Multivariate Analysis of Raman Spectroscopy Data

Richard Haydock, BSc (Hons)

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy

November 2015

For my mum, Kathryn, who supported me through everything and always at least pretended to understand when I talked about my work.

Abstract

This thesis is concerned with developing techniques for analysing Raman spectroscopic images. A Raman spectroscopic image differs from a standard image as in place of red, green and blue quantities for each pixel a Raman image contains a spectrum of light intensities at each pixel. These spectra are used to identify the chemical components from which the image subject, for example a tablet, is comprised. The study of these types of images is known as chemometrics, with the majority of chemometric methods based on multivariate statistical and image analysis techniques.

The work in this thesis has two main foci. The first of these is on the spectral decomposition of a Raman image, the purpose of which is to identify the component chemicals and their concentrations. The standard method for this is to fit a bilinear model to the image where both parts of the model, representing components and concentrations, must be estimated. As the standard bilinear model is nonidentifiable in its solutions we investigate the range of possible solutions in the solution space with a random walk. We also derive an improved model for spectral decomposition, combining cluster analysis techniques and the standard bilinear model. For this purpose we apply the expectation maximisation algorithm on a Gaussian mixture model with bilinear means, to represent our spectra and concentrations. This reduces noise in the estimated chemical components by separating the Raman image subject from the background.

The second focus of this thesis is on the analysis of our spectral decomposition results. For testing the chemical components for uniform mixing we derive test statistics for identifying patterns in the image based on Minkowski measures, grey level co-occurence matrices and neighbouring pixel correlations. However with a non-identifiable model any hypothesis tests performed on the solutions will be specific to only that solution. Therefore to obtain conclusions for a range of solutions we combined our test statistics with our random walk. We also investigate the analysis of a time series of Raman images as the subject dissolved. Using models comprised of Gaussian cumulative distribution functions we are able to estimate the changes in concentration levels of dissolving tablets between the scan times. The results of which allowed us to describe the dissolution process in terms of the quantities of component chemicals.

Acknowledgements

I would first like to thank my supervisors, Chris Brignell and Simon Preston, who provided me with invaluable advice and support throughout my studies, without them this would not have been possible. I would also like to thank Jonathan Burley (School of Pharmacy) for gathering all my data and helping me with the chemistry behind my work.

I am also thankful for the support of the staff in the School of Mathematical Sciences, especially Dave Parkin who always fixed my computer when I broke it and without whom this thesis would likely be carved into stone tablets. I am also grateful to my fellow research students, in particular Nate, Rosanna, Cerian, Laurence and Emma for their friendship and comic relief.

Further mention goes to the University of Nottingham Archery Club who, throughout my years at university, contributed a small amount of stress but some great memories.

Finally my thanks go to all my family and Neira for their continued support and encouragement.

Contents

Notation convention

Lower case characters are used for scalars, bold and upper case characters are used to represent matrices and bold and lower case characters are used for vectors. This applies to both Latin and Greek characters.

Chapter 1

Introduction

1.1 Raman Spectroscopy

In this thesis we investigate methods for analysing Raman spectroscopic images. We can separate analysis of this data into two parts, methods for decomposing the data to identify component chemicals and their concentrations, and methods for analysing the decomposition results. Alongside investigating the existing methods we devise our own methods for more accurate decomposition and testing of the decomposition results. Most of the statistical techniques used for this analysis are well established in the field of mathematics however were only applied to chemical data from the late 1960s when they were given the name chemometrics [36].

Before describing the methods of analysis we first define a Raman spectroscopic image. Raman imaging is a specific type of spectroscopic imaging where Raman spectroscopy is used to gather the data [82]. Raman spectroscopy is a spectroscopic technique used to study the molecular vibrations of a scanned object. More specifically these vibrations are specific to chemical bonds and so can be used to identify individual chemicals present in the subject. The theory behind Raman spectroscopy was pioneered by George Placzek, a Czechoslovak physicist, between 1930 and 1934 using Raman scattering [88]. Raman scattering is an effect discovered by Sir C. V. Raman and K. S. Krishnan in 1928 for which Raman received a Nobel prize in 1930 [88].

Raman scattering is the inelastic scattering of photons. Inelastic scattering is fundamental scattering process in physics where, instead of the kinetic energy of a photon being preserved when it collides with an incident particle, some energy is lost or gained. It is this change that is measured in Raman spectroscopy. Molecules, the incident particles, can be in a ground state (minimum energy), vibrational state or virtual state (high energy). When a photon interacts with a molecule, that molecule gains energy and so is briefly elevated to the high

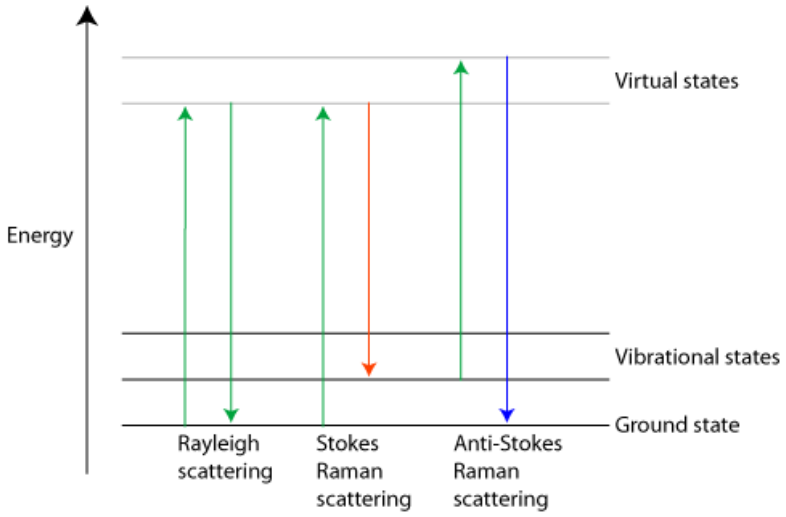


Figure 1.1: The different forms of photon scattering [88].

energy virtual state before a photon is re-emitted or reflected. The reflected molecule has three energy level possibilities [82], shown in figure 1.1, which can be described as follows

- The first energy level is an elastic process known as Rayleigh scattering. The result of Rayleigh scattering is a photon re-emitted from the molecule with the same energy as the initial photon. In this situation the molecule returns from the virtual state to the initial state. This type of scattering is far more common than the other two and provides no useful information for Raman spectroscopy. Therefore presenting the greatest problem in that it must be filtered out by the spectrometer.

- The second type of scattering is a form of inelastic Raman scattering. As with Rayleigh scattering the photon is absorbed by the molecule which is then elevated to a virtual state. However now the re-emitted photon has less energy than the incident photon and so the molecular vibrations fall back down to a higher vibrational state. This causes the photon to have a longer wavelength and is known as Stokes scattering.

- The third type of scattering is similar to the second however now the molecule begins in a vibrational state instead of a ground state. When the photon is re-emitted it has more energy causing the molecular vibrations to fall to a lower energy state. This causes the photon to have a shorter wavelength and is known as anti-Stokes scattering. It is also the least common type of scattering.

A Raman spectrometer is designed such that it can detect the three types of scattering and filter out the first and third type, the spectrometer used to gather the data in this thesis is the Horiba LabRAM HR seen in figure 1.2. The Raman spectrometer targets a

Figure 1.2: The Horiba LabRAM HR Raman spectrometer.

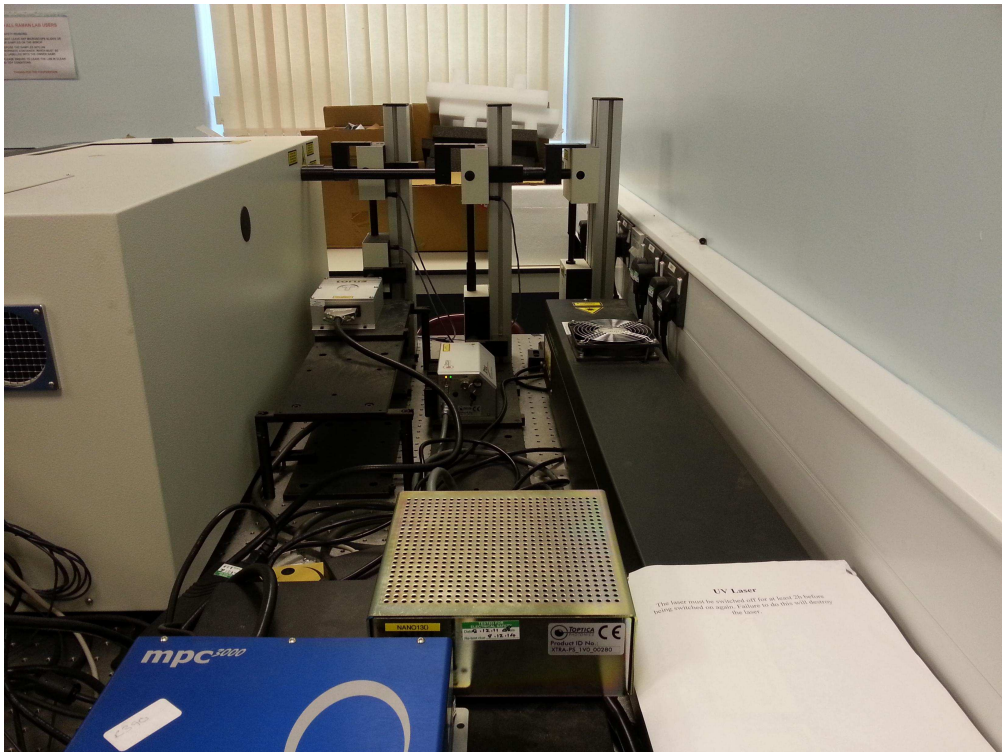


Figure 1.3: Horiba LabRAM HR lasers, red, green, ultraviolet and infra-red.

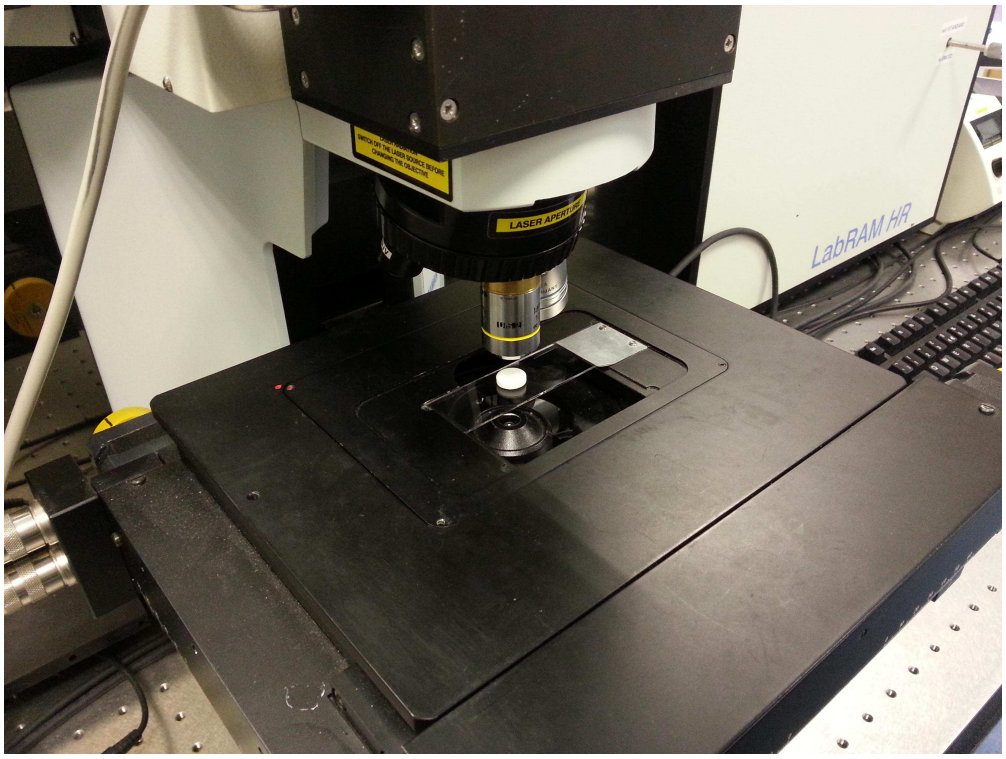


Figure 1.4: Horiba LabRAM HR laser aperture with tablet.

monochromatic light, usually a laser (as seen in figure 1.3), at a portion of the object, as seen in figure 1.4. The re-emitted photons from this laser, in the form of reflected light, are collected using a lens then sent through a monochromator which transmits only the required wavelengths [37]. For organic molecules this is the range 500-2000cm^{-1} [37]. At this stage in the process all three types of scattering are present. Therefore the Rayleigh scattering and anti-Stokes scattering are filtered out leaving only the Stokes scattered light to be passed onto a detector. This is the most challenging part of the process as Raman scattered light usually comprises only about 0.001% of the reflected light [82]. The detector then produces a Raman spectrum for the molecule using the change in energy from the initial to the final state. Each spectrum contains the intensity of the light at the selected set of wavenumbers. Figure 1.5 is an example of the Raman spectra for the chemical paracetamol plotted against the wavenumbers.

Spectroscopic imaging is a technique in which an object is divided into pixels and the spectrum at each pixel is measured using any one of a number of forms of spectroscopy [89]. We see an example of this in figure 1.6 where the grid represents the object divided into pixels and each pixel has a detected spectrum. This technique is useful in chemical analysis as it provides both spectral and spatial information relating to the scanned object. However the spectrum detected at each pixel is a combination of all component chemical spectra in the sample and must therefore be decomposed.

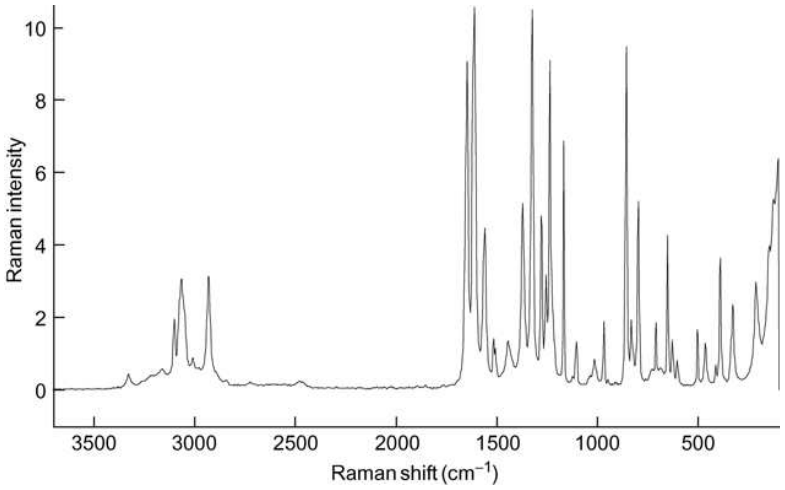


Figure 1.5: The Raman spectrum of paracetamol [75].

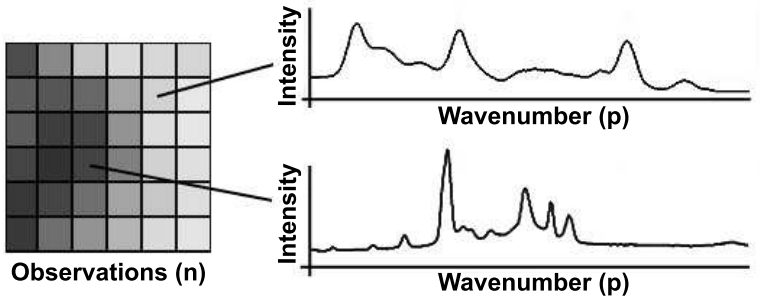


Figure 1.6: Graphical representation of Raman imaging.

The combination of spectroscopic imaging and Raman spectroscopy is useful for examining objects such as pharmaceutical tablets. With the measured spectra at each pixel of the tablet, investigation into the spread of chemical compounds throughout the tablet is possible. Raman spectroscopy has several benefits over other forms of spectroscopy. For example infra-red spectroscopy cannot be performed through glass or water and attenuated total reflectance (ATR) requires contact with the object, problematic if a powder or dissolving object is the subject.

1.2 Raman Spectroscopy Data and a Bilinear Model

To describe the form of a Raman spectroscopic image dataset we will define the general model used for decomposing a Raman image into component parts [62], the first process required in the analysis of a Raman image. A single pixel of a Raman image containing n pixels is an observation we denote as $\mathbf{y}_i$, a vector of dimension p, where p is the number of detected wavenumbers; examples of these are shown in figure 1.6. We assume that the detected spectrum at each pixel is a linear combination of the spectra of each component

chemical in the subject. Therefore we define the model as

$$\mathbf{y}_i = \mathbf{S}\mathbf{c}_i + \mathbf{e}_i \qquad \mathbf{e}_i \sim \mathcal{N}_p(\mathbf{0}, \sigma^2 \mathbf{I}_p) \quad \text{for } i = 1, ..., n \tag{1.2.1}$$

with

$$\overset{p\times r}{\mathbf{S}} = \begin{bmatrix} \mathbf{s}_1 & \dots & \mathbf{s}_r \end{bmatrix} \quad \text{and} \quad \overset{r\times 1}{\mathbf{c}_i} = \begin{bmatrix} c_1 \\ \vdots \\ c_r \end{bmatrix},$$

where the $\mathbf{s}_k$ are the r spectra corresponding to the component chemicals. The vector $\mathbf{c}_i$ contains the linear coefficients corresponding to the components of the $\mathbf{s}_k$ which we interpret as the concentrations of each component chemical. We include errors, $\mathbf{e}_i$, in this model as Raman spectroscopy is subject to interference from external sources causing noise in the data. We assume these errors are normal and independent, therefore $\text{cov}(\mathbf{e}_i, \mathbf{e}_{i'}) = \mathbf{0}$ for all pixels i and i', where $i \neq i'$. This normality assumption is commonly made as it is required by some methods of spectral decomposition and post-decomposition analysis. This assumption is also made as including covariance estimation in the maximum likelihood estimates, generalised least squares, does not improve results sufficiently to justify the increase in computation time and the higher number of parameters requiring estimation.

With the Raman image containing n pixels scanned over p wavenumbers we get an $n \times p$ data matrix. We can therefore write model (1.2.1) as

$$\mathbf{Y} = \mathbf{C}\mathbf{S}^T + \mathbf{E} \tag{1.2.2}$$

for

$$\overset{n\times p}{\mathbf{Y}} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_n^T \end{bmatrix}, \qquad \overset{n\times r}{\mathbf{C}} = \begin{bmatrix} \mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_n^T \end{bmatrix}, \qquad \overset{n\times p}{\mathbf{E}} = \begin{bmatrix} \mathbf{e}_1^T \\ \vdots \\ \mathbf{e}_n^T \end{bmatrix}, \qquad \mathbf{e}_i \sim \mathcal{N}_p(\mathbf{0}, \sigma^2 \mathbf{I}_p),$$

where we note that the spectra are common to all pixels in the Raman image but there are concentration parameters specific to each pixel. To decompose our Raman image into component spectra and their corresponding concentrations we now seek $\mathbf{S}$ and $\mathbf{C}$ such that

$$\underset{\mathbf{S},\mathbf{C}}{\text{argmin}} \left\| \mathbf{Y} - \mathbf{C}\mathbf{S}^T \right\|, \tag{1.2.3}$$

where $\|\cdot\|$ is an extension of the Euclidean norm to matrices, i.e.

$$\|\mathbf{X}\|^2 = \text{tr}\left(\mathbf{X}^T\mathbf{X}\right). \tag{1.2.4}$$

We must further restrict this model to ensure results can be estimated and any results we obtain are physically plausible from a chemistry perspective. To do this we impose three

constraints,

$$\begin{array}{lll} 1. & \mathbf{s}_k^T \mathbf{s}_k = 1 & \text{for all } k = 1, ..., r \\ 2. & s_{j,k} \geq 0 & \text{for all } j = 1, ..., p \text{ and } k = 1, ..., r \\ 3. & c_{i,k} \geq 0 & \text{for all } i = 1, ..., n \text{ and } k = 1, ..., r \end{array} \tag{1.2.5}$$

where $\mathbf{s}_k = \left(s_{1,k}, ..., s_{p,k}\right)^T$ and $\mathbf{c}_k = \left(c_{1,k}, ..., c_{n,k}\right)^T$. The first of these limits the non-identifiability of (1.2.3) however does not remove it entirely. The second and third enforce non-negativity as a chemical spectrum should represent light intensities at each wavenumber and a concentration represents the quantity of a component chemical at pixel i, therefore neither should be negative. We also note that in the situation where σ, in (1.2.2), is sufficiently large constraints 2 and 3 may not be entirely possible. In practice however this does not cause an issue as the non-negatively constraints are usually not enforced as hard constraints and so some small violations are allowed. This is true when using non-negative least squares, §2.6.2, as we do for the majority of this thesis. If the measurement error, σ, is too large and scales with the measured values log transforming the data may prove useful in reducing this and making constraints 2 and 3 more easy to enforce.

There are a number of methods available for solving (1.2.3) however they can be split into two groups. The first, including the original method proposed by Lawton and Sylvestre [62], uses eigendecomposition followed by steps to impose the constraints. The second group uses least squares solutions including a non-negative least squares variant [61]. However no method will have a unique solution due to non-identifiability of the model, to be detailed in §1.2.1. It is also important to note that in the field of chemometrics the process of solving (1.2.3) subject to (1.2.5) is known as spectral decomposition, in other words identifying $\mathbf{S}$ and $\mathbf{C}$. Therefore in this thesis the term *spectral decomposition* will always relate to the solving of (1.2.3) rather than eigendecomposition.

Throughout this thesis we display the results from this problem in two ways. The spectra are contained in the columns of $\mathbf{S}$ and will be plotted against a vector of wavenumbers to give us component spectra, an example of which is seen in figure 1.5. To illustrate the concentrations for component chemical k at each pixel we take the k^{th} column of the matrix $\mathbf{C}$, defined as

$$\overset{n\times 1}{\mathbf{v}} = \begin{bmatrix} c_{1,k} \\ \vdots \\ c_{n,k} \end{bmatrix}$$

and reshape this vector into a matrix with the dimensions of the original rectangular Raman

image, say $x \times y$, giving

$$\overset{x\times y}{\mathbf{V}} = \begin{bmatrix} v_1 & \dots & v_{(y-1)x+1} \\ \vdots & \ddots & \vdots \\ v_x & \dots & v_n \end{bmatrix}.$$

From this matrix we can create a heatmap of the levels of component chemical k at each pixel.

1.2.1 Model Non-identifiability

We say our model, (1.2.3) is non-identifiable as the solutions are non-unique. For example, suppose $\left(\hat{\mathbf{C}}, \hat{\mathbf{S}}\right)$ are solutions to (1.2.3) subject to (1.2.5), for some invertible matrices $\mathbf{R}$, from a set of matrices subject to constraints (1.2.5), we can find a range of solutions to (1.2.3) by writing

$$\mathbf{Y} = \hat{\mathbf{C}}\hat{\mathbf{S}}^T + \mathbf{E} = \hat{\mathbf{C}}\mathbf{R}^{-1}\mathbf{R}\hat{\mathbf{S}}^T + \mathbf{E} = \tilde{\mathbf{C}}\tilde{\mathbf{S}}^T + \mathbf{E}$$

where $\tilde{\mathbf{C}} = \hat{\mathbf{C}}\mathbf{R}^{-1}$ and $\tilde{\mathbf{S}}^T = \mathbf{R}\hat{\mathbf{S}}^T$. It is clear that $\mathbf{Y} - \tilde{\mathbf{C}}\tilde{\mathbf{S}}^T = \mathbf{Y} - \mathbf{C}\mathbf{S}^T$ and so both are solutions to (1.2.3). However from a chemical perspective they are different solutions as $\hat{\mathbf{S}}$ may not show the same chemical spectra as $\tilde{\mathbf{S}}$. Whilst this does not effect our optimisation of (1.2.3) in a mathematical sense it does effect our physical interpretation of the results. For some model sizes, for example if $r = 1$, then $R = 1$ and the model is identifiable subject to (1.2.5) as $R = -1$ violates (1.2.5), constraints 2 and 3. Some methods of spectral decomposition, such as those detailed in §2.5, make use of this benefit from dimension reduction. The implications of this non-uniqueness, more specifically that caused by non-identifiability of the solutions, will be dealt with in Chapter 4 and Chapter 5.

In the field of chemometrics this is known as *rotational ambiguity* [12, 53] in spite of the fact that $\mathbf{R}$ is not necessarily a rotation matrix but could be a more general non-singular transformation matrix such that constraints (1.2.5) are still met.

1.2.2 Raman Spectroscopy Datasets

Throughout this thesis we use a number of datasets, the majority of which are Raman images of pharmaceutical tablets. Our first dataset, and the smallest, is taken from Lawton and Sylvestre [62] and is five observations over 30 wavenumbers of an unspecified material comprised of two chemical components. This gives us a 5×30 data matrix. For some of our investigations we will also use this dataset solving for 3 spectra, $r = 3$. Whilst not chemically sensible the simple nature of the dataset is all we require when testing methods

such as those investigated in Chapter 4, therefore the nature of the third estimated spectra is of no importance.

Another small dataset, containing a single component spectrum, is a 70×75 pixel scan of a paracetamol tablet over 1600 wavenumbers resulting in a 5250×1600 data matrix. Our third, and slightly larger, dataset is a 269×57 pixel scan of a human tooth over 1024 wavenumbers. This contains enamel and dentine component spectra. Due to the very small size of Lawton's dataset we will use that when testing new methods and when investigating the original method proposed by Lawton and Sylvestre [62] as the true solutions are known and there is very little noise in the data. We will use the paracetamol dataset on new methods as a single component spectrum is present and so solving (1.2.3) is very simple. The dataset also contains at least one example of the most common anomalies so we will use it extensively in our testing of pre-processing methods in Chapter 3. Finally the human tooth dataset contains two spectra which are very similar in structure and so we use this as a more rigorous test of new methods to ensure they can detect similar spectra.

We also have a large dataset containing 12 scans of tablets of bicalutamide, a drug commonly used in the treatment of prostate cancer, and copovidone, a filler and binder. The scans in this set are 86×86 pixel scans over 1024 wavenumbers giving a 7396×1024 data matrix for each scan. There are six tablets in total with the upper and lower faces of each scanned giving the 12 data matrices. The six tablets are also split into two groups with different manufacturing processes. Three are made by mixing dry components and forming them into a tablet with a binder whilst the other three have their component chemicals melted, mixed and then extruded into the tablet, in theory providing a more thorough mix. We will use these datasets for hypothesis testing the results of spectral decomposition as we have repeated scans of each tablet and of tablets which will ideally be almost identical. We will also use the different production methods in devising tests for the quality of mixing of component chemicals in a tablet.

Our final two datasets are images recorded at discrete times as the scan subject dissolves. The first is nine 73×21 scans, over 1600 wavenumbers, of a caffeine tablet dissolving over 211 minutes in water; this results in a 13797×1600 data matrix comprised of nine 1533×1600 individual data matrices for each time point. This is expected to contain caffeine and caffeine hydrate. The second dataset is fifteen 34×34 scans, over 1024 wavenumbers, of a bicalutamide tablet dissolving over 2131 minutes. We therefore have a 17340×1024 data matrix comprised of fifteen 1156×1024 individual data matrices. This data is more complicated than the caffeine dataset as the initial component, the extrudate, dissolves into either bicalutamide type I or II. Bicalutamide type II then becomes bicalutamide type I as the process continues. We will use these datasets for modelling how

a tablet dissolves over time with the objective of predicting the dissolution pattern between the discrete scan times. As the caffeine tablet has concentrations of one component decreasing whilst the other increases we will use it for the initial testing of a model. We will then use the more complicated bicalutamide tablet for a more rigorous investigation of the proposed method.

An important point to note when we fit (1.2.2) to these datasets is that we will sometimes set r, the number of component chemicals, to be one greater than the number of component chemicals we expect the subject to contain. We do this with Raman images containing a large number of background pixels, that is pixels representing only the scan bed not the scan subject. In these situations we include an extra $\mathbf{s}_k$ to represent this background signal as it will account for variance in the data. Selecting the correct number of component spectra to explain the majority variance is vital to obtaining the most accurate estimates as detailed in §2.2.

1.3 Thesis Outline

The main contributions to the field of chemometrics in this thesis are contained in Chapters 4 to 7. In Chapters 4 and 5 we apply existing statistical techniques, such as random walks and bootstrapping, in novel ways to Raman spectroscopy data in order to perform hypothesis tests on estimators of (1.2.2). In Chapter 6 we define a new model for specifically for Raman spectroscopy data with a time component. With this model more accurate analyses and visualisations of dissolution processes are possible. In Chapter 7 we propose a new method for finding solutions to (1.2.2) by incorporating a clustering technique. This method provides some of the benefits of unique resolution methods, to be discussed in §2.5, such as dimension reduction in the estimates without requiring used experience.

In Chapter 2 we outline the more frequently used mathematical techniques in this thesis. These are split into the methods used for solving (1.2.3), which are either based on principal component analysis or least squares optimisation, and methods used for analysing the results of spectral decomposition of a dataset. We also provide a brief comparison of the more common techniques for solving (1.2.3).

The focus of Chapter 3 is the pre-processing of Raman spectroscopy data. Our objectives in pre-processing the data are to remove anomalies introduced in the data acquisition stage or caused by the scanned object. We devise and explore techniques for adjusting the data in order to improve the clarity of results. The investigation of anomaly removal methods involves existing statistical techniques such as a Windsorised mean and variance scaling. For

data adjustment we modify a standard linear model incorporating ideas from the quadratic penalty method of numerical optimisation. We also work on a method for adjusting small differences in spectrometer calibration in the data so it can be easily compared to other datasets. Finally we consider the use of a robust principal component analysis algorithm in a process called chemical rank analysis, used in estimating the number of component spectra in a dataset.

In Chapter 4 we look at building on the work of Lawton and Sylvestre [62] who originally proposed (1.2.2) for decomposing a dataset into a linear combination of two component spectra. Our aim is to extend their method to data containing more than two component chemicals. We will first prove the link between the objective functions of the eigendecomposition based approach of Lawton and the least squares optimisation performed by the commonly used modern methods. With this link shown our extension of Lawton's solution space provides a useful tool for exploring the solutions to the bilinear model (1.2.2). Analytically defining the solution space used by Lawton is possible up to three dimensions however further expansion requires numerical methods with only partial analytical bounds. With this knowledge of the solution space in higher dimensions we then propose a random walk to explore this region and give us a greater understanding of the range of possible solutions least squares based methods can obtain. This random walk is useful when hypothesis testing the solutions estimated with the MCR-ALS least squares method in Chapter 5.

Throughout Chapter 5 we concentrate on the ways in which we can conduct hypothesis tests on Raman spectroscopy data. The aims of our hypothesis tests are to detect a difference in the quantities of chemical components between similar tablets and to test how well mixed the chemical components are in a scan subject. We first describe some standard statistical methods such as Hotelling's T^2 test and the principle of bootstrapping. With these defined we initially conduct a simple T^2 test on the estimated concentration matrices for multiple Raman images of tablets containing the same component chemicals. With non-identifiability in our model having a negative effect on the reliability of our tests we then incorporate the random walk derived in Chapter 4 to construct a more robust test.

The second area we investigate is how to test the mixing of chemicals in a Raman image, for example how thoroughly mixed component chemicals in a tablet are. Before we can test this we derive test statistics for quantifying how uniformly mixed the estimated concentrations of a component chemical are. Using Minkowski functionals, grey level co-occurrence matrices and Pearson's correlation coefficient we suggest and test three potential test statistics. As with the T^2 test we also must deal with the non-identifiability of the solutions and so incorporate the random walk so we can test a range of possible solutions. We finally test the combination of our new test statistics and the random walk using simulated data and

real world images of tablets containing the same chemicals yet produced using different techniques. With a reliable algorithm we aim to provide a non-destructive test of how well mixed a tablet is for the purpose of quality control.

In Chapter 6 we work with Raman images gathered over time as the scan subject dissolved. Our aim is to devise a model capable of predicting how the component chemical concentrations change in between the discrete scans and even after the final scan. Using a two part model we map the dissolution of a simple dataset with one component decreasing and the other increasing in concentration. For our more complicated data we require a model capable of fitting to concentrations which increase, decrease or increase and decrease. We achieve this by combining the separate models used on our simple dataset to form a continuous model which fits all combinations of concentration change. With our new model we are able to estimate continuous dissolution times from the discrete Raman images.

For our final area of research in Chapter 7 we investigate modifications of our bilinear model (1.2.2). With the scan bed appearing around the scan subject in almost all Raman images we use cluster analysis to split the dataset into pixels containing either the background or the subject. The motivation behind this is to decrease the level of background signal in the dataset to improve the clarity of estimated spectra and concentrations. In testing several clustering methods we conclude that a Gaussian mixture model produces the most accurate clusters when fit with the expectation-maximisation algorithm, defined in §2.12.1. We therefore fit a bipartite bilinear model using the clustering results to obtain estimates of the component spectra displaying less noise. In our dataset containing component chemicals with very similar spectra we also saw more reliable detection of both spectra compared to our results using the unclustered data. With our concept proving successful we propose a single stage algorithm to both cluster and perform spectral decomposition. To achieve this we combine our bipartite bilinear model with the Gaussian mixture model and recalculate the steps of the expectation-maximisation algorithm. Whilst the results we obtain from this algorithm match those from the two stage process we see a reduction in computation time with no user input required between clustering and fitting the model.

Finally in Chapter 8 we discuss our results from the previous chapters and provide areas in which future work could be done to further this research and benefit the field of chemometrics.

Chapter 2

Literature Review

2.1 Introduction

In this chapter we will discuss and define the multivariate methods and dimension reduction techniques commonly used in this thesis. In §1.2 we define the model we wish to solve for most Raman images,

$$\mathbf{Y} = \mathbf{C}\mathbf{S}^T + \mathbf{E} \tag{1.2.2}$$

with solutions

$$\underset{\mathrm{S,C}}{\operatorname{argmin}} \left\| \mathbf{Y} - \mathbf{C}\mathbf{S}^T \right\|$$

subject to

$$\begin{array}{lll} 1. & \mathbf{s}_k^T \mathbf{s}_k = 1 & \text{for all } k = 1, ..., r \\ 2. & s_{j,k} \geq 0 & \text{for all } j = 1, ..., p \text{ and } k = 1, ..., r, \\ 3. & c_{i,k} \geq 0 & \text{for all } i = 1, ..., n \text{ and } k = 1, ..., r. \end{array} \tag{1.2.5}$$

We can separate the methods for solving (1.2.2) into two groups. The first group indirectly solves (1.2.2), and is the basis of the very first method for spectral decomposition [36]. This technique, to be discussed in §2.4 and §2.5, first finds a solution to

$$\mathbf{Y} = \mathbf{G}\mathbf{H}^T + \mathbf{E} \tag{2.1.1}$$

by finding

$$\underset{\mathrm{H,G}}{\operatorname{argmin}} \left\| \mathbf{Y} - \mathbf{G}\mathbf{H}^T \right\|$$

for some $n \times r$ matrix $\mathbf{G}$ and $p \times r$ matrix $\mathbf{H}$

$$\overset{p \times r}{\mathbf{H}} = \begin{bmatrix} \mathbf{h}_1 & \dots & \mathbf{h}_r \end{bmatrix}$$

with the constraints that

$$\begin{array}{lll} 1. & \|\mathbf{h}_k\| = 1 & \text{for all } k = 1, ..., r \\ 2. & \mathbf{h}_k^T \mathbf{h}_{k'} = 0 & \text{for all } k = 1, ..., r \text{ and } k' = 1, ..., r \text{ such that } k \neq k' \end{array} \tag{2.1.2}$$

where r is, once again, the number of component spectra in the dataset. Linear combinations of the $\mathbf{h}_k$ then construct the $\mathbf{s}_k$

$$\mathbf{S} = \mathbf{HR}$$

and $\mathbf{c}_i$

$$\mathbf{C} = \mathbf{G}\left(\mathbf{R}^{-1}\right)^T$$

such that they minimise (1.2.2) and still satisfy constraints (1.2.5). The situation may arrise when constraints (1.2.5)(2,3) cannot be satisfied and so will likely require relaxing. As previously mentioned these constraints are often not strictly imposed so this should cause little in the way of problems.

The second group of methods directly solves (1.2.2) subject to (1.2.5), usually using non-negative matrix factorisations, and currently forms the basis of the most common methods [89]. In the field of chemometrics the methods using the indirect technique of finding a solution to (1.2.2) are known as unique resolution methods. The methods using the direct approach are known as rational resolution methods. Unique resolution methods, an example of which is defined in Chapter 7.2, are usually more complicated to implement as they require detailed knowledge of the Raman image to be analysed and therefore require an experienced analyst. The detailed knowledge is usually around the structure of the scanned object allowing the analyst to select regions containing a single component chemical, reducing the dimension of (1.2.2) and removing the non-uniqueness of solutions. In comparison rational resolution methods require little experience and although the solutions may not be as clear the ease of implementation often outweighs any improved clarity in the results [52, 53]. Regardless of the method they are all now known as self-modelling curve resolution, the name originally given to the first method proposed for solving (1.2.2) [62].

In this chapter we will also define other commonly used methods in this thesis such as forms of cluster analysis and a technique for estimating the number of component spectra in a dataset.

2.2 Chemical Rank Analysis

Whatever chemometric method we select for solving (1.2.2) we must first know the number of chemical components in a dataset, also known as the chemical rank. For the data used in this thesis we know the exact components to expect however this may not always be the case. In these situations a technique known as chemical rank analysis should be used. The method was first used by Wernimont [99] and has become the usual method of estimating chemical rank [53].

Chemical rank analysis estimates the number of component spectra in a dataset using principal component analysis on mean centred data. The theory behind this process is that the spectra act as bases for the data as each observation is a linear combination of the component spectra. These spectra usually account for more variance than the standard subspace and so the number of bases identified by PCA as explaining the majority of variance should coincide with the number of component spectra in the data as these are the true bases [32]. Given a set of eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$ we calculate the vector, $\mathbf{l}$, of cumulative sums

$$l_m = \sum_{j=1}^{m} \frac{\lambda_j}{\sum_{J=1}^{p} \lambda_J} \tag{2.2.1}$$

where $\frac{\lambda_j}{\sum_{J=1}^{p} \lambda_J}$ is the proportion of variance explained by eigenvector j [73, §8.2.3]. We therefore look at the point where vector $\mathbf{l}$ begins to plateau and select that as the estimated number of component chemicals as including further eigenvectors will provide a minimal increase in explained variance. This method is not perfect however as similar chemical spectra can appear as a single eigenvector leading us to believe one fewer chemical component is present.

2.3 Principal Component Analysis

To use the first technique for solving (1.2.2) subject to (1.2.5) we must first solve (2.1.1) subject to (2.1.2). We can interpret (2.1.1) as defining the $\mathbf{y}_i$ as a linear combination of orthonormal vectors $\mathbf{h}_k$. We may therefore be able to use principal component analysis to find these orthonormal vectors. We prove in §4.2 that under the constraints (2.1.2) the principal component loadings are indeed least squares minimisers of (2.1.1) and can thus be used in the first step of the solution process.

Principal component analysis (PCA) is a procedure for generating a new set of basis vectors with the objective of reducing the number of bases required to explain the majority of

variance in the data. PCA is usually applied to mean centred data which, in the terminology of this thesis involves setting

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i - \frac{1}{n}\sum_{j=1}^{p} y_{i,j} \tag{2.3.1}$$

for all $i = 1, ..., n$ to form mean centred data matrix $\tilde{\mathbf{Y}}$ of the same form as described in §1.2. To perform PCA and find new bases for the data we first calculate the covariance matrix $\tilde{\mathbf{A}} = \frac{\tilde{\mathbf{Y}}^T\tilde{\mathbf{Y}}}{n}$ [1]. We then find matrices $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{K}}$ such that

$$\tilde{\mathbf{A}} = \tilde{\mathbf{K}}\tilde{\mathbf{L}}\tilde{\mathbf{K}}^T$$

where

$$\overset{p\times p}{\tilde{\mathbf{L}}} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_p \end{bmatrix} \quad \text{and} \quad \overset{p\times p}{\tilde{\mathbf{K}}} = \begin{bmatrix} \mathbf{k}_1 & \dots & \mathbf{k}_p \end{bmatrix}$$

for eigenvectors, or principal component loadings, $\mathbf{k}_j$ ordered corresponding to eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$ with

$$\|\mathbf{k}_j\| = 1 \quad \text{for } j = 1, ..., p$$

and

$$\mathbf{k}_j^T\mathbf{k}_{j'} = 0 \quad \text{for all } j = 1, ..., p \text{ and } j' = 1, ..., p \text{ such that } j \neq j'.$$

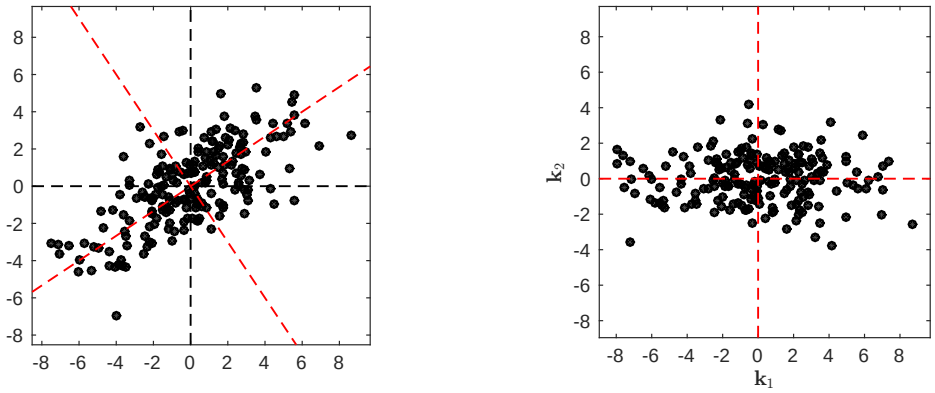
The result of this is that we can decompose the data into two matrices such that

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{W}}\tilde{\mathbf{K}}^T \tag{2.3.2}$$

where $\tilde{\mathbf{W}} = \tilde{\mathbf{Y}}\tilde{\mathbf{K}}$ are the principal component scores, the data projected onto the new bases. From the properties of PCA the amount of variance explained by eigenvector j is proportional to $\left(\lambda_j / \sum_{J=1}^{p} \lambda_J\right)$ [73, §8.2.3][79]. Therefore we can use the eigenvectors explaining the majority of the variance as the new basis vectors. With these new bases we can reduce the dimension of the data with little loss of information as the later eigenvectors should explain a minimal amount of variance.

Algorithm 2.3.1. Principal Component Analysis Dimension Reduction

1. Calculate the covariance matrix $\tilde{\mathbf{A}} = \frac{\tilde{\mathbf{Y}}^T\tilde{\mathbf{Y}}}{n}$.
2. Calculate the eigenvectors, $\tilde{\mathbf{K}}$, from $\tilde{\mathbf{A}}$.



(a) Standard bases. **(b)** New bases $\mathbf{k}_1$ and $\mathbf{k}_2$.

Figure 2.1: An example of new bases, shown in red, compared to standard bases, shown in black, derived by PCA for random data.

3. Calculate cumulative sum vector $\mathbf{l}$

$$l_m = \sum_{j=1}^{m} \frac{\lambda_j}{\sum_{J=1}^{p} \lambda_J} \tag{2.3.3}$$

and identify the smallest m such that $l_m \geq 0.95$.

4. Project the data onto new bases using

$$\tilde{\mathbf{Y}}' = \tilde{\mathbf{Y}}\tilde{\mathbf{K}}'.$$

for

$$\tilde{\mathbf{K}}' = \begin{bmatrix} \mathbf{k}_1 & \dots & \mathbf{k}_m \end{bmatrix}.$$

This process is equivalent to transforming data from the standard basis vectors to new basis vectors equal to the eigenvectors, $\{\mathbf{k}_1, \mathbf{k}_2, ..., \mathbf{k}_m\}$. A transformation easily visualised in 2-dimensions as shown in figure 2.1.

When dealing with Raman spectroscopy data, and the methods of analysis to be used in this thesis, dimension reduction is not of interest to us. This is because it will result in fewer variables which will be a linear combination of the original wavelengths and thus will be harder to interpret. It is the new bases found by PCA which are of use with Raman spectroscopic images. As seen in figure 2.1b the new bases closely follow the data structure, therefore with data which is a linear combination of r spectra the bases explaining the majority of the variance will likely resemble the spectra [32]. It is this property that was used in the original method of solving (1.2.2), to be detailed in §2.4.

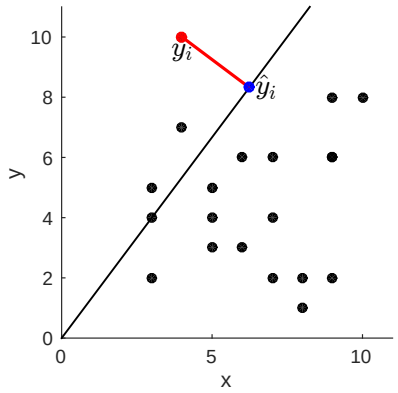


Figure 2.2: Optimisation performed by PCA on uncentred data for the primary eigenvector in 2-dimensions The red point denotes $\mathbf{y}_i$ and the blue point $\mathbf{K}\mathbf{w}_i = \hat{\mathbf{y}}_i$.

2.3.1 Existence of a Single Eigenvector of Constant Sign

Lawton and Sylvestre [62] made one key assumption on the results of PCA when applied to an entirely positive matrix $\mathbf{Y}$. The assumption was that the eigenvector corresponding to the largest eigenvalue, i.e. the first principal component, will have components all of the same sign. As we will be using Lawton's results in Chapter 4 we will prove this assumption for data which is uncentred due to the centring process creating negative values. To the best of our knowledge this proof has not been published or referenced in chemometric works.

As stated by Hung et al. [49], principal component analysis minimises the objective function

$$\min_{\mathbf{K}} \sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{K}\mathbf{w}_i\|^2 \tag{2.3.4}$$

for eigenvectors $\mathbf{K}$ of uncentred data $\mathbf{Y}$, defined previously, and scores $\mathbf{w}_i = \mathbf{K}\mathbf{y}_i$. In 2-dimensions we can visualise this as minimising the perpendicular distance between the eigenvectors and the data points, as shown in figure 2.2. We now define

$$\sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{K}\mathbf{w}_i\|^2 = \sum_{i=1}^{n} d_i^2 \tag{2.3.5}$$

for Euclidean norm $\|\cdot\|$.

As PCA is a dimension reduction technique we wish to reduce the number of $\mathbf{k}_k$ therefore, with the definition $\mathbf{w}_i = \mathbf{K}\mathbf{y}_i$, we define $\hat{\mathbf{y}}_i$ as

$$\hat{\mathbf{y}}_i = \mathbf{k}\mathbf{k}^T\mathbf{y}_i$$

for single eigenvector $\mathbf{k}$, giving

$$\begin{aligned}\boldsymbol{\varepsilon}_i &= \mathbf{y}_i - \mathbf{k}\mathbf{k}^T\mathbf{y}_i \\ &= \left(\mathbf{I}_p - \mathbf{k}\mathbf{k}^T\right)\mathbf{y}_i\end{aligned}$$

and thus

$$d_i^2 = \boldsymbol{\varepsilon}_i^T \boldsymbol{\varepsilon}_i.$$

We now show that, for entirely positive data $\mathbf{Y}$, $\sum_{i=1}^{n} d_i^2$ is minimised by an eigenvector, $\mathbf{k}$, with all components of the same sign. By considering the data matrix as a whole rather than by each observation we see

$$\mathbf{E}^T = \left(\mathbf{I}_p - \mathbf{k}\mathbf{k}^T\right)\mathbf{Y}^T$$

and

$$\begin{aligned} d^2 &= \text{tr}\left(\mathbf{E}^T\mathbf{E}\right) \\ &= \text{tr}\left(\mathbf{Y}\left(\mathbf{I}_p - \mathbf{k}\mathbf{k}^T\right)\mathbf{Y}^T\right) \\ &= \text{tr}\left(\mathbf{Y}\mathbf{Y}^T\right) - \text{tr}\left(\mathbf{Y}^T\mathbf{Y}\mathbf{k}\mathbf{k}^T\right) \end{aligned} \tag{2.3.6}$$

using the linearity and cyclic invariance properties of trace and the idempotency of $\mathbf{k}\mathbf{k}^T$. Minimising (2.3.6) is now a case of maximising $\text{tr}\left(\mathbf{Y}^T\mathbf{Y}\mathbf{k}\mathbf{k}^T\right)$. We write each diagonal element of $\mathbf{Q} = \mathbf{Y}^T\mathbf{Y}\mathbf{k}\mathbf{k}^T$ as

$$\mathbf{Q}_{l,l} = \sum_{j=1}^{p}\left[k_j k_l \sum_{i=1}^{n} y_{i,j}y_{i,l}\right], \tag{2.3.7}$$

whose trace we must maximise.

From Mardia et al. [73, Example 8.2.4 (d)], (2.3.6) is minimised when $\mathbf{k}$ is the eigenvector corresponding to the largest eigenvalue of $\mathbf{Y}^T\mathbf{Y}$ as

$$\begin{aligned} \text{E}\left[d^2\right] &= \text{tr}\left(\mathbf{Y}\mathbf{Y}^T\right) - \sum_{j=1}^{p}\left[k_j k_l \lambda\right] \\ &= \text{tr}\left(\mathbf{Y}\mathbf{Y}^T\right) - \lambda, \end{aligned}$$

which is therefore minimised when λ is the largest eigenvalue which corresponds to the first eigenvector. To show all elements of this eigenvector are of the same sign we let the primary eigenvector of $\mathbf{Y}^T\mathbf{Y}$ be $\bar{\mathbf{k}}$ with some $\bar{k}_m < 0$ and $\bar{k}_j > 0$ for $j \neq m$. With $\mathbf{Y}$ entirely positive we know

$$\sum_{i=1}^{n} y_{i,j}y_{i,l} > 0 \quad \text{for } j = 1, \ldots, p.$$

Therefore

$$\sum_{j=1}^{p}\left[\bar{k}_j\bar{k}_m\sum_{i=1}^{n} y_{i,j}y_{i,m}\right] < \sum_{j=1}^{p}\left[\hat{k}_j\hat{k}_m\sum_{i=1}^{n} y_{i,j}y_{i,m}\right]$$

for some $\hat{k}_j = |\bar{k}_j|$ for all j and our true maximiser of (2.3.7) is therefore an eigenvector with components of the same sign.

2.3.2 Uncentred Principal Component Analysis

The only area of PCA which may cause problems when applied to Raman spectroscopy data is the mean centring of the observations, (2.3.1). As detailed in §1.2 Raman spectroscopy data is almost entirely positive as it represents the intensity of light at specific wavelengths, a value which, from a physical standpoint, cannot be negative. However mean centring the data will almost certainly result in negative elements in each observation. Therefore the eigenvectors calculated as new bases for the data, representing the component spectra, may not be physically sensible solutions.

As such we may be required to perform uncentred principal component analysis on our data $\mathbf{Y}$. This simply involves calculating the eigenvectors and eigenvalues of $\frac{\mathbf{Y}^T\mathbf{Y}}{n}$ which, as shown by Cadima and Jolliffe [16], has no effect on the method or the objective function. Our results should therefore still represent component spectra and satisfy our need for r vectors which are a solution to (2.1.1), proven in §4.2, that can be transformed into solutions to (1.2.2).

2.3.3 Robust Principal Component Analysis

Principal component analysis can suffer when the data contains anomalies large enough to effect the eigenvectors. To negate the effects of these anomalies we can implement robust principal component analysis, a technique suggested by Campbell [17]. The robust principal component analysis algorithm takes the standard eigenvectors and eigenvalues and iteratively recalculates them whilst downweighting anomalies that have too large an effect on the estimated covariance matrix. This algorithm therefore calculates the eigenvectors and the corresponding eigenvalues on a subset of the data[17]. Aside from being robust to anomalies in the data a further advantage of this algorithm is that we can halt it when the required number of eigenvectors have been estimated. This may prove useful for large Raman spectroscopy datasets as not all the eigenvectors will be required and computing all p may be computationally expensive since p typically exceeds 1000.

2.4 Lawton's Method for Self Modelling Curve Resolution

The first method for decomposing a spectroscopic image, initially known as self modelling curve resolution, was proposed by Lawton and Sylvestre [62] and was the method which began the field of chemometrics [36]. This method solves (1.2.2), with $r = 2$, by first finding a solution to (2.1.1) using uncentred PCA, a solution we prove in §4.2. The

solutions to (2.1.1) are then transformed, as in §1.2.1, to comply with constraints (1.2.5) instead of (2.1.2) and thus form a solution to (1.2.2).

SMCR was originally devised to determine the shapes of two overlapping functions, $s_1(x)$ and $s_2(x)$, from a set of additive mixtures of these two functions.

$$y(\omega) = c_1 s_1(\omega) + c_2 s_2(\omega), \qquad \Omega_1 \leq \omega \leq \Omega_2 \tag{2.4.1}$$

where $y(\omega)$ is a continuous curve. In (2.4.1) the ω are the wavenumbers over which each observation is scanned, the s_k are the component spectra from which the observation is comprised and the c_k are the concentrations of each component spectra.

When fitting (2.4.1) Lawton defined three assumptions.

1. The $s_k(\omega)$ are normalised so that
$$\int_{\Omega_1}^{\Omega_2} s_k(\omega) = 1.$$
2. The $s_k(\omega)$ are non-negative and linearly independent.
3. The c_k are non-negative.

Although written as functions these three assumptions are clearly equivalent to (1.2.5). The last two of these ensure the solutions to (2.4.1) are physically and chemically plausible, as in (1.2.5). The first is a mathematical constraint to ensure we can identify the $s_k(\omega)$. If these assumptions are satisfied and there is no region in $[\Omega_1, \Omega_2]$ in which both $s_1(x)$ and $s_2(x)$ are simultaneously zero then we can construct $s_1(x)$ and $s_2(x)$ satisfying (2.4.1).

The idea behind the method is to find two functions to act as bases which span the data and can be combined to form any observation. As an initial estimate for these two functions Lawton used the first two eigenvectors of the data. Eigenvectors are a sensible starting point as they explain the greatest proportion of the variance, entirely span the data and solve (2.1.1) as we will prove in §4.2. Aside from the first eigenvector however they do not satisfy the non-negativity requirement for the basis functions, shown in §2.3.1. Lawton therefore used the non-identifiability in the model to transform these eigenvectors into physically plausible solutions, see §1.2.1.

In practice, when working with a data matrix $\mathbf{Y}$, it is convenient to discretise (2.4.1) giving

$$\mathbf{y}_i = c_{i,1}\mathbf{s}_1 + c_{i,2}\mathbf{s}_2 + \mathbf{e}_i = \mathbf{S}\mathbf{c}_i + \mathbf{e}_i, \qquad \text{for } i = 1, ..., n \tag{2.4.2}$$

where $\mathbf{y}_i = (y_i(\omega_1), ..., y_i(\omega_p))^T$, $\mathbf{s}_k = (s_k(\omega_1), ..., s_k(\omega_p))^T$ and $\mathbf{e}_i$ are the experimental errors for the i^{th} observation. This model is clearly equal to (1.2.1) in §1.2 with $r = 2$. All

three assumptions of (2.4.1) remain and the limit $\Omega_1 \leq \omega \leq \Omega_2$ is replaced with a known set of discrete ω_j for $j = 1, ..., p$. The unit area restriction is also discretised and becomes

$$\sum_{j=1}^{p} s_k(\omega_j)\Delta_j = 1$$

with

$$\Delta_j = \begin{cases} |\omega_2 - \omega_1| & \text{if } j = 1 \\ |\omega_p - \omega_{p-1}| & \text{if } j = p \\ |\frac{\omega_k - \omega_{k-1}}{2}| & \text{if } 1 < j < p. \end{cases}$$

Providing the two spectra assumption made by Lawton holds it is also possible to write $\mathbf{y}_i$ as

$$\mathbf{y}_i = \kappa_{i,1}\mathbf{k}_1 + \kappa_{i,2}\mathbf{k}_2 + \boldsymbol{\varepsilon}_i \tag{2.4.3}$$

where the $\kappa_{i,k}$ are linear combination coefficients and $\mathbf{k}_1$ and $\mathbf{k}_2$ are the first two eigenvectors of the second moment matrix $\mathbf{M} = \frac{\mathbf{Y}^T\mathbf{Y}}{n}$ calculated using uncentred PCA as in §2.3. The definition of (2.4.3) uses the same property of eigenvectors as chemical rank analysis, §2.2, in that for data comprised of linear combinations of two spectra the two eigenvectors corresponding to the two largest eigenvalues should explain the majority of variance.

Lawton now uses (2.4.2) and (2.4.3) to write

$$\mathbf{y}_i = c_{i,1}\mathbf{s}_1 + c_{i,2}\mathbf{s}_2 + \boldsymbol{\varepsilon}_i = \kappa_{i,1}\mathbf{k}_1 + \kappa_{i,2}\mathbf{k}_2 + \boldsymbol{\varepsilon}_i$$

allowing us to define

$$\begin{aligned} \mathbf{s}_1 &= \xi_{1,1}\mathbf{k}_1 + \xi_{2,1}\mathbf{k}_2 \\ \mathbf{s}_2 &= \xi_{1,2}\mathbf{k}_1 + \xi_{2,2}\mathbf{k}_2 \end{aligned} \tag{2.4.4}$$

which we can write in matrix form as

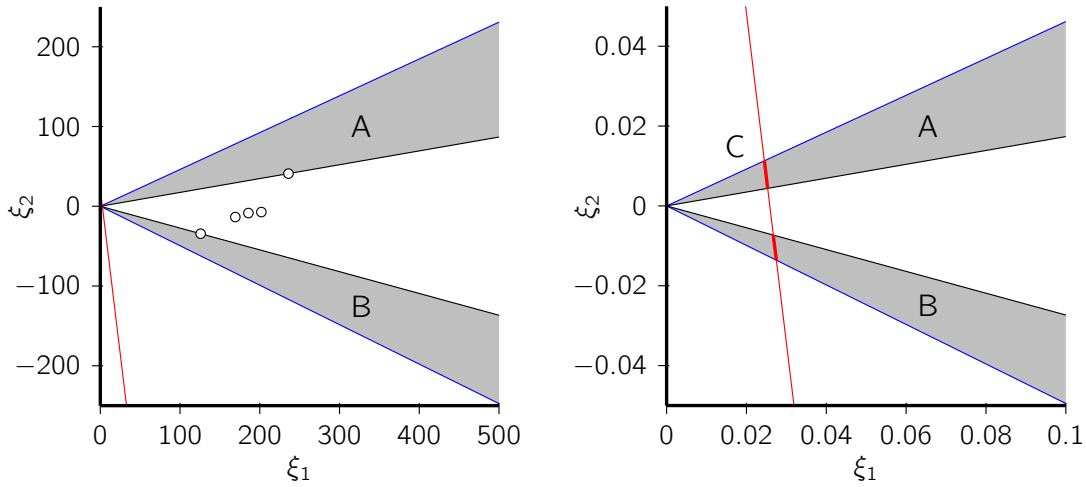
$$\mathbf{S} = \mathbf{K}\Xi$$

for

$$\mathbf{S} = [\mathbf{s}_1\ \mathbf{s}_2], \quad \mathbf{K} = [\mathbf{k}_1\ \mathbf{k}_2], \quad \Xi = \begin{bmatrix} \xi_{1,1} & \xi_{1,2} \\ \xi_{2,1} & \xi_{2,2} \end{bmatrix}.$$

Therefore our matrix of component spectra is equivalent to a transformation of the matrix containing the first two eigenvectors.

We now see, from (2.4.3) and (2.4.4), that both the $\mathbf{y}_i$ and $\mathbf{s}_k$ can be expressed as linear combinations of the eigenvectors, $\mathbf{k}_k$. We can therefore represent both the $\mathbf{y}_i$ and $\mathbf{s}_k$ on



(a) Plot of the $\mathbf{y}_i$, marked as five circles, and the region in which $\mathbf{s}_i$ can be defined in terms of the $\mathbf{k}_i$.

(b) The intersection between line C and regions A and B.

Figure 2.3: Reproduced from Lawton and Sylvestre [62]. The (ξ_1, ξ_2) plane of linear combinations of $\mathbf{k}_i$.

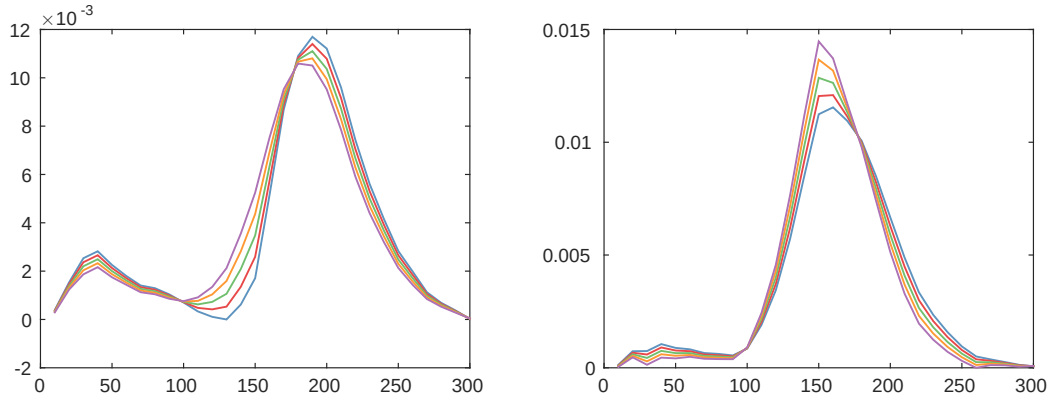
the same (ξ_1, ξ_2) plane. This of course places the $\xi_{l,k}$ in (2.4.4) as equivalent in purpose to the $\kappa_{i,1}$ and $\kappa_{i,2}$ in (2.4.3).

Using Lawton's original data we have an example of this (ξ_1, ξ_2) plane in figure 2.3 where the circles correspond to the linear combination coefficients in (2.4.3). As the linear combination coefficients to produce the $\mathbf{s}_k$ in (2.4.4) are unknown and require estimation Lawton defined boundaries for the region from which the coefficients may be selected. Lawton first restricted this region, from which we select the $\xi_{l,m}$, by noting that the $\mathbf{s}_i$ must be non-negative. Therefore the (ξ_1, ξ_2) plane is cut down to only those points satisfying

$$(\xi_1 \mathbf{k}_1 + \xi_2 \mathbf{k}_2)_j \geq 0 \quad \text{for } j = 1, ..., p \tag{2.4.5}$$

from the definition of the $\mathbf{s}_k$ in (2.4.4). For Lawton's data this region is marked on figure 2.3a as the area between the upper boundary of region A and the lower boundary of region B. Note that the five observations all lie within this area as they are all non-negative. The derivation of these boundaries and the following two are provided and proven to always exist in §4.3.

Lawton further restricted this plane by using the assumption that $c_{i,1}$ and $c_{i,2}$ in (2.4.2) must also be non-negative. This restriction adds the lower boundary to region A and the upper boundary to region B in figure 2.3a. The divide we see between the two feasible regions comes from the existence of two component spectra, therefore all spectra obtainable



(a) $\mathbf{s}_1$ **(b)** $\mathbf{s}_2$

Figure 2.4: Resulting $\mathbf{S}$ from Lawton's method, §2.4, over a range of possible points on the line segments identified in figure 2.3.

from one region will look similar. Due to this if $(\xi_{1,1}, \xi_{2,1})$ were to come from region A $(\xi_{1,2}, \xi_{2,2})$ must come from region B in order to avoid two copies of a single spectra when we calculate (2.4.4).

The final restriction to the plane is derived from the assumption that the $\mathbf{s}_k$ are normalised to have unit area. We show an example of this bound in figure 2.3 with line C. Therefore the line segments from which we can select the two pairs of coordinates, $(\xi_{1,1}, \xi_{2,1})$ and $(\xi_{1,2}, \xi_{2,2})$, are the intersection points between line C and regions A and B in figure 2.3b. We describe and expand upon these boundaries in §4.3 and §4.4.

The range of possible spectra, $\mathbf{s}_k$, cannot be restricted any further using the current assumptions. The width of the segments from which we select our component spectra depends on the data being analysed and how close the scanned object is to containing the pure spectra $\mathbf{s}_k$. Figure 2.4 shows the possible range of solutions we obtained for Lawton's dataset split by the two line segments. These constraints will be explored further, and the solution space proved finite, in §4.3 and §4.4.

2.5 Other Indirect Approaches for Spectral Decomposition

Whilst Lawton's method was the first there are more modern algorithms for indirectly solving (1.2.2) by first optimising (2.1.1) which further narrow down the possible solution space. Examples of these are evolving factor analysis (EFA) [71] and window factor analysis (WFA) [72], which use regions in the spectroscopic image where concentrations are zero, and heuristic evolving latent projection (HELP) [59], which uses a region in the image where only a single chemical component is present.

2.6 Non-Negative Matrix Factorisation

Most methods which directly solve (1.2.2) use some form of non-negative matrix factorisation to both estimate $\mathbf{C}$ and $\mathbf{S}$ and enforce (1.2.5) [53]. Non-negative matrix factorisation (NNMF) is a process which seeks to decompose a matrix with positive elements, $\mathbf{Y}$, such that

$$\mathbf{Y} \approx \mathbf{C}\mathbf{S}^T \tag{2.6.1}$$

where both $\mathbf{C}$ and $\mathbf{S}$ are also matrices with positive elements [90]. The majority of these methods seek to minimise [90]

$$f(\mathbf{C}, \mathbf{S}) = \|\mathbf{Y} - \mathbf{C}\mathbf{S}^T\|. \tag{2.6.2}$$

We can immediately see that, aside from a lack of unit length constraint on the columns of $\mathbf{C}$ or $\mathbf{S}$, this method is almost identical to our optimisation problem (1.2.2) and constraints (1.2.5). There are three main branches of methods to perform this factorisation, multiplicative update algorithms, gradient descent methods and alternating least squares based methods. The former two of these methods are not popular in chemometrics for solving (1.2.2). The multiplicative update algorithm was made popular by Lee and Seung [63, 64], improved upon by Gonzales and Zhang [39] and proven to converge by Lin [67] however is still notoriously slow [9]. Gradient descent methods on the other hand are more complicated to implement as they require extra parameters which can affect the convergence of the algorithm if not carefully chosen [9].

2.6.1 Alternating Least Squares

The most common form of non-negative matrix factorisation in chemometrics, and the most simple to implement, is alternating least squares. The algorithm works simply and quickly by alternately applying least squares optimisation to

$$\mathbf{Y} = \mathbf{C}\mathbf{S}^T + \mathbf{E}$$

to refine estimates $\hat{\mathbf{C}}$ and $\hat{\mathbf{S}}$, we detail this algorithm in §2.7.1. The non-negativity constraint was initially, and still often is, enforced by setting negative values to zero [64]. Whilst this algorithm is fast Bro and Jong [14] showed that the non-negative projection can lead to a non-least squares solution.

2.6.2 NNLS

The final form of NNMF we consider is the one to be predominantly used in this thesis and one of the most common modifications of the standard ALS algorithm to be used in chemometrics [89]. Non-negative least squares (NNLS) was suggested by Lawson and Hanson [61] and minimises

$$\|\mathbf{Ex} - \mathbf{f}\| \quad \text{subject to} \quad \mathbf{x} \geq 0 \tag{2.6.3}$$

which differs from the (2.6.1) in that it requires a vector $\mathbf{x}$ in place of one of the matrices.

It is an active set optimisation method [77], see appendix A.2, as it uses a pair of logical indicator vectors to denote the elements of $\mathbf{f}$ which violate or conform to the constraint. The algorithm solves the problem by first calculating a partial solution using only the elements which violate the constraint. These elements are then scaled, with the least negative element scaled first, until they are non-negative. Each time an element is moved between the active and inactive sets the partial solutions are recalculated. This process continues until a non-negative solution is found or an iteration limit is reached. A detailed intuitive explanation can be found in Bro and Jong [14] where standard NNLS was modified to become fast NNLS (FNNLS). This resulted in a reduction in time of between 80-95% for different implementations dependent on the exact model being optimised.

When used in place of standard least squares solutions to (2.6.1) this becomes an effective method for performing ALS based NNMF with convergence proved by Bertsekas [10]. ALS-NNLS also has a reduced possibility of returning a non-least squares solution when compared to standard ALS [14]. It does however decrease the speed of the algorithm. Increased computation time is due to the algorithm finding a vector solution to (2.6.3) where (2.6.1) uses the entire matrix. Therefore we must estimate each column of $\mathbf{W}$ and $\mathbf{H}$ individually.

2.7 Karjalainen's Methods of Spectral Decomposition

Karjalainen [55] proposed two methods of solving (1.2.2) to the field of chemometrics. The first was a constrained optimisation problem similar to a quadratic penalty method, see appendix A.4. This method however required numerical optimisation to fit the model and is therefore slow and expensive to calculate. The second proposed method was a far simpler least squares based approach using alternating least squares, the same non-negative matrix factorisation we describe in §2.6.1.

2.7.1 Alternating Least Squares for Spectral Reconstruction

Karjalainen's second proposed method is simply an implementation of the alternating least squares method of non-negative matrix factorisation, §2.6.1, and hence shares the same name, ALS [74, 85]. Although it is a relatively new method in chemometrics, due to the speed and simplicity of the algorithm it has has become popular for analysing Raman spectroscopy data [89]. We can define the full ALS algorithm of §2.6.1 in terms of (1.2.2) as

Algorithm 2.7.1. Alternating Least Squares

1. Estimate r, the number of component spectra, using chemical rank analysis, §2.4.
2. Fill the $n \times r$ matrix $\mathbf{C}_{(0)}$ with random numbers.
3. Calculate
$$\mathbf{S}_{(t)}^T = (\mathbf{C}_{(t-1)}^T \mathbf{C}_{(t-1)})^{-1} \mathbf{C}_{(t-1)}^T \mathbf{Y}$$
4. Constrain all $\mathbf{s}_k$ to unit length by assigning
$$\mathbf{s}_k \leftarrow \frac{\mathbf{s}_k}{\|\mathbf{s}_k\|} \quad \text{for } k = 1, ..., r$$
and non-negative elements by setting
$$s_{j,k} = 0 \quad \forall s_{j,k} \in \mathbf{S} \text{ with } s_{j,k} < 0.$$
5. Calculate
$$\mathbf{C}_{(t)}^T = (\mathbf{S}_{(t)}^T \mathbf{S}_{(t)})^{-1} \mathbf{S}_{(t)}^T \mathbf{Y}^T$$
6. Constrain all elements of $\mathbf{C}_{(t)}$ to positive values as in step 4.
7. Calculate the residuals, $\left\| \mathbf{Y} - \mathbf{C}_{(t)} \mathbf{S}_{(t)}^T \right\|$, and compare to the previous iteration.
8. If the improvement in fit is above a set tolerance return to step 3.

A more versatile version of ALS, known as multivariate curve resolution-alternating least squares (MCR-ALS), was proposed by Tauler and Casassas [96], and is the method used almost exclusively in this thesis. This method is simply a combination of ALS, §2.6.1, and NNLS, §2.6.2 allowing us to combine steps 3–4 and 5–6, although in practice NNLS is only required for one of the estimation steps with standard least squares used for the other. [55] suggested testing the stability of the solutions by adding small levels of noise

to the observations and recalculating the solutions. This will highlight any instability in the solutions as a small change in the data should only induce a small change in the solution.

Tauler and Casassas [96] applied the ALS algorithm to a variety of simulated datasets, and combinations of datasets, to investigate the performance of the algorithm with regards to spectral decomposition. The conclusions of their investigation were that ALS can easily be adapted to combined data matrices with two or more spectra in common, a property we will use in Chapter 5.

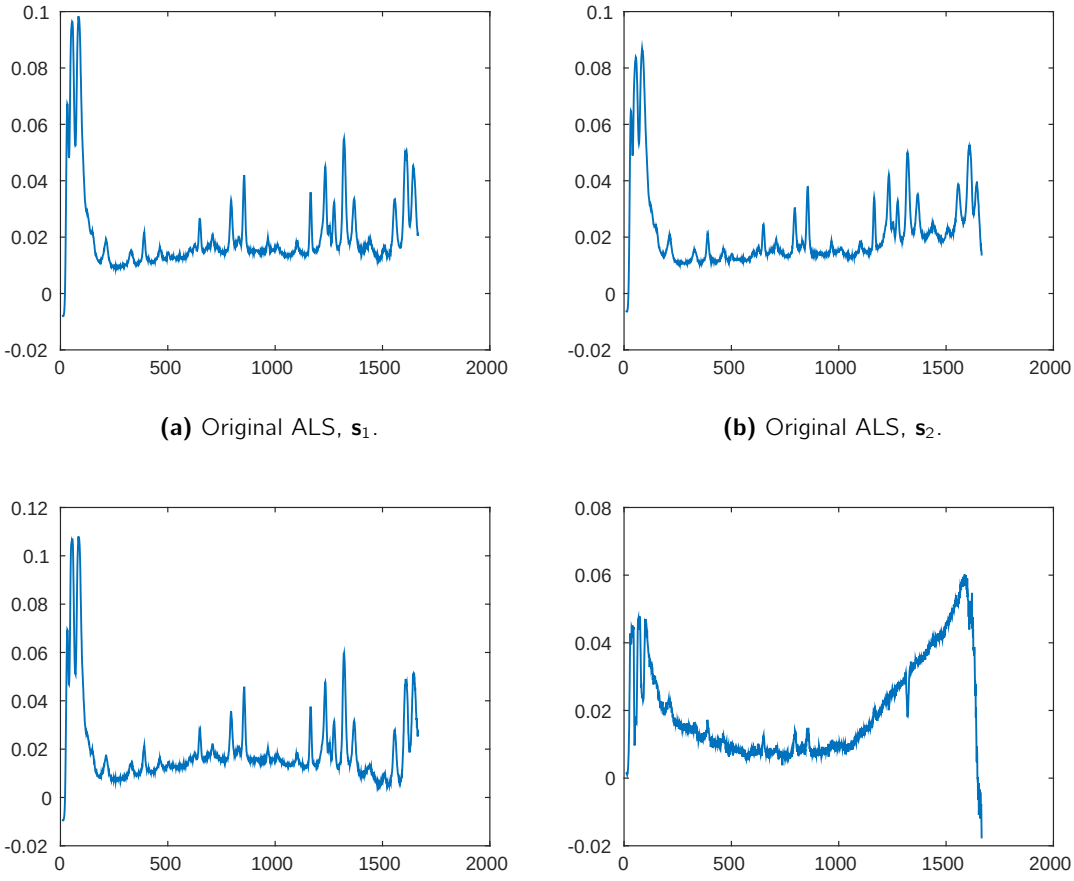
We also performed our own comparison of standard ALS against MCR-ALS using the paracetamol dataset, detailed in §1.2.2. In applying both algorithms we set $\mathbf{S}$ to have dimension $p \times 2$ with the aim of finding the known paracetamol component spectra and a second spectra containing the background signal from the edges of the Raman image. Figure 2.5 illustrates the improvements of MCR-ALS over standard ALS. The results from the original ALS method give two almost identical spectra whereas MCR-ALS gives the expected results of one clear spectra and one containing background signal. Further results of this algorithm are shown in figure 2.6 where we see heatmaps of the estimated concentrations, $\mathbf{c}_1$ and $\mathbf{c}_2$, relating to the spectra in figures 2.5c and 2.5d respectively. Here it is clear that whilst the algorithm was given an entirely random starting point the paracetamol component spectra and corresponding concentrations were detected very clearly.

2.8 Other Direct Approaches for Spectral Decomposition

Whilst other methods exist for directly solving (1.2.2) they are all based on the ALS algorithm of non-negative matrix factorisation, §2.6.1. They only differ in the starting values used for the algorithm where MCR-ALS uses randomly filled matrices for $\mathbf{C}$ and $\mathbf{S}$. Orthogonal projection analysis (OPA) [86, 87] uses observations from the dataset as initial estimates for the spectra, $\mathbf{s}_k$. The SIMPLISMA method (Simple-to-use interactive self-modelling mixture analysis) [102] uses the dataset, $\mathbf{Y}$, to calculate an initial estimate for the concentration matrix, $\mathbf{C}$.

2.9 Rational Resolution Method Comparison

To test these rational resolution techniques we applied MCR-ALS, OPA, SIMPLISMA and SIMPLEX1 [52] and SIMPLEX2 [52] to Lawton's data [62] and our paracetamol data, both detailed in §1.2.2. When applying OPA and SIMPLISMA we used the newer algorithms



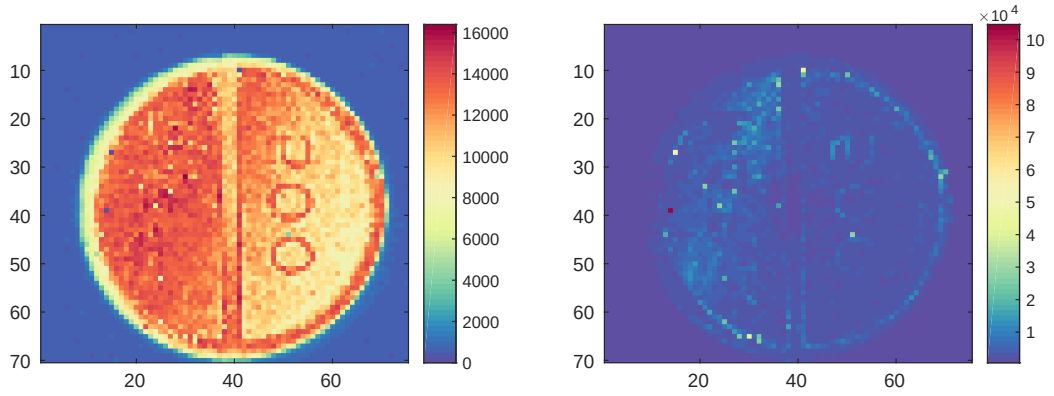
(a) Original ALS, $\mathbf{s}_1$.

(b) Original ALS, $\mathbf{s}_2$.

(c) MCR-ALS, $\mathbf{s}_1$.

(d) MCR-ALS, $\mathbf{s}_2$.

Figure 2.5: Results from the original ALS method [55] compared with those from MCR-ALS [96] when applied to the paracetamol dataset with $r = 2$. MCR-ALS returned the correct results of a single paracetamol spectra and background noise.



(a) MCR-ALS, $\mathbf{c}_1$. **(b)** MCR-ALS, $\mathbf{c}_2$.

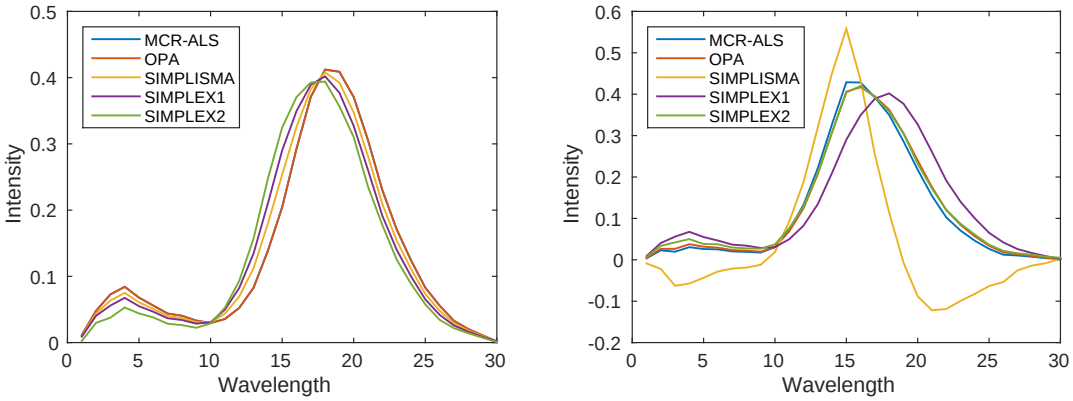
Figure 2.6: Concentration results from MCR-ALS corresponding to the spectra in figures 2.5c and 2.5d respectively. Given no prior knowledge of the spectra or concentrations the MCR-ALS algorithm returned very clear results.

	MCR-ALS	OPA	SIMPLISMA	SIMPLEX1	SIMPLEX2
Lawton	68.7658	68.7544	68.7383	69737	3070.3
Paracetamol	1.6160×10^{11}	1.5288×10^{11}	1.5280×10^{11}	2.3814×10^{11}	1.8908×10^{10}

Table 2.1: Residual Sums of Squares from (1.2.2) for each of the five tested SMCR methods when applied to both data from Lawton and Sylvestre [62] and the Paracetamol dataset.

involving NNLS in the least squares refinement stage so they were directly comparable to MCR-ALS, only differing in the initialisation of $\mathbf{C}$ and $\mathbf{S}$. SIMPLEX1 and SIMPLEX2 however were designed to avoid least squares optimisation of the solutions, instead projecting the data onto the proposed set of spectra, recalculating the concentrations and iteratively repeating until (1.2.2) is minimised. Whilst they minimise 1.2.2 there is no guarantee they achieve a least squares solution.

Table 2.1 shows us the expected result that the three ALS based methods all achieve almost the same minima whilst the SIMPLEX methods very in their solution. We therefore refer to figures 2.7 and 2.8 where it is clear that for Lawton's data all five algorithms detect $\mathbf{s}_1$ however SIMPLEX1 and SIMPLISMA appear to falter in detecting $\mathbf{s}_2$. When viewing the paracetamol data the only algorithm failing to clearly detect the component spectra is SIMPLEX2. From these simple tests we can see that no single algorithm is more effective, rather the effectiveness of each algorithm may depend on the dataset being analysed. Due to the simplicity we will use MCR-ALS for the majority of this thesis.



(a) $\mathbf{s}_1$ **(b)** $\mathbf{s}_2$

Figure 2.7: Comparison of the spectra detected by five SMCR methods when applied to Lawton's dataset [62].

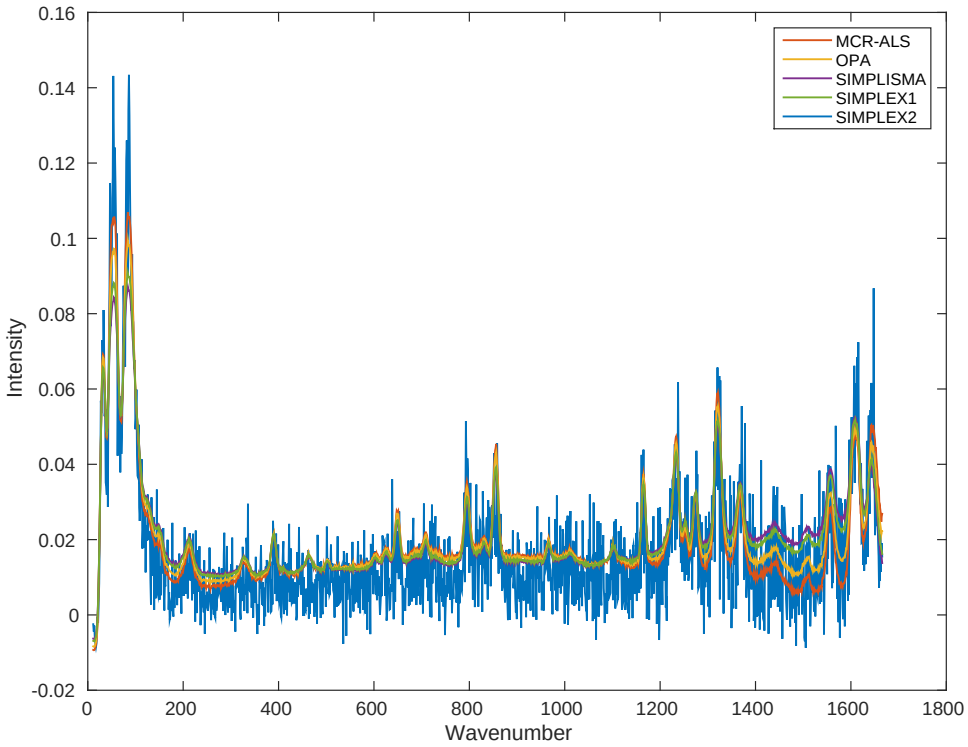


Figure 2.8: Comparison of the spectra detected by five SMCR methods when applied to the paracetamol dataset, §1.2.2. All algorithms except SIMPLEX2 returned a clear representation of the known component spectra of paracetamol.

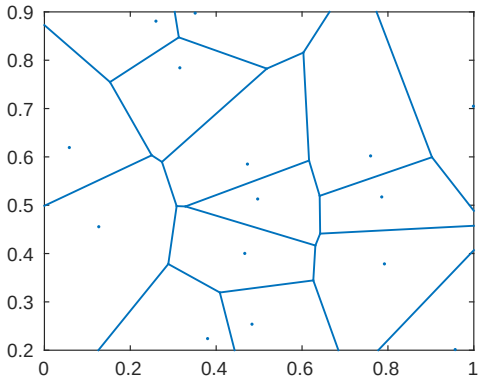


Figure 2.9: Randomly generated Voronoi diagram.

2.10 k-means Clustering

During our investigation into the methods with which we can analyse spectral decomposition results we regularly generate repeated samples of component spectra which must be grouped together, see Chapters 4 and 5. Also in our research into an improved method for solving (1.2.2) we are required to assign pixels in a Raman image into groups, Chapter 7. Therefore to achieve this we must implement forms of cluster analysis.

A very common method for partitioning data is k-means clustering. For a set of n observations k-means assigns observations to clusters based on the distance between the observation and the mean of each cluster [70]. This is analogous to partitioning the data into Voronoi cells as seen in figure 2.9 where the means are the cell centroids [7, 78]. We can write this as an optimisation problem for the objective function

$$\arg\min_{G} \sum_{m=1}^{k} \sum_{\mathbf{x}_i \in G_m} \|\mathbf{x}_i - \boldsymbol{\mu}_m\|^2 \tag{2.10.1}$$

for observations $(\mathbf{x}_1, ..., \mathbf{x}_n)$ and clusters of observations $G = \{G_1, ..., G_k\}$ such that

$$G_m = \{\mathbf{x}_i : \|\mathbf{x}_i - \boldsymbol{\mu}_m\|^2 \leq \|\mathbf{x}_i - \boldsymbol{\mu}_l\|^2 \, \forall l, 1 \leq l \leq k\}$$

with means $\{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_k\}$ calculated as

$$\boldsymbol{\mu}_m = \frac{1}{|G_m|} \sum_{\mathbf{x}_i \in G_m} \mathbf{x}_i.$$

In practice the Euclidean norm, $\|\cdot\|$, need not always be used in (2.10.1), for example we could instead use the L1 norm [92].

However this can prove computationally difficult. The most common algorithm for optimising (2.10.1) is Lloyd's Algorithm [69] which iteratively refines the clusters. This algorithm works as follows

Algorithm 2.10.1. *k*-means

1. Initialise means, $\left\{\boldsymbol{\mu}_1^{(1)}, ..., \boldsymbol{\mu}_k^{(1)}\right\}$, set $t = 1$.
2. **Assignment step**: Assign each observation to a cluster such that $\left\|\mathbf{x}_i - \boldsymbol{\mu}_m^{(t)}\right\|^2$ is minimised, to find the sets $G^{(t)} = \left\{G_1^{(t)}, ..., G_k^{(t)}\right\}$, defined as
$$G_m^{(t)} = \{\mathbf{x}_i : \left\|\mathbf{x}_i - \boldsymbol{\mu}_m^{(t)}\right\|^2 \leq \left\|\mathbf{x}_i - \boldsymbol{\mu}_l^{(t)}\right\|^2 \forall l, 1 \leq l \leq k\}$$
3. Halt if $G^{(t)} = G^{(t-1)}$ or $t = t_{max}$, for iteration limit t_{max}. Else continue to step 4.
4. **Update step**: Recalculate the means of the observations in each set, to be used as the centroids in the next iteration, with
$$\boldsymbol{\mu}_m^{(t+1)} = \frac{1}{|G_m^{(t)}|} \sum_{\mathbf{x}_i \in G_m^{(t)}} \mathbf{x}_i.$$

We can calculate initial estimates for the means, $\left\{\boldsymbol{\mu}_1^{(1)}, ..., \boldsymbol{\mu}_k^{(1)}\right\}$, using several methods [45, 80]. For example the Forgy method uses k randomly chosen observations as the initial estimates and the Random Partition method randomly assigns a cluster to each observation [45]. In testing different initialisations Hamerly and Elkan [45] noted that certain k-means based algorithms perform better when particular initialisation methods are used. For example the most common k-means algorithm, 2.10.1, is best used alongside the Forgy method.

2.11 Unweighted Pair Group Method with Arithmetic Mean

Another method for cluster analysis we consider is the unweighted pair group method with arithmetic mean (UPGMA). This is an agglomerative hierarchical clustering method proposed by Sokal and Michener [91]. The term agglomerative hierarchical implies this algorithm works from the bottom up building a hierarchy of clusters. Each observation begins as a separate cluster, with the algorithm pairing and combining clusters until only one cluster remains. This pairing is based on the minimum distance between clusters [51].

UPGMA constructs a dendrogram and pairwise similarity matrix which contains pairs of observations and the distance between them calculated using distance function $d(\mathbf{x}, \mathbf{y})$. Our choices of $d(\mathbf{x}, \mathbf{y})$ are any function returning a distance metric between two observations. To pair the closest two clusters we calculate the average distance between them using

$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{\mathrm{x} \in \mathcal{A}} \sum_{\mathrm{y} \in \mathcal{B}} d(\mathbf{x}, \mathbf{y}) \tag{2.11.1}$$

for sets of observations $\mathcal{A}$ and $\mathcal{B}$ and set cardinality $|\cdot|$.

2.12 Expectation-Maximisation Algorithm

The final clustering method we use is a model assuming the data is composed of two separate mixtures. Before defining the specific model we will first detail the algorithm with which it is fit.

The expectation-maximisation (EM) algorithm is an iterative method for finding maximum likelihood estimates (MLE) for parameters in statistical models containing unobserved variables. Dempster et al. [26] first generalised the algorithm to work with any MLE and convergence was proved by Wu [104]. We can use the algorithm on a statistical model consisting of observed data $\mathbf{X}$, unobserved data $\mathbf{Z}$, the parameters to be estimated, $\boldsymbol{\theta}$, and the likelihood function $L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})$. The MLE of the parameters $\boldsymbol{\theta}$ is determined using the marginal likelihood of $\mathbf{X}$.

$$L(\boldsymbol{\theta}; \mathbf{X}) = p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

As this is often difficult to analytically maximise or even define, the EM algorithm performs the maximisation iteratively using two steps [11, 26].

- Expectation step (E-step): Calculate the expected value of the log likelihood for the complete data, with respect to $\mathbf{Z}$ given $\mathbf{X}$, under the current parameter estimates $\boldsymbol{\theta}^{(t-1)}$.

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}\right) = \mathrm{E}\left[\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) \,|\mathbf{X}, \boldsymbol{\theta}^{(t-1)}\right]. \tag{2.12.1}$$

- Maximisation step (M-step): Find $\boldsymbol{\theta}$ such that Q is maximised

$$\boldsymbol{\theta}^{(t)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t-1)}\right) \tag{2.12.2}$$

2.12.1 Expectation-Maximisation for a Gaussian Mixture Model

A Gaussian mixture model is a statistical model comprised of a number of Gaussian densities [11]. When applied to data $\mathbf{X}$,

$$\overset{n\times d}{\mathbf{X}} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix},$$

we define the mixture model with m independent component densities as

$$p(\mathbf{x}_i|\Theta) = \sum_{k=1}^{m} \alpha_k p_k(\mathbf{x}_i|\boldsymbol{\theta}_k) \tag{2.12.3}$$

$$= \sum_{k=1}^{m} \alpha_k p_k(\mathbf{x}_i|\boldsymbol{\mu}_k, \Sigma_k) \tag{2.12.4}$$

where p_k is a d-dimensional multivariate normal PDF, $\sum_{k=1}^{m} \alpha_k = 1$ and

$$\Theta = (\alpha_1, ..., \alpha_m, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_m, \Sigma_1, ..., \Sigma_m).$$

Bilmes [11] shows the incomplete data log likelihood for a Gaussian mixture model to be

$$\log\left(L\left(\Theta|\mathbf{X}\right)\right) = \log\prod_{i=1}^{n} p\left(\mathbf{x}_i|\Theta\right) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{m} \alpha_k p_k\left(\mathbf{x}_i|\boldsymbol{\theta}_k\right)\right). \tag{2.12.5}$$

Due to (2.12.5) containing the natural logarithm of a summation this is a difficult equation to maximise. Therefore we maximise using the EM algorithm giving the name expectation-maximisation for a Gaussian mixture model (EMGM). The following equations were derived by Bilmes [11], performing both the E-step and M-step simultaneously, giving the algorithm.

Algorithm 2.12.1. EMGM

1. Initialise group memberships, α_k. Commonly performed using k-means, §2.10.
2. Estimate new α_k, for $k = 1, ..., m$, by summing the values of the normal PDF for each component density over all $\mathbf{x}_i$ and dividing by n

$$\alpha_k^{new} = \frac{1}{n}\sum_{i=1}^{n} p\left(k|\mathbf{x}_i, \Theta^{t-1}\right). \tag{2.12.6}$$

3. Estimate new μ_k, for $k = 1, ..., m$,

$$\boldsymbol{\mu}_k^{new} = \frac{\sum_{i=1}^{n} x_i p\left(k|\mathbf{x}_i, \Theta^{t-1}\right)}{\sum_{i=1}^{n} p\left(k|\mathbf{x}_i, \Theta^{t-1}\right)}. \tag{2.12.7}$$

4. Estimate new Σ_k, for $k = 1, ..., m$,

$$\Sigma_k^{new} = \frac{\sum_{i=1}^{n} p\left(k|\mathbf{x}_i, \Theta^{t-1}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_k^{new}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_k^{new}\right)^T}{\sum_{i=1}^{n} p\left(k|\mathbf{x}_i, \Theta^{t-1}\right)}. \tag{2.12.8}$$

5. Recalculate the log likelihood (2.12.5), increase iteration t by one and repeat steps 1–3 until the increase in the log likelihood falls below a set tolerance.

With our final estimate of Θ we now calculate the group membership, z_i, of each $\mathbf{x}_i$ with

$$z_i = \max_k\left(\alpha_k p_k\left(\mathbf{x}_i|\boldsymbol{\theta}_k\right)\right)$$

which identifies the Gaussian distribution from which $\mathbf{x}_i$ is most likely to have come.

2.12.2 Relationship between EMGM and k-means clustering

A key difference between EMGM and k-means is in their methods of assigning clusters. Where k-means assigns each $\tilde{y}_i$ to a cluster based on the distance from the cluster mean μ_m, calculated as

$$\|\tilde{y}_i - \mu_m\|,$$

EMGM assigns the $\tilde{y}_i$ by calculating how likely they are to have come from Gaussian distribution $\mathcal{N}\left(\mu_m, \sigma_m^2\right)$. Therefore EMGM will take even small features of a $\tilde{y}_i$ into consideration where k-means may ignore them in favour of a lower difference. We can see this in a simple example with two groups where we assume that k-means and EMGM find groups with the same means, μ_1 and μ_2, however EMGM also estimates the variances of these groups as σ_1^2 and σ_2^2. If the observations in one group were to represent noise $\sigma_1^2 < \sigma_2^2$ as very few, if any, features will be present. Now for an observation displaying a large amount of noise with weak features, and therefore a low $\tilde{y}_i$ in (7.2.1) in comparison to observations mode clearly from group 2, k-means may assign the observation to the first cluster because

$$\|\tilde{y}_i - \mu_1\| < \|\tilde{y}_i - \mu_2\|.$$

However EMGM calculates the likelihoods

$$l\left(\tilde{y}_i|\mu_1, \sigma_1^2\right)$$

and

$$l\left(\tilde{y}_i|\mu_2, \sigma_2^2\right)$$

and calculates group assignments as in step 2 of algorithm 2.12.1. Due the first group observations displaying little in the way of structure σ_1^2 may be small enough to render $\tilde{y}_i$ extremely unlikely to have come from $\mathcal{N}(\mu_1, \sigma_1^2)$. EMGM therefore would assign observation $\mathbf{y}_i$ to the second cluster as in spite of a greater difference $\|\tilde{y}_i - \mu_2\|$ the likelihood of $\mathbf{y}_i$ coming from $\mathcal{N}(\mu_2, \sigma_2^2)$ is larger.

Chapter 3

Pre-processing Raman Spectroscopy Data

3.1 Objectives of Pre-Processing

Raman spectroscopic images can present a number of anomalies. In this chapter we will investigate methods of removing or adjusting these outlying elements in a data matrix. We will classify the types of anomalies into local and global anomalies. Local anomalies effect a single pixel, $\mathbf{y}_i$, or wavenumber of data matrix $\mathbf{Y}$, whilst global anomalies feature throughout the dataset. Our aims in this chapter are to answer the questions, what can we do to adjust these anomalies and what effects will this have on the results obtained by solving our standard bilinear model (1.2.3). We must also be careful not to over adjust the data by mistaking features of the Raman image as anomalies.

Whilst it is possible to reduce the occurrence of anomalies at the scanning stage this can extend the data acquisition time, by using the average of multiple scans at each pixel, or require specialised equipment which in turn increases the cost. For example scanning a subject through different colours of glass can introduce peaks in the spectra therefore the effects of coloured glass must be mitigated by specific wavelength lasers [15]. Another equipment improvement is the use of more sensitive laser receivers less susceptible to external interference in the light source [76].

Alongside adjusting anomalies we also look at a technique to adjust the wavenumber scale of similar datasets so we can directly compare them without the risk of misaligned spectral peaks corrupting our results. This technique will be used in Chapter 5. We also look at a potential improvement to the method of chemical rank analysis, §2.2, using a robust principal component analysis algorithm, §2.3.3.

3.2 Common Types of Anomaly

There are several common types of anomaly which can appear in a Raman spectroscopic image which we interpret as outliers in the data matrix. The anomalies which we see on a local level are,

- Cosmic ray - These anomalies occur when external interference alters the light wave received by the spectrometer. Sources of external interference can vary however one of the more common is gamma radiation from space, hence the name cosmic ray anomaly [13]. These anomalies appear in the pixels scanned whilst the interference was present and only at specific wavenumbers. We therefore see unusually high values in the data matrix at

$$y_{i,j} \quad \text{for } i_1 \leq i \leq i_2 \text{ and } j_1 \leq j \leq j_2 \tag{3.2.1}$$

  where $[i_1, i_2]$ and $[j_1, j_2]$ are the pixel and wavenumber ranges effected by the interference. We see a cosmic ray anomaly in figure 3.2a where the peaks shown in figure 3.1 are now far exceeded by the peak introduced from the external interference.

- Burnt pixel - If the laser remains on a pixel for too long the surface of the subject may burn [13]. This leads to unusually high light intensities for that pixel which we see as a $\mathbf{y}_i$ with a mean far larger than the dataset mean,

$$\frac{1}{p}\sum_{j=1}^{p} y_{i,j} >> \frac{1}{np}\sum_{i,j} y_{i,j}. \tag{3.2.2}$$

  An example of a burnt pixel is shown in figure 3.2b where we see the intensities are far greater than those of a correct observation as seen in figure 3.1.

The global anomalies we often encounter are,

- Excessive noise - There are many causes for high levels of noise in a Raman image, one such cause is known as weak instrument response, a situation where the spectrometer receives a weak reflection of the light source from the image [13]. High noise is difficult to immediately detect in the dataset unlike the local anomalies however it is noticeable after fitting $\mathbf{Y} = \mathbf{C}\mathbf{S}^T + \mathbf{E}$ as the estimated spectra or concentrations will appear noisy.

- Incorrect baseline - An incorrect baseline in a Raman spectra manifests as a linear trend in non-peak sections of the spectrum. For example a correct spectrum will have intensities at, or just above, zero with only the peaks varying from this. If the

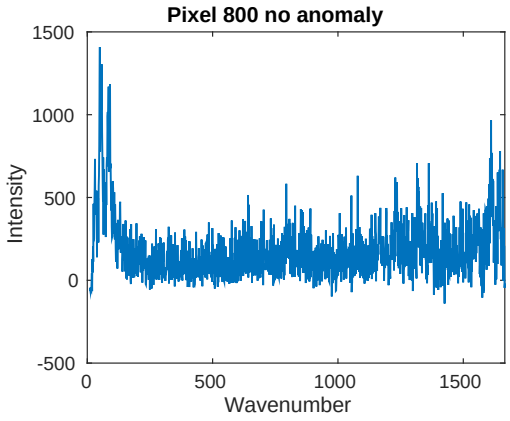
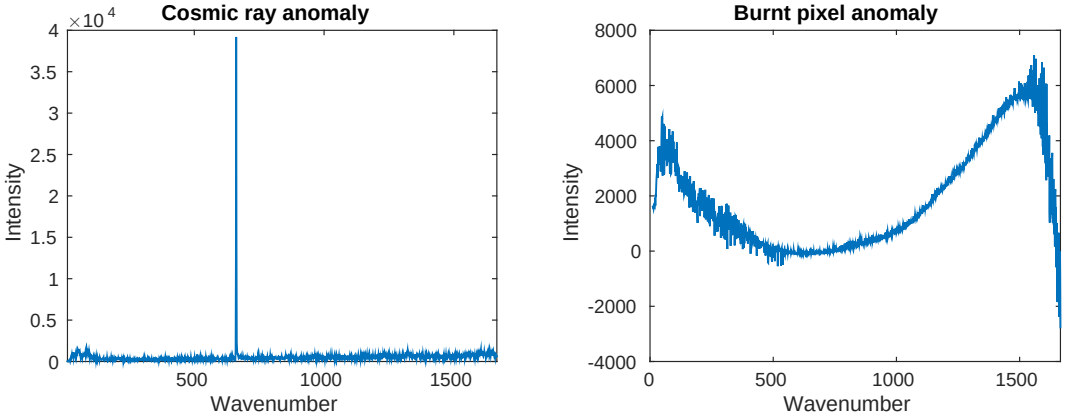


Figure 3.1: The received spectra of a single pixel of Paracetamol with no anomalies.



(a) Cosmic ray. **(b)** Burnt pixel.

Figure 3.2: Comparison of anomalies in the Paracetamol data.

baseline is incorrect the zero intensity values may lie on a gradient. The cause of this effect can be hardware and the inbuilt processing sources [20] however in the case of our paracetamol data, §1.2.2, the age of the tablet may have caused the chemical components to degrade [50].

3.3 Anomaly Trimming

We first consider the local anomalies, the two main types of which are described in §3.2. Current methods for removing these anomalies only cover the cosmic ray anomaly as a burnt pixel is irreversible in that the scan subject has been physically altered. Cosmic rays can be avoided at the scanning stage by using a specialised laser receiver [76] or by simply taking the average of numerous scans of a single pixel [83, 98]. This latter method works as cosmic ray interference passes fast enough for successive scans to be unaffected. Current methods for removing cosmic ray anomalies in datasets involve identification by detecting

the edges of affected clusters of pixels [98] or by locating pixels which differ too greatly from the standard deviation of pixels in the image [83].

3.3.1 Cosmic Ray Anomaly Trimming

Cosmic ray anomalies, as described by (3.2.1), are high values for a select number of pixels and wavenumbers. In correcting cosmic ray anomalies we first consider the very simple method of removing the affected wavenumbers. For example if a cosmