between D_k and $TIC_k(G, C)$ could be very big. There is a local maximum at $k = 4$, and this point indicates the correct cluster number. Fig. 7.16 (c) shows the changes in $CH(k)$. The maximum value is obtained when $k = 4$, so the number of clusters can be correctly predicted by the CH index. For the silhouette index (shown in Fig. 7.16 (d)), the maximum value of S_k is achieved when $k = 4$, so the silhouette index is effective in this case.

All these four methods are applied to the SDS1 set with $\alpha = 0.5$. The global maximum for the B-index and CH index is obtained at $k = 3$ (Fig. 7.17). The local maximum is achieved at $k = 3$, for the I-index and for the silhouette index, the first local maximum is achieved at $k = 3$. Comparison between different computing indices shows that the B-index, I-index, CH index and silhouette index are all effective in determining the number of clusters for the simple cases, where data contains distinct patterns.

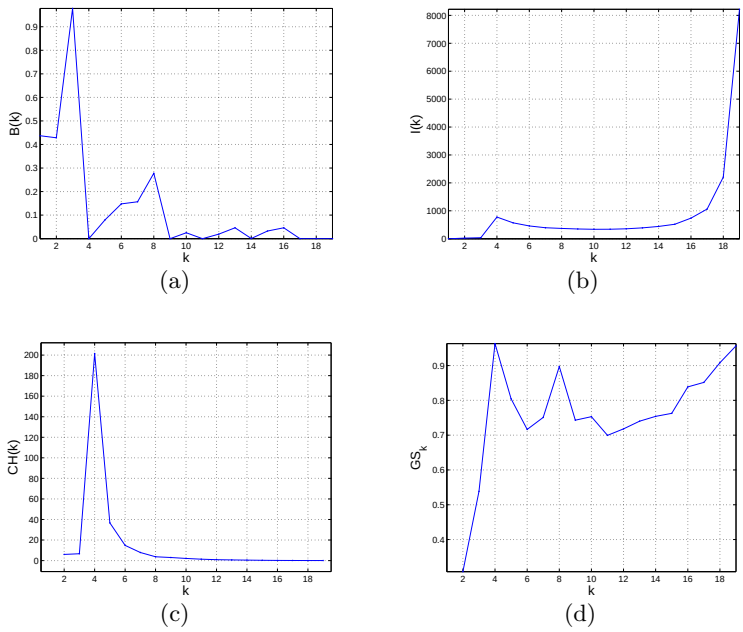


Figure 7.16: Comparison between the different clustering performance measures for SDS1 ($\alpha = 1$): (a) B-index; (b) I-Index; (c) CH index; (d) Silhouette index.

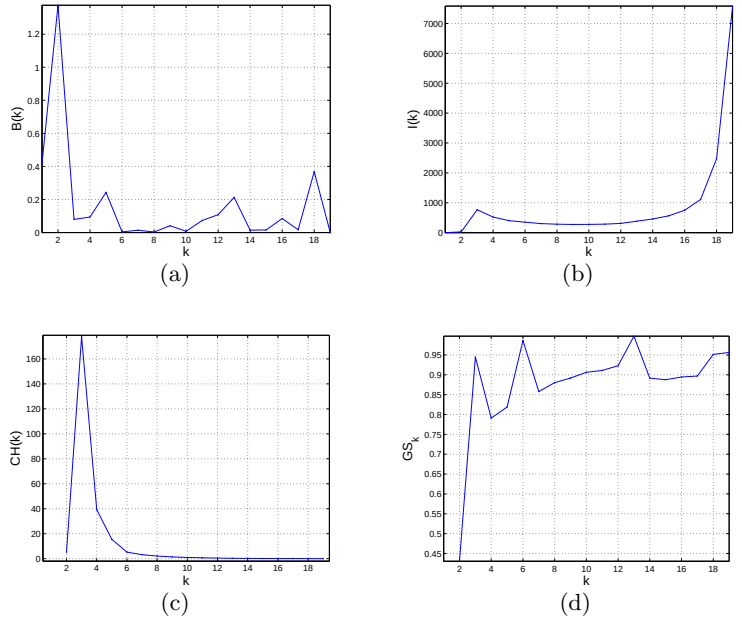


Figure 7.17: Comparison between the different clustering performance measures for SDS1 ($\alpha = 0.5$): (a) B-index; (b) I-Index; (c) CH index; (d) Silhouette index.

No. of channels contained in each cluster	Average power of the channels in each cluster	No. of clusters
1798	124.32	1
40	1.9	1
20	0.26	1
16	2.03	1
13	3.16	1
11	0.22	1
10	0.87	1
9	0.9	1
8	0.28	1
7	2.18, 0.23	2
6	0.22	1
5	0.23, 0.25, 0.93	3
4	0.66	1
3	0.21	1
2	0.45, 3.86, 0.22, 0.19, 0.18, 2.13, 0.18, 0.17	8
1	0.156-6.6372	62

Table 7.6: Simple statistics of the channel distribution in each cluster

7.7.2 Performance On OES Data

The results of applying the B-index, I-index, CH index and silhouette index to the SLHC clusters for IDS1 and IDS1Filt are shown in Fig. 7.18 and Fig. 7.19, respectively. The number of clusters selected by these four indexes for the IDS1 data set are 87, 2, 60 and 19, respectively. For IDS1Filt, the corresponding selected number of clusters are 7, 2, 4 and 6. Although the number of clusters selected by different computing indices varies for the IDS1 data set, the result is relatively consistent for the IDS1Filt data set. According to the results given by MSC, the IDS1 and IDS1Filt data sets can be divided into 264 and 8 clusters, respectively. This confirms the effectiveness of the B-index at detecting the appropriate clustering resolution, especially for high dimensional OES data sets.

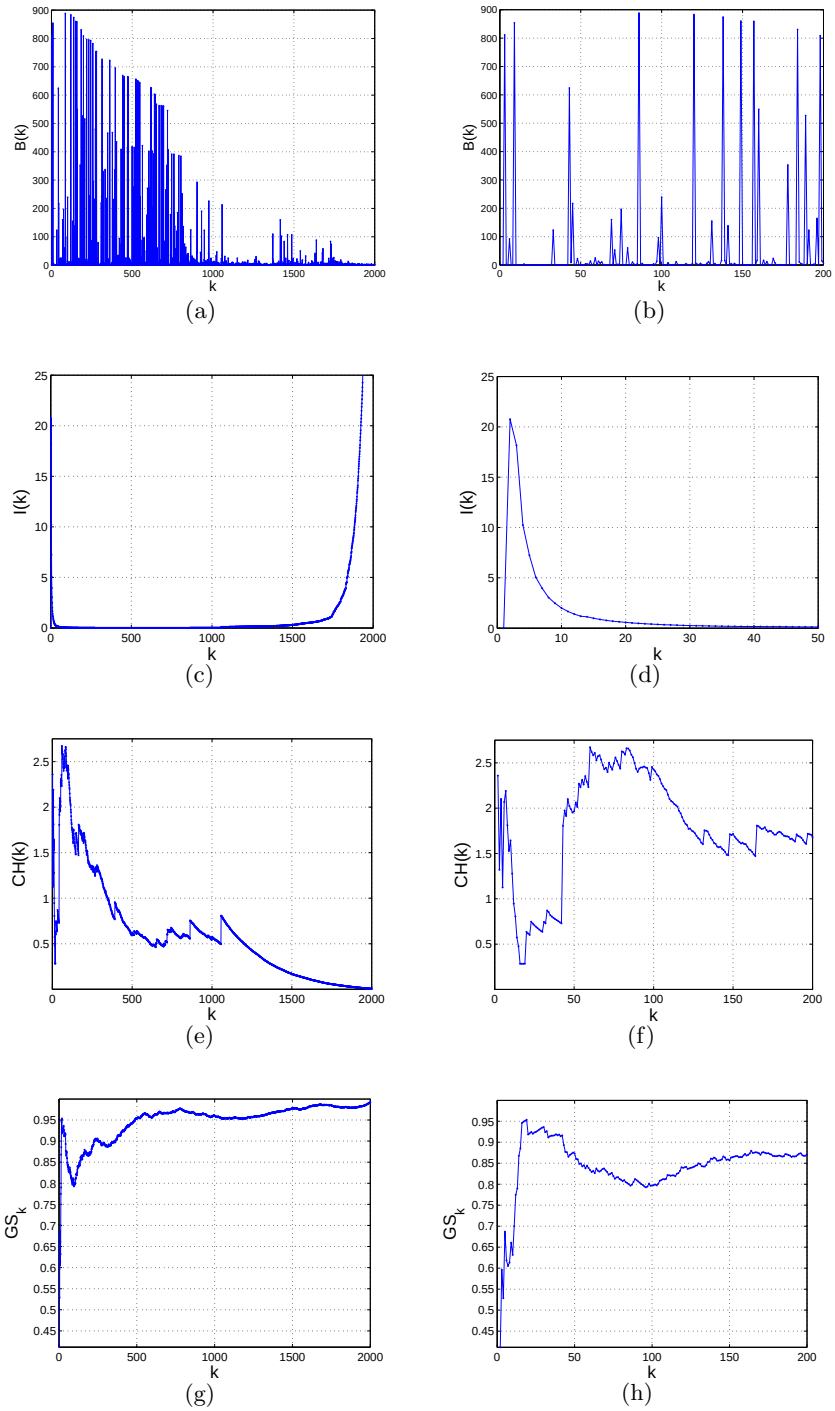


Figure 7.18: Comparison between the different clustering performance measures for IDS1: (a) B-index; (b) Interval $k \in [0, 200]$ of (a); (c) Index I; (d) Interval $k \in [0, 50]$ of (c); (e) CH index; (f) Interval $k \in [0, 200]$ of (e); (g) Silhouette index; (h) Interval $k \in [0, 200]$ of (g).

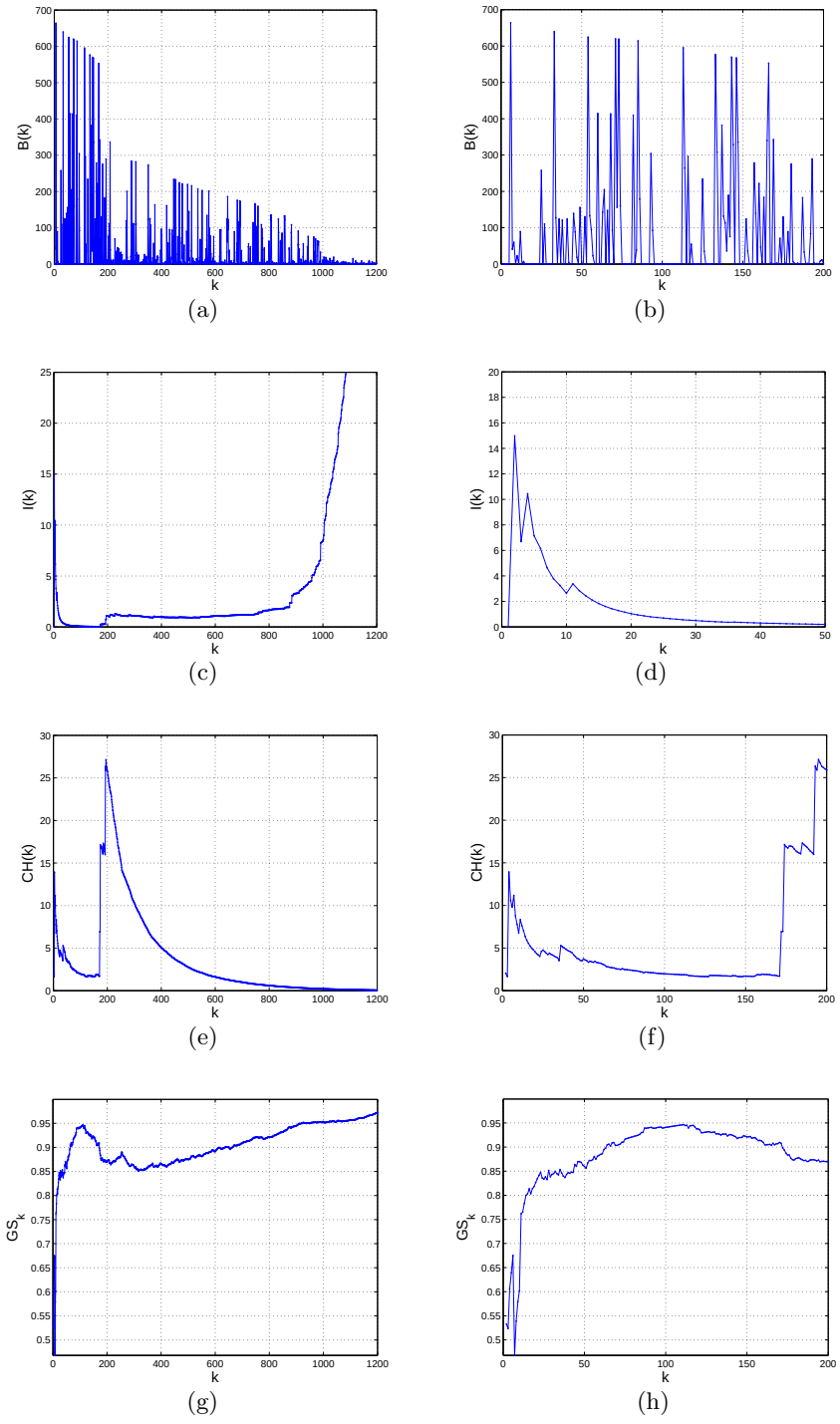


Figure 7.19: Comparison between the different clustering performance measures for IDS1Filt: (a) B-index; (b) Interval $k \in [0, 200]$ of (a); (c) Index I; (d) Interval $k \in [0, 50]$ of (c); (e) CH index; (f) Interval $k \in [0, 200]$ of (e); (g) Silhouette index; (h) Interval $k \in [0, 200]$ of (g).

7.8 Discussion and Conclusions

In this chapter a review of the typical hierarchical clustering approaches and the methods for selecting the number of clusters has been presented. To deal with OES data, a custom SLHC algorithm has been implemented. The main advantage of the SLHC is that it can be used to disclose the relationship between clusters and between intra-cluster objects. For example, when one channel (object) is known to be important, the related channels from the same species can be captured by SLHC and the inter-channel relationship can be visualised in dendrogram in order of similarity level. This is a useful feature for practical use.

To estimate the appropriate number of clusters, a novel B-index measure has been developed. The ability of the B-index to predict the number of clusters has been demonstrated with the aid of a simulated data set and the effectiveness of the B-index has been confirmed by comparison with other computing indices, namely the I-index, CH index and silhouette index. The clusters obtained by using SLHC with the B-index and MSC on IDS1Filt are quite consistent and thus confirm the effective of either method in clustering the high-dimensional OES data sets.

Chapter 8

Concluding Summary and Future Work

8.1 Concluding Summary

The research presented in this thesis was dedicated to the development of algorithms for effective unsupervised feature extraction from complex and highly redundant semiconductor plasma etching sensor data sets. These newly proposed methods can be clearly divided into two categories as follows:

1. statistical transformation
    - Principal Component Analysis based data summarisation;
    - Sparse Principal Component Analysis;
    - Adaptive Weighting Sparse Principal Component Analysis;
2. clustering
    - Max Separation Clustering;
    - Single Linkage Hierarchical Clustering/B-Index.

Existing feature extraction techniques employed in semiconductor plasma etching were reviewed in Chapter 2. This provides important methodology background for further algorithm development. As the most widely applied multivariate analysis method in

plasma etch, PCA (Principal Component Analysis) was employed in Chapter 3 to analyse the high-volume OES data sets. Graphical display of the results obtained using PCA is computationally expensive. As a low cost alternative, two PCA-based data summarisation methods were proposed. One is implemented as an improvement on conventional data unfolding approaches and the other is realised by monitoring changes in the directions of the PC loading vectors. Experimental results show that the two proposed methods are effective for identifying plasma etching process variations across wafers and lots. However, the issue with PCA is that since the PCs are linear combination of all underlying variables, it cannot be used to identify the key wavelengths, which are important when trying to determine the chemistry underlying causes of process changes. Moreover, the objective of PCA is to maximise the variance explained by low dimension representations of the data, and hence is not tailored to, or optimised for, relevant feature extraction.

Seeking possible solutions, the recently proposed Sparse Principal Component Analysis (SPCA) algorithm was employed and applied to OES data analysis for the first time. As experimental results showed, SPCA is useful for variable selection and is able to identify the variables that are highly correlated (variable grouping). However, it is not effective at separating the variables with reverse patterns and hence is not ideal for variable selection based on pattern differences. Since the variables in different components are not mutually exclusive, SPCA is not effective for variable classification.

As an improvement on SPCA, AWSPCA (Adaptive Weighting Sparse Principal Component Analysis) was proposed. AWSPCA can achieve effective selection of variables with different features, variable grouping and classifications. However, AWSPCA is unable to identify representative variables, nor the similarity levels between different variables. These are the common drawbacks of data transformation methods targeting at information summarisation rather than feature extraction.

Variable classification based on pattern differences is the objective of clustering. As a consequence methods drawn from the clustering domain were examined. The charac-

teristics and properties of three of the most powerful and widely used non-hierarchical clustering methods, K-means, SOM (Self-Organizing Map) and QT (Quality Threshold) were explored and discussed in detail. Addressing the issues of using these methods for feature extraction from OES data led to the development of the MSC (Max Separation Clustering) algorithm. MSC is effective for extracting and summarising the different patterns contained in OES data and the newly proposed maxoid in MSC is effective in representing the distinctive patterns. Moreover, MSC is not subject to inter-run variability and has no requirement for *a priori* knowledge of the number of clusters. However, MSC cannot provide detailed information on the levels of similarity between intra-cluster objects (variables) or across clusters.

Addressing this problem motivated us to develop the SLHC/B-Index (Single Linkage Hierarchical Clustering and B-Index) algorithms. The main advantage of the SLHC is that it can be used to disclose the relationship between clusters and intra-cluster objects. Used in conjunction with B-index, SLHC can provide effective clustering of objects with distinctive patterns. However, simply using SLHC itself does not provide information on the appropriate level of clustering (number of clusters). This is the function of the B-index metric. Whereas MSC naturally identifies representative variables for each cluster generated, SLHC does not.

For clarity, a summary of the strengths and weaknesses of the new algorithms proposed in this thesis is provided in Table 8.2 and the algorithm running time for different data sets is shown in Table 8.1. As can be seen, each method has advantages, but no methods stands out as having all the desired properties in terms of feature extraction. While SHLC provides the most complete information on a data set, it is computationally very expensive compared to other methods. Hence, in practice, the appropriate choice of algorithm will depend on problem requirements and time available. For the algorithms developed in this thesis, it is recommended that if the target is to achieve effective dimension reduction with respect to variations in the analysed variables, AWSPCA is applicable. If instead, the target is to achieve effective clustering of distinctive patterns within the data and pattern representation by a small number of variables within a limited time, then MSC is the appropriated choice. If computational time is not a

Data Set	PCA	SPCA	AWPCA	MSC	SLHC
SDS1	3.93×10^{-3}	6.29×10^{-2}	9×10^{-2}	1.19×10^{-2}	3.6×10^{-1}
IDS1Filt	1.96×10^{-1}	2.4×10^{-1}	6.54	6.03	1.996×10^{4}

Table 8.1: The algorithm running time for different data sets (unit: second), when running on a computer with 1.6GHz single core Intel Pentium processor and 752MB of memory.

concern and the goal is to achieve a detailed visualisation of the correlation between each individual variable and to a large extend, across all groups containing variables with different patterns, then SLHC should be considered.

In this thesis, MSC provides an effective summarisation and representation of the patterns contained in OES data, opening up possibilities of using OES data for accurate process control for semiconductor chip manufacturing. Moreover, SLHC provides a complete and detailed exploration and visulisation of the insight relationship between variables and across clusters, making it possible to achieve correct interpretation of complex plasma chemicals. In industry, this will help engineers to achieve a complete understanding of the underlying plasma chemistry, impossible presently but significantly important for the future accurate manufacturing with tiny features.

Methods		Dimension Reduction	Variable grouping	Feature classification	Relationship between intra-cluster objects and across clusters	Representative variable selection	Tuning Parameters
Data transformation	PCA-based data summarisation	√	X	X	X	X	1
	SPCA	√	√	X	X	X	3
	AWSPCA	√	√	√	X	X	2
Clustering	MSC	√	√	√	X	√	1
	SLHC/B-Index	√	√	√	√	X	1

Table 8.2: Summary of the strengths and weaknesses of the new algorithms proposed in this thesis for feature extraction from OES data.

8.2 Future Work

The algorithms proposed in this thesis open up new possibilities for future work. Some suggestions are discussed below, which are certainly not an exhaustive list.

Sensor Data Fusion

As discussed in Section 1.2.3, OES data contains rich chemical information and has the potential to be used to track the root causes of process variations. However, from the whole plasma etching process standpoint, the generated chemical species which OES measures are only process outputs. Inputs such as RF power supply, gas flow rate, plasma chamber pressure, wafer temperature that determine the plasma physics and chemistry were not considered.

From general data fusion theory, the use of various diagnostic data can reduce the effect of measurement noise and improve algorithm robustness. As such, it is worth exploring the relationship between variables measured by OES and other diagnostic sensors, such as Plasma Impedance Monitors and Process State Monitors. Analysing the combined data set, statistical modeling techniques, *e.g.* Partial Least Squares (PLS) and Factor Analysis (FA) are applicable. PLS attempts to model the relationship between input and output variables, while FA attempts to model the relationship between all variables included in the data set. If successful, the modeling process can then be followed by further information extraction processes to remove data redundancy introduced by the merging of data, leading to a complete implementation of data fusion.

Process Spatial Modeling

Modeling techniques such as artificial neural networks have been widely used in plasma etching process control, finding application in the prediction of the average etch rate or etch completion [33]. This kind of modeling is problematic, because it ignores the fact that significant spatial variations occur in etch across the wafer, known as the uniformity problem [134]. White *et. al.* [170] have shown that it is feasible to model line width reduction across the wafer based on the use of spatially resolved OES (placing three independent OES beams to resolve the spectral data across the wafer). The technique

of constructing 3D modeling based on spatially resolved 2D images is reasonably mature in the computer vision and image processing domains. If applicable, it may be feasible to construct a 3D model for simulating the spatial and temporal distribution of the optical emissions of chemical species in an etch chamber, enabling complete quality control at the wafer scale.

Microarray Data Analysis

The newly proposed algorithms, AWSPCA, MSC and SLHC/B-Index, open up potential opportunities to extend the existing techniques to microarray data analysis.

DNA microarrays are devices used to measure the expression of many thousands of genes in parallel [152]. Microarrays can be used diagnostically to determine the disease that an individual is suffering from and to predict the effectiveness of a course of therapy. The last decade has been seen a rapid growth in the use of microarray technology in medicine and pharmaceutical industries. The principal characteristics that microarray data has in common with OES data is that they are both high-volume, high dimension and highly redundant data sets. The SPCA technique, in fact, originated from microarray data analysis. In addition, many of the existing clustering techniques have been identified as typical algorithms in Bioinformatics for microarray data analysis. Due to rapid development of microarray technology, there are increased needs for more sophisticated and complete microarray data analysis techniques. The algorithms developed in this thesis can definitely be used to meet such needs.

Bibliography

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of SIGMOD*, 1998.

[2] R. L. Allen, R. Moore, and M. Whelan. Application of neural networks to plasma etch end point detection. *Journal of Vacuum Science and Technology B*, 14:498–503, 1996.

[3] A. Avoyan, F. C. Dassapa, and B. McMillin. Method of plasma etch endpoint detection using a VI probe diagnostics. *US Patent, 2005/0217795A1*, 2005.

[4] L. M. E. Beale. *Cluster analysis*. Scientific Control Systems, London, 1969.

[5] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6:281–297, 1999.

[6] P. Biolsi, L. Drachnik, S. Ellinger, and D. Morvay. An advanced endpoint detection solution for < 1 percentage open area applications: contact and via. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pages 391–396, Cambridge, MA, USA, 1996.

[7] A. Bouguettaya. On-line clustering. *IEEE Transactions on Knowledge and Data Engineering*, 8(2):333–339, 1996.

[8] L. M. Brandeau and S. S. Chiu. Parametric facility location in a tree network with an l_p norm cost function. *Transporation Science*, 22:59–69, 1988.

[9] E. Brigham. *The fast fourier transform and its applications*. Prentice Hall, 1988.

[10] S. Butterworth. On the theory of filter amplifiers. *Experimental Wireless and the Radio Engineer*, 7:536–541, 1930.

[11] J. Cadima and I. T. Jolliffe. Loadings and Correlations in the Interpretation of Principal Components. *Journal of Applied Statistics*, 22(2):203–214, 1995.

[12] B. R. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3:1–27, 1974.

[13] L. J. Cao and W. K. Chong. Feature extraction in support vector machine: A comparison of pca, kpca and ica. *Proceedings of the 9th International Conference on Neural Information Processing*, 2:1001–1005, 2002.

[14] J. F. Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE*, 86:2009–2025, 1998.

[15] B. R. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1:245–276, 1966.

[16] G. Chang, J. P. McVittie, J. T. Walker, and R. W. Dutton. Electrical endpoint detection of VLSI contact plasma etching. *IEEE Electron Device Letters*, 5:514–517, 1984.

[17] B. Chapman. *Glow discharge processes*. John Wiley & Sons Inc., 1980.

[18] S. Chatterjee and A. S. Hadi. *Regression Analysis by Example*. Wiley-Interscience, 4th edition, 2006.

[19] F. F. Chen and J. P. Chang. *Lecture notes on principles of plasma processing*. Kluwer Academic/Plenum Publishers, 2003.

[20] R. Chen, H. Huang, C. J. Spanos, and M. Gatto. Plasma etch modelling using optical emission spectroscopy. *Journal of Vaccum Science & Technology A: Vacuum, Surfaces, and Films*, 14:1901–1906, 1996.

[21] G. Cherry and S. J. Qin. Fisher discriminant analysis for semiconductor batch tool matching. In *Proceedings of the 17th World Congress, The International Federation of Automatic Control*, pages 9144–9148, 2008.

[22] G. A. Cherry and S. J. Qin. Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis. *IEEE Transactions on Semiconductor Manufacturing*, 19:159–172, 2006.

[23] D. A. Cliff, O. Haggett, R. M. Smallman-Raynor, F. D. Stroup, and D. G. Williamson. The application of multidimensinal scaling methods to epidemiological data. *Statistical Methods in Medical Research*, 4:102–123, 1995.

[24] International Roadmap Committee. Executive summary. Technical report, International Technology Roadmap for Semiconductors, 2007.

[25] J. M. Czebiniak. End point detection of plasma etching using optical methods. In *24th. Annual Microelectronic Engineering Conference Proceedings*, pages 46–48, Rochester, NY, 2006.

[26] F. Daly, D. J. Hand, M. C. Jones, A. D. Lunn, and K. J. McConway. *Elements of Statistics*. Addison-Wesley Publishing Company, 1995.

[27] A. D'Aspremont, L. Ghaoui, M. I. Jordan, and G. Lanckriet. A Direct Formulation for Sparse PCA Using Semidefinite Programming. *SIAM Review*, 49(3):434–448, 2007.

[28] D. L. Davies and D. W. Bouldin. A cluster separation measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:224–227, 1979.

[29] M. N. A. Dewan, P. J. McNally, T. Perova, and P. A. F. Herbert. Use of plasma impedance monitoring for the determination of SF_6 reactive ion etch process end points in a SiO_2/Si system. *Mat Res Innovat*, 5:107–116, 2001.

[30] M.N.A. Dewan, P.J. McNally, T. Perova, and P.A.F. Herbert. Determination of SF_6 reactive ion etching end point of the SiO_2/Si system by plasma impedance monitoring. *Microelectronic Engineering*, 65:25–46, 2003.

[31] O. R. Duda and E. P. Hart. *Pattern classification and scene analysis*. Wiley: New York, 1973.

[32] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1973.

[33] T. F. Edgar, S. W. Butler, W. J. Campbell, C. Pfeiffer, C. Bode, S. B. Hwang, K. S. Balakrishnan, and J. Hahn. Automatic control in microelectronic manufacturing: practives, challenges, and possibilities. *Automatica*, 36:1567–1603, 2000.

[34] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *Annals of Statistics*, 32(2):407–499, 2004.

[35] V. Eruhimov, V. Martyanov, and E. Tuv. Change-point detection with supervised learning and feature selection. In *Proceedings of ICINCO*, pages 359–363, Angers, France, 2007.

[36] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of KDD-96*, 1996.

[37] B. S. Everitt, S. Landar, and M. Leese. *Clustering analysis*. Arnold and Oxford University Press, 2001.

[38] A. Ferreira, A. Roussy, and L. Conde. Virtual metrology models for predicting physical measurement in semiconductor manufacturing. *Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pages 149–154, 2009.

[39] G. Fortunato. End-point determination by reflected power monitoring. *Journal of Physics E: Scientific Instruments*, 20:1051–1052, 1987.

[40] C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[41] K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, 1989.

[42] X. Ge and P. Smyth. Segmental semi-markov models for change-point detection with application to semiconductor manufacturing. Technical Report UCI-ICS 00-08, University of California, Irvine, 2000.

[43] X. Ge and P. Smyth. Hidden markov models for endpoint detection in plasma etch processes. Technical Report UCI-ICS 01-54, University of California, Irvine, 2001.

[44] P. Geladi. Analysis of multiway (multi-mode) data. *Chemometrics and Intelligent Laboratory Systems*, 7:11–30, 1989.

[45] P. Geladi and B. R. Kowalski. Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

[46] D. Gervini and V. Rousson. Criteria for Evaluating Dimension-reducing Components for Multivariate Data. *American Statistician*, 58(1):72–76, 2004.

[47] R. Gnanadesikan. *Methods for statistical data analysis of multivariate observations*. John Wiley & Sons, Inc., second edition edition, 1997.

[48] B. E. Goodlin. *Multivariate Endpoint Detection of Plasma Etching Processes*. PhD thesis, Massachusetts Institute of Technology, United States, 2002.

[49] D. A. Gordon. *Classification*. Chapman & Hall/CRC, 1999.

[50] C. J. Gower. A comparison of some methods of cluster analysis. *Biometrics*, 23:623–628, 1967.

[51] K. Han, S. Kim, K. J. Park, E. S. Yoon, and H. Chae. Principal component analysis based support vector machine for the end point detection of the metal etch process. *Proceedings of the 17th IFAC World Congress*, pages 4560–4565, 2008.

[52] K. Han, J. W. Lee, H. Chae, K. H. Han, K. J. Park, S. K. Park, and E. S. Yoon. Automatic end point detection of plasma etching process using the multiway PCA of the whole optical emission spectrum. In *SICE-ICASE International Joint Conference Proceedings*, pages 1709–1714, Bexco, Busan, Korea, 2006.

[53] K. Han, E. S. Yoon, J. Lee, H. Chae, K. H. Han, and K. J. Park. Real-time endpoint detection using modified principal component analysis for small open area SiO2 plasma etching. *Industrial & Engineering Chemistry Research*, 47:3907–3911, 2008.

[54] E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76:175–181, 2000.

[55] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2001.

[56] S. Haykin. *Neural network, a comprehensive foundation.* Prentice-Hall, Inc, 1999.

[57] N. He, J. M. Zhang, and S. Q. Wang. Combination of independent component analysis and multi-way principal component analysis for batch process monitoring. *IEEE International Conference on System, Man and Cybernetics*, pages 530–535, 2004.

[58] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysisof coexpressed genes. *Genome Research*, 9:1106–1115, 1999.

[59] C. D. Himmel and G. S. May. Advantages of plasma etch modelling using neural networks over statistical techniques. *IEEE Transactions on Semiconductor Manufacturing*, 6:103–111, 1993.

[60] A. E. Hoerl and R. W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970.

[61] S. Hong and G. May. Neural-network-based sensor fusion of optical emission and mass spectroscopy data for real-time fault detection in reactive ion etching. *IEEE Transactions on Industrial Electronics*, 52:1063–1072, 2005.

[62] S. J. Hong and G. S. May. Neural network-based real-time malfunction diagnosis of reactive ion etching using in situ metrology data. *IEEE Transactions on Semiconductor Manufacturing*, 17:408–421, 2004.

[63] S. J. Hong, G. S. May, and D-C. Park. Neural network modeling of reactive ion etcing using optical emission spectroscopy data. *IEEE Transactions on Semiconductor Manufacturing*, 16:598–608, 2003.

[64] S. J. Hong, G. S. May, and D. C. Park. Neural network modelling of reactive ion etching using principal component analysis of optical emission spectroscopy data. *IEEE Transactions on Semiconductor Manufacturing*, 16:598–608, 2003.

[65] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441; 498–520, 1933.

[66] J. Huang and M. Yang. Endpoint control for small open area by RF source parameter VDC. *US Patent, 6930049B2*, 2005.

[67] A. Hyvarinen, H. Karhunen, and E. Oja. *Independent component analysis*. John Wiley & Sons, Inc., 2001.

[68] JR. J. F. Hair, R. E. Anderson, R. L. Tatham, and W. C. Black. *Multivariate Data Analysis*. Prentice Hall, 1998.

[69] J. E. Jackson. *A user's guide to principal components*. Wiley Interscience, New York, 1991.

[70] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, NJ, 1988.

[71] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[72] L. C. Jain and V. R. Vemuri. *Industrial applications of neural networks*, chapter The self-organizing map in industry analysis, pages 87–112. CRC Press, 1999.

[73] R. Jain and A. Koronios. Innovation in the cluster validating techniques. *Fuzzy Optimization and Decision Making*, 7:233–241, 2008.

[74] J. N. R. Jeffers. Two Case Studies in the Application of Principal Component Aanlysis. *Applied Statistics*, 16(3):225–236, 1967.

[75] D. Jiang, J. Pei, and A. Zhang. DHC: a density-based hierarchical clustering method for time-series gene expression data. *Proceedings BIBE: Third IEEE International Symposium Bioinformatics and Bioengineering*, pages 393–400, 2003.

[76] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: a survey. *IEEE Transactions on knowledge and data engineering*, 16(11):1370–1386, 2004.

[77] D. E. Johnson. *Applied multivariate methods for data analysts.* Duxbury Press, 1998.

[78] I. T. Jolliffe. *Principal component analysis.* Springer, 2002.

[79] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A Modified Principal Component Technique Based on the LASSO. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

[80] F. H. Kaiser. The Varimax Criterion for Analytic Rotation in Factor Analysis. *Psychometrika*, 23:187–200, 1958.

[81] L. Kaufman and J. P. Rousseeuw. *An introduction to cluster analysis*, chapter Finding groups in data. Wiley-Interscience, New York, 1990.

[82] B. Kim, J. K. Bae, and W. S. Hong. Plasma control using neural network and optical emission spectroscopy. *Journal of Vacuum Science and Technology A*, 23:355–358, 2005.

[83] B. Kim and W. Choi. Discrete wavelet monitoring of plasma impedance matching for process control. 6^{th} *IEEE International Symposium on Industrial Electronics*, II:171–175, 2001.

[84] B. Kim, D. Han, S. Moon, and K. K. Lee. Modeling of sidewall bottom etching using a neural network. *Journal of the Korean Physical Society*, 46:1365–1370, 2005.

[85] B. Kim and S. Kim. Partial diagnostic data to plasma etch modeling using neural network. *Microelectronic Engineering*, 75(4):397–404, 2004.

[86] B. Kim, K. Kwon, S. Kwon, J. Park, S. Yoo, K. Park, and I. You. Modeling etch rate and uniformity of oxide via etching in a CHF_3/CF_4 plasma using neural networks. *The Solid Films*, 426(1-2):8–15, 2003.

[87] B. Kim and B. T. Lee. Effect of plasma and control parameters on SiC etching in a C_2F_6 plasma. *Plasma Chemistry and Plasma Processing*, 23(3):489–499, 2003.

[88] J. Kogan. *Introduction to clustering large and high-dimensional data*. Cambridge University Press, 2007.

[89] A. T-C Koh, N. F. Thornhill, and V. J. Law. Principal component analysis of plasma harmonics in end-point detection of photoresist stripping. *Electronics Letters*, 35:1383–1385, 1999.

[90] T. Kohonen. *Self-organizing maps*. Springer-Verlag, Heidelberg, 1995.

[91] T. Kohonen. Exploration of very large databases by self-organizing maps. *International conference on neural networks*, 1:PL1–PL6, 1997.

[92] T. Kourt. Application of latent variable methods to process control and multivariate statistical process control in industry. *International Journal of Adaptive Control Signal Process*, 19:213–246, 2005.

[93] T. Kourti and F. J. Macgregor. Process analysis monitoring and diagnosis using the multivariate projection methods. *Chemometrics & Intelligent Laboratory Systems*, 28:3–21, 1995.

[94] T. Kourti and J. F. MacGregor. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28:3–21, 1995.

[95] T. Kourti and J. F. MacGregor. Recent developments in multivariate SPC methods for monitoring and diagnosing process and product performance. *Journal of Quality Technology*, 28:409–428, 1996.

[96] F. E. Krause. *Taxicab Ceometry*. Addison Wesley, Menlo Park, CA, 1975.

[97] N. G. Lance and T. W. Williams. Computer programs for hierarchical polythetic classification. *Computer Journal*, 9:60–64, 1966.

[98] N. G. Lance and T. W. Williams. A general theory of classificatory sorting strategies I. hierarchical systems. *Computer Journal*, 9:373–380, 1967.

[99] C. R. Larson and G. Sadiq. Facility locations with the Manhattan metric in the presence of barriers to travel. *Operations Research*, 31:652–699, 1983.

[100] J. M. Lee, S. J. Qin, and I. B. Lee. Fault detection and diagnosis based on modified independent compoenent analysis. *AIChE Journal*, 52:3501–3514, 2006.

[101] S. Lee and C. Spanos. Prediction of wafer state after plasma processing using real-time tool data. *IEEE Transactions on Semiconductor Manufacturing*, 8(3):252–261, 1995.

[102] C. T. Leondes. *Image Processing and Pattern Recognition.* Academic Press, 1998.

[103] M. D. Levine. Feature extraction: a survey. *Proceedings of the IEEE*, 57:1391–1407, 1969.

[104] F. Li, G. C. Runger, and E. Tuv. Supervised learning for change-point detection. *International Journal of Production Research*, 44:2853–2868, 2006.

[105] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin. Recursive pca for adaptive process monitoring. *Journal of Process Control*, 10:471–486, 2000.

[106] M. A. Lieberman and A. J. Lichtenberg. *Principle of plasma discharges and material processing.* John Wiley & Sons, New York, 1994.

[107] T. H. Lin, F. T. Cheng, W. M. Wu, C. A. Kao, A. J. Ye, and F. C. Chang. Nn-based key -variable selection method for enhancing virtual metrology accuracy. *IEEE Transactions on Semiconductor Manufacturing*, 22(1):204–211, 2009.

[108] H.-P. Kriegel J. Sander M. Ankerst, M. Breunig. Optics: ordering points to identify the clustering structure. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1999.

[109] B. B. Ma, S. McLoone, and J. Ringwood. Tracking plasma etch process variations using principal component analysis of OES data. In *International Conference on Informatics in Control, Automation and Robotics Proceedings*, pages 361–364, Anger, France, 2007.

[110] J. F. MacGregor and T. Kourti. Statistical process control of multivariate process. *Control Engineering Practice*, 3:403–414, 1995.

[111] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[112] R. Malone, M. Schardin, and A. Steinbach. A new method for very low open area endpoint detection.

[113] D. M. Manos and D. L. Flamm. *Plasma etching, an introudction.* Academic Press, New York, 1988.

[114] J. Mao and A. K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6:296–317, 1995.

[115] R. L. Mason, Y. Chou, and J. C. Young. Applying hotelling's t2 statistic to batch processes. *Journal of Quality Technology*, 33:466–479, 2001.

[116] U. Maulik and S. Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1650–1654, 2002.

[117] G. S. May and C. J. Spanos. *Fundamentals of Semiconductor Manufacturing and Process Control.* John Wiley & Sons. Inc. New Jersey, 2006.

[118] H. L. Maynard, E. A. Rietman, J. T. Lee, and N. Layadi. Plasma etching endpointing by monitoring radio-frequency power systems with an artificial neural network. *Journal of Electrochemical Society*, 143:2029–2035, 1996.

[119] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

[120] B. Mirkin. *Clustering for data mining.* Chapman & Hall/CRC, 2005.

[121] D. Montgomery and G. Runger. *Applied statistics and probability for engineers.* John Wiley & Sons, New York, 2002.

[122] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38:114–117, 1965.

[123] V. Patel, B. Singh, and J. H. Thomas III. Reactive ion etching end-point determination by plasma impedance monitoring. *Applied Physics Letters*, 61:1912–1914, 1992.

[124] K. Pearson. On lines and planes of closest fit to a system of points in space. *Philosophical Magazine*, 2:557–572, 1901.

[125] S. J. Qin and R. Dunia. Determining the number of principal components for best reconstruction. *Journal of Process Control*, 10:245–250, 2000.

[126] E. Ragnolia, S. McLoone, S. Lynn, J. Ringwood, and N. MacGearailt. Identifying key process characteristics and predicting etch rate from high dimensional datasets. In *Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pages 106–111, 2009.

[127] E. Ragnolia, S. McLoone, J. Ringwood, and N. MacGearailt. Matrix factorisation techniques for endpoint detection in plasma etching. In *Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pages 156–161, 2008.

[128] S. Rangan, C. Spanos, and K. Poolla. Modeling and filtering of optical emission spectroscopy data for plasma etching systems. *IEEE International Symposium on Semiconductor Manufacturing Conference Proceedings*, pages B41–B44, 1997.

[129] T. Reis. In situ plasma etch process endpoint control in integrated circuit manufacturing. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference Proceedings*, pages 55–57, Munich, Germany, 2001.

[130] A. C. Rendner. *Methods of Multivariate Analysis.* A John Wiley & Sons Inc., 2003.

[131] E. A. Rietman. A neural network model of a contact plasma etch process for vlsi production. *IEEE Transactions on Semiconductor Manufacturing*, 9:95–100, 1996.

[132] E. A. Rietman, R. C. Frye, E. R. Lory, and T. R. Harry. Active neural network control of wafer attributes in a plasma etch process. *Journal of Vacuum Science and Technology B*, 11:1314–1316, 1993.

[133] E. A. Rietman, J. T. Lee, and N. Layadi. Dynamic images of plasma processes: use of Fourier blobs for endpoint detection during plasma etching of patterned wafers. *Journal of Vacuum Science and Technology A*, 16:1449–1453, 1998.

[134] J. Ringwood, S. Lynn, G. Bacelli, B. Ma, E. Ragnoli, and S. McLoone. Estimation and control in semiconductor etch: Practice and possibilities. *IEEE Transactions on Semiconductor Manufacturing*, in publication, 2009.

[135] H. Ritter, T. Martinetz, and K. Schulten. *Neural computation and self-organizing maps: an introduction*. Addison-Wesley Longman Publishing Co., Inc., 1992.

[136] J. P. Roland, P. J. Marcoux, G. W. Ray, and G. H. Rankin. Endpoint detection in plasma etching. *Journal of Vacuum Science and Technology A*, 3:631–636, 1985.

[137] T. Sarmiento, S. J. Hong, and G. S. May. Fault detection in reactive ion etching systems using one-class support vector machines. In *Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pages 1–4, 2005.

[138] O. G. Selfridge and U. Neisser. *Computers and Thought*, chapter Pattern Recognition by Machine, pages 237–250. McGraw-Hill, 1963.

[139] G. S. Selwyn. *Optical diagnostic techniques for plasma processing*. American Vacuum Society Press, 1993.

[140] R. Shamir and R. Sharan. Click: a clustering algorithm for gene expression analysis. In *Proceedings of Eighth International Conference Intelligent Systems for Molecular Biology*, 2000.

[141] H. Shen and J. Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99(6):1015–1034, 2008.

[142] P. H. Singer. Diagnosing plasma and detecting endpoints. *Semiconductor International*, 11:66–70, 1988.

[143] K. Sjöstrand, M. B. Stegmann, and R. Larsen. Sparse Principal Component Analysis in Medical Shape Modeling. In *Proceedings of SPIE*, volume 6144, page 61444X, March 2006.

[144] A.K. Smilde and D. A. Doornbos. Three way methods for the calibration of chromatographic systems: Comparing parafac and three-way pls. *Journal of Chemometrics*, 5:345–360, 1991.

[145] T. H. Smith and D.S. Boning. Process control in semiconductor manufacturing. In *Industrial Research Conference Proceedings*, 1999.

[146] A. H. P. Sneath. The application of computers to taxonomy. *Journal of General Microbiology*, 17:201–226, 1957.

[147] R. R. Sokal and D. C. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.

[148] T. Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the begetation on danish commons. *Biology Skrifter*, 5:1–34, 1948.

[149] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72–101, 1904.

[150] G. Spitzlsperger, C. Schmidt, G. Ernst, H. Strasser, and Michaela Speil. Fault detection for a via etch process using adaptive multivariate methods. *IEEE Transactions on Semiconductor Manufacturing*, 18:528–533, 2005.

[151] M. Splichal and H. Anderson. Application of chemometrics to optical emission spectroscopy for plasma monitoring. *Proceedings of SPIE*, 2:189–203, 1987.

[152] D. Stekel. *Microarray bioinformatics*. Cambridge University Press, 2003.

[153] S. J. Strauss, J. J. Bartko, and T. W. Carpenter. The use of clustering techniques for the classification of psychiatric petients. *British Journal of Psychiatry*, 122:351–540, 1973.

[154] M. Sugawara. *Plasma Etching: Fundamentals and Applications*. Oxford University Press, New York, 1998.

[155] E. M. Tamil, H. M. Radzi, M. Y. I. Idris, and A. M. Tamil. A review on feature extraction & classification techniques for biosignal processing. In *Proceedings of* 4^{th} *Kuala Lumpur International Conference on Biomedical Engineering*, pages 113–116, 2008.

[156] C. Tang, L.Zhang, A. Zhang, and M. Ramanathan. Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In *Proceedings of BIBE: Second IEEE International Symposium Bioinformatics and Bioengineering*, pages 41–48, 2001.

[157] C. Tang and A. Zhang. An iterative strategy for pattern discovery in high-dimensional data sets. In *Proceedings of 11th International Conference of Information and Knowledge Management (CIKM)*, pages 10–17, 2002.

[158] A. Tefferi, E. Bolander, M. Ansell, D. Wieben, and C. Spelsberg. Primer on medical genomics part III: microarray experiments and data analysis. In *Mayo Clinic Proceedings*, volume 77, pages 927–940, 2002.

[159] S. Thomas, H. H. Chen, C. K. Hanish, J. W. Grizzle, and S. W. Pang. Minimized response time of optical emission and mass spectrometric signals for optimized endpoint detection. *J. Vac. Sci. Technol. B*, 14:2531–2536, 1996.

[160] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of Royal Statistical Society B*, 58(1):267–288, 1996.

[161] A. J. Toprac and H. Yue. Methods of determining etch endpoint using principal components analysis of optical emission spectra. *US Patent, 6582618 B1*, 2003.

[162] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas. SOM toolbox for Matlab 5. Technical report, SOM Toolbox Team, Helsinki University of Technology, 2000.

[163] S. K. Vines. Simple Principal Components. *Applied Statistics*, 49(4):441–451, 2000.

[164] H. Wang and C. Leng. Unified lasso estimation by least squares approximation. *Journal of the American Statistical Association*, 102(479):1039–1048, 2007.

[165] W. Wang, J. Bi, and J. Zhao. Plasma etching process monitoring with optical emission spectroscopy. *Proceedings of International Conference on Industrial Mechatronics and Automation*, pages 45–47, 2009.

[166] H. J. Ward. Hierarchical groupings to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[167] M. West, C. Blanchettem, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Marks, and I. Nevins. Predicting the clinical status of human breast cancer using gene expression profiles. In *Proceedings of the National Academy of Sciences*, volume 98, pages 11462–11467, USA, 2001.

[168] D. A. White. *Multivariate analysis of spectral measurements for the caracterization of semiconductor processes.* PhD thesis, Massachusetts Institute of Technology, United States, 2001.

[169] D. A. White and D. Boning. Spatial characterization of wafer state using principal component analysis of optical emission spectra in plasma etch. *IEEE Transactions on Semiconductor Manufacturing*, 10:52–61, 1997.

[170] D. A. White, D. Boning, S. W. Butler, and G. G. Barna. Spatial characterization of wafer state using principal component analysis of optical emission spectra in plasma etch. *IEEE Transactions on Semiconductor Manufacturing*, 10:52–61, 1997.

[171] D. A. White, B. E. Goodlin, A. E. Gower, D. S. Boning, H. Chen, H. H. Sawin, and T. J. Dalton. Low open-area endpoint detection using a PCA-based T2 statistic and Q statistic on optical emission spectroscopy measurements. *IEEE Transactions on Semiconductor Manufacturing*, 13:193–207, 2000.

[172] D. A. White, B. E. Goodlin, A. E. Gower, D. S. Boning, H. Chen, H. H. Sawin, and T. J. Dalton. Low open-area endpoint detection using a PCA-based T2 statistic and Q statistic on optical emission spectroscopy measurements. *IEEE Transactions on Semiconductor Manufacturing*, 13:193–207, 2000.

[173] J. W. Winniczek. Full spectrum endpoint detection. *US Patent, 6969619 B1*, 2005.

[174] B. M. Wise, N. B. Gallagher, S. W. Butler, D. D. White JR, and G. G. Barna. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semicondutor etch process. *Journal of Chemometrics*, 13:379–396, 1999.

[175] B. M. Wise, N. B. Gallagher, and E. B. Martin. Application of PARAFAC2 to fault detection and diagnosis in semiconductor etch. *Journal of Chemometrics*, 15:285–298, 2001.

[176] M. B. Wise and N. B. Gallagher. Pls toolbox version 2.0. Technical report, Eigenvector Research, Inc., 1998.

[177] S. Wold. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4):397–405, 1978.

[178] S. Wold, P. Geladi, K. Esbensen, and J. Ohman. Multi-way principal components and pls analysis. *Journal of Chemometrics*, 1:41–56, 1987.

[179] X. L. Xie and G. Beni. A validity measures for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:841–847, 1991.

[180] E. P. Xing and R. M. Karp. Cliff: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17(1):306–315, 2001.

[181] J. Y. Yang, Y. T. Chen, and W. B. Lin. Method for detecting endpoint in plasma etching by impedance change. *US Patent, 6599759 B2*, 2003.

[182] H. C. Yew. Method for determining endpoint of etch layer and etching process implementing said method in semiconductor element fabrication. *US Patent, 7045467 B2*, 2006.

[183] H. Yue. Method and apparatus for detecting endpoint. *WIPO Patent, WO/2004/042803*, 2004.

[184] H. H. Yue and J. Qin. Reconstruction-based fault identification using a combined index. *Ind. Eng. Chem. Res.*, 40:4403–4414, 2001.

[185] H. H. Yue, J. Qin, J. Wiseman, and A. Toprac. Plasma etching endpoint detection using multiple wavelengths for small open-area wafers. *J. of Vacuum Science and Technology A: Vacuum, Surfaces, and Films*, 19:66–75, 2001.

[186] H. H. Yue, J. S. Qin, J. Wiseman, C. Nauert, and M. Gatto. Fault detection of plasma etchers using optical emission spectra. *IEEE Transactions on Semiconductor Manufacturing*, 13:374–385, 2000.

[187] H. H. Yue, S. J. Qin, J. Wiseman, and A. Toprac. Plasma etching endpoint detection using multiple wavelengths for small open-area wafers. *Journal of Vaccum Science & Technology A: Vacuum, Surfaces, and Films*, 19:66–75, 2001.

[188] H. H. Yue and M. Tomoyasu. Weighted principal component analysis and its applications to improve fdc performance. *Proceedings of 43rd IEEE Conference on Decision and Control*, pages 4262–4267, 2004.

[189] A. Zhang. *Advanced analysis of gene expression microarray data*. World Scientific Publishing Co. Pte. Ltd., 2006.

[190] J. Zhang, E. B. Martin, and A. J. Morris. Fault detection and diagnosis using multivariate statistical techniques. *Transactions of Chemical Engineering Research & design*, 74:89–96, 1996.

[191] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *11th International Conference of Information and Knowledge Management (CIKM)*, pages 515–524, 2002.

[192] X. Zhu and A. B. Goldberg. *Lectures on Artificial Intelligence and Machine Learning*, chapter Introduction to Semi-Supervised Learning. Morgan & Claypool Publishers, 2009.

[193] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[194] H. Zou and T. Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of Royal Statistical Society B*, 67(2):301–320, 2005.

[195] H. Zou, T. Hastie, and R. Tibshirani. Sparse Principal Component Analysis. Technical report, Department of Statistics, Standford University, 2004. Available at http://www-stat.stanford.edu/ hastie/Papers/sparsepc.pdf.

[196] H. Zou, T. Hastie, and R. Tibshirani. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

Appendix A

Mathematical Proofs

A.1 Proof of Theorem 1 in Section 4.2.6

Theorem 1 provides a self-contained ridge regression model for computing PCA.

Theorem: Given ridge regression estimates of $\mathbf{a}$ and $\mathbf{b}$, denoted as $\widehat{\mathbf{a}}^{\mathrm{R}}$ and $\widehat{\mathbf{b}}^{\mathrm{R}}$, computed as

$$(\widehat{\mathbf{a}}^{\mathrm{R}}, \widehat{\mathbf{b}}^{\mathrm{R}}) = \arg\min_{\mathbf{a},\mathbf{b}} \{\sum_{i=1}^{m} ||\mathbf{z}_i - \mathbf{a}\mathbf{b}^{\mathrm{T}}\mathbf{z}_i||_2^2 + \gamma_2 ||\mathbf{b}||_2^2\}, \text{ s.t. } \mathbf{a}^{\mathrm{T}}\mathbf{a} = 1 \tag{A.1}$$

then

$$\mathbf{p}_1 = \widehat{\mathbf{b}}^{\mathrm{R}}(1 + \frac{\gamma_2}{\sigma_1^2}), \tag{A.2}$$

σ_1 is the largest singular value of $\mathbf{X}$. $\mathbf{z}_i$ is the i^{th} column of matrix $\mathbf{Z}$, $\mathbf{Z} = \mathbf{X}^{\mathrm{T}}$.

To prove this theorem, some properties of vectors and matrices will be referred to as follows.

Properties:

1. $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m], \mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_m]$, $\mathbf{B} \in \mathbb{R}^{q \times m}$, then
$$\sum_{i=1}^{m} \mathbf{a}_i \mathbf{b}_i^{\mathrm{T}} = \mathbf{A}\mathbf{B}^{\mathrm{T}};$$
2. $a \in \mathbb{R}$, then $\mathrm{Tr}\{a\} = a$;
3. $\mathbf{A} \in \mathbb{R}^{m \times m}$, then $\mathrm{Tr}\{\mathbf{A}\} = \mathrm{Tr}\{\mathbf{A}^{\mathrm{T}}\}$;
4. $\mathbf{a} \in \mathbb{R}^{n \times 1}, \mathbf{b} \in \mathbb{R}^{n \times 1}$, then
$$\mathbf{a}^{\mathrm{T}}\mathbf{b} = \mathrm{Tr}\{\mathbf{a}\mathbf{b}^{\mathrm{T}}\} = \mathrm{Tr}\{\mathbf{b}^{\mathrm{T}}\mathbf{a}\} = \mathrm{Tr}\{\mathbf{a}^{\mathrm{T}}\mathbf{b}\} = \mathrm{Tr}\{\mathbf{b}\mathbf{a}^{\mathrm{T}}\};$$

Define J as a scalar cost function of $\mathbf{a}$ and $\mathbf{b}$, $J = \sum_{i=1}^{m} ||\mathbf{z}_i - \mathbf{a}\mathbf{b}^{\mathrm{T}}\mathbf{z}_i||_2^2 + \gamma_2||\mathbf{b}||_2^2$. This can be expressed as

$$
\begin{aligned}
J &= \sum_{i=1}^{m} ||\mathbf{z}_i - \mathbf{a}\mathbf{b}^{\mathrm{T}}\mathbf{z}_i||_2^2 + \gamma_2||\mathbf{b}||^2 \\
&= \sum_{i=1}^{m} [(\mathbf{I} - \mathbf{a}\mathbf{b}^{\mathrm{T}})\mathbf{z}_i]^{\mathrm{T}}[(\mathbf{I} - \mathbf{a}\mathbf{b}^{\mathrm{T}})\mathbf{z}_i] + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b} \\
&= \sum_{i=1}^{m} \mathrm{Tr}\{[(\mathbf{I} - \mathbf{a}\mathbf{b}^{\mathrm{T}})\mathbf{z}_i]^{\mathrm{T}}[(\mathbf{I} - \mathbf{a}\mathbf{b}^{\mathrm{T}})\mathbf{z}_i]\} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b} \\
&= \sum_{i=1}^{m} \mathrm{Tr}\{\mathbf{z}_i^{\mathrm{T}}(\mathbf{I} - \mathbf{b}\mathbf{a}^{\mathrm{T}})(\mathbf{I} - \mathbf{a}\mathbf{b}^{\mathrm{T}})\mathbf{z}_i\} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b} \\
&= \sum_{i=1}^{m} \mathrm{Tr}\{(\mathbf{I} - \mathbf{b}\mathbf{a}^{\mathrm{T}})(\mathbf{I} - \mathbf{a}\mathbf{b}^{\mathrm{T}})\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}}\} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b} \\
&= \mathrm{Tr}\{(\mathbf{I} - \mathbf{b}\mathbf{a}^{\mathrm{T}})(\mathbf{I} - \mathbf{a}\mathbf{b}^{\mathrm{T}})\sum_{i=1}^{n}(\mathbf{z}_i\mathbf{z}_i^{\mathrm{T}})\} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b} \\
&= \mathrm{Tr}\{(\mathbf{I} - \mathbf{b}\mathbf{a}^{\mathrm{T}})(\mathbf{I} - \mathbf{a}\mathbf{b}^{\mathrm{T}})\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b} \\
&= \mathrm{Tr}\{\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} - \mathrm{Tr}\{\mathbf{b}\mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} - \mathrm{Tr}\{\mathbf{a}\mathbf{b}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} + \mathrm{Tr}\{\mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b} \\
&= \mathrm{Tr}\{\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} - \mathrm{Tr}\{\mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b}\} - \mathrm{Tr}\{\mathbf{a}\mathbf{b}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} + \mathrm{Tr}\{\mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b} \\
&= \mathrm{Tr}\{\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} - \mathrm{Tr}\{\mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b}\} - \mathrm{Tr}\{\mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b}\} + \mathrm{Tr}\{\mathbf{b}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b}\} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b} \\
&= \mathrm{Tr}\{\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} - \mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b} - \mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b}
\end{aligned}
$$

Thus, finally J can be rewritten as follows

$$J = \mathrm{Tr}\{\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} - 2\mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b}. \tag{A.3}$$

Minimize J for a given $\mathbf{a}$ (taking $\mathbf{a}$ as a known parameter), gives

$$\frac{\partial J}{\partial \mathbf{b}} = \mathbf{0}.$$

Substituting J by Eq. (A.3), gives

$$\frac{\partial(\mathrm{Tr}\{\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} - 2\mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b} + \gamma_2\mathbf{b}^{\mathrm{T}}\mathbf{b}}{\partial \mathbf{b}} = \mathbf{0}.$$

$$
\begin{aligned}
-2(\mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{\mathrm{T}} + 2(\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b}) + 2\gamma_2\mathbf{b} &= \mathbf{0} \\
\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{b} - \mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{a} + \gamma_2\mathbf{b} &= \mathbf{0}.
\end{aligned}
$$

Therefore,

$$\mathbf{b} = (\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \gamma_2\mathbf{I})^{-1}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{a}. \tag{A.4}$$

Applying this result to replace $\mathbf{b}$ in Eq. (A.3) gives

$$J = \mathrm{Tr}\{\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\} - \mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \gamma_2\mathbf{I})^{-1}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{a}, \quad \text{s.t. } \mathbf{a}^{\mathrm{T}}\mathbf{a} = 1.$$

Since $\mathrm{Tr}\{\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\}$ is constant, minimising J is equivalent to maximising

$$\mathbf{a}^{\mathrm{T}}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \gamma_2\mathbf{I})^{-1}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}\mathbf{a}, \quad \text{s.t. } \mathbf{a}^{\mathrm{T}}\mathbf{a} = 1.$$

According to the Rayleigh quotient theory, $\mathbf{a}$ should be the first eigenvector of $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}} + \gamma_2\mathbf{I})^{-1}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}$. In fact, the eigenvectors of $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}}+\gamma_2\mathbf{I})^{-1}\mathbf{Z}\mathbf{Z}^{\mathrm{T}}$ are the same as the eigenvectors of $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}$. It is notable that the introduction of γ_2 only causes changes to the values of elements in the main diagonal of $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}$. Therefore, the eigenvalues of $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}$ are changed, but not the eigenvectors.

Therefore,

$$\widehat{\mathbf{a}} = \mathbf{v}_1.$$

Using this result and noting that $\mathbf{Z}\mathbf{Z}^{\mathrm{T}} = \mathbf{X}^{\mathrm{T}}\mathbf{X} = \mathbf{V}\Sigma^2\mathbf{V}^{\mathrm{T}}$ ($\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, SVD of $\mathbf{X}$), Eq. (A.4) can be rewritten as

$$\begin{aligned}
\widehat{\mathbf{b}} &= (\mathbf{V}\Sigma^2\mathbf{V}^{\mathrm{T}} + \gamma_2\mathbf{I})^{-1}\mathbf{V}\Sigma^2\mathbf{V}^{\mathrm{T}}\mathbf{v}_1 \\
&= (\mathbf{V}\Sigma^2\mathbf{V}^{\mathrm{T}} + \gamma_2\mathbf{V}\mathbf{V}^{\mathrm{T}})^{-1}\mathbf{V}\Sigma^2\mathbf{V}^{\mathrm{T}}\mathbf{v}_1 \\
&= [\mathbf{V}(\Sigma^2 + \gamma_2\mathbf{I})\mathbf{V}^{\mathrm{T}}]^{-1}\mathbf{V}\Sigma^2\mathbf{V}^{\mathrm{T}}\mathbf{v}_1 \\
&= \mathbf{V}(\Sigma^2 + \gamma_2\mathbf{I})^{-1}\Sigma^2\mathbf{V}^{\mathrm{T}}\mathbf{v}_1 \qquad\qquad (\mathbf{V} = \mathbf{V}^{-1}, \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}) \\
&= \mathbf{V}\mathbf{\Phi}\mathbf{V}^{\mathrm{T}}\mathbf{v}_1,
\end{aligned}$$

where

$$\mathbf{\Phi} = \begin{bmatrix} \frac{\sigma_1^2}{\sigma_1^2+\gamma_2} & & 0 \\ & \cdots & \\ 0 & & \frac{\sigma_p^2}{\sigma_p^2+\gamma_2} \end{bmatrix}$$

Therefore,

$$\widehat{\mathbf{b}}^{\mathrm{R}} = \frac{\sigma_1^2}{\sigma_1^2 + \gamma_2}\mathbf{v}_1 = \frac{\sigma_1^2}{\sigma_1^2 + \gamma_2}\mathbf{p}_1.$$

That is

$$\mathbf{p}_1 = \widehat{\mathbf{b}}^{\mathrm{R}}(1 + \frac{\gamma_2}{\sigma_1^2}). \tag{A.5}$$

This completes the proof. This theorem can be extended to the case of the first k loadings. The corresponding mathematical proof follows the same proof procedures as shown above and can be found in [194].

A.2 Relationship Between SNR and Correlation Coefficient

Consider two mean-centered signals, x and y. Let $\widetilde{x}$ and $\widetilde{y}$ be the corresponding signals corrupted by independent noise n_1 and n_2, respectively:

$$\widetilde{x} = x + n_1,\ \widetilde{y} = y + n_2, \tag{A.6}$$

where $E(n_1) = E(n_2) = 0$ and $E(n_1^2) = E(n_2^2) = \sigma^2$. Then, it follows that

$$\frac{corr(\widetilde{x}, \widetilde{y})}{corr(x, y)} = \sqrt{\frac{SNR_x}{1 + SNR_x}\frac{SNR_y}{1 + SNR_y}}. \tag{A.7}$$

The signal mean and variance, $\overline{x}$ and σ_x^2, can be expressed as:

$$\overline{x} = E(\widetilde{x}) = \frac{1}{m}\sum_{i=1}^{m}(\widetilde{x}_i) \tag{A.8}$$

and

$$\sigma_x^2 = E(\widetilde{x} - \overline{x})^2 = E(\widetilde{x}^2) - \overline{x}^2, \tag{A.9}$$

respectively. The Pearson's correlation between $\widetilde{x}$ and $\widetilde{y}$ can be expressed as:

$$corr(\widetilde{x}, \widetilde{y}) = \frac{E[(\widetilde{x} - \overline{x})(\widetilde{y} - \overline{y})]}{\sqrt{E(\widetilde{x} - \overline{x})^2 E(\widetilde{y} - \overline{y})^2}} \tag{A.10}$$

Because $\widetilde{x}$ and $\widetilde{y}$ are mean-centered, there is $E(\widetilde{x}) = E(\widetilde{y}) = 0$. Then,

$$corr(\widetilde{x}, \widetilde{y}) = \frac{E(\widetilde{x}\widetilde{y})}{\sqrt{E(\widetilde{x}^2)E(\widetilde{y}^2)}} = \frac{E(xy + n_1 y + n_2 x + n_1 n_2)}{\sqrt{E(x^2 + 2n_1 x + n_1^2)E(y^2 + 2n_2 y + n_2^2)}} \tag{A.11}$$

With the assumption that noise signals are independent, there is $E(n_1) = E(n_2) = 0, E(n_1 n_2) = 0$, so

$$corr(\widetilde{x}, \widetilde{y}) = \frac{E(xy)}{\sqrt{[E(x^2) + \sigma_{n_1}^2][E(y^2) + \sigma_{n_2}^2]}} \tag{A.12}$$

Similarly, there is

$$corr(x,y) = \frac{E(xy)}{\sqrt{E(x^2)E(y^2)}} \tag{A.13}$$

Therefore

$$\frac{corr(\widetilde{x},\widetilde{y})}{corr(x,y)} = \frac{\sqrt{E(x^2)E(y^2)}}{\sqrt{[E(x^2)+\sigma_{n_1}^2][E(y^2)+\sigma_{n_2}^2]}} \tag{A.14}$$

$$= \frac{\sqrt{pow(x)pow(y)}}{\sqrt{[pow(x)+pow(n_1)][pow(y)+pow(n_2)]}} \tag{A.15}$$

$$= \sqrt{\frac{SNR_x}{1+SNR_x}\frac{SNR_y}{1+SNR_y}}. \tag{A.16}$$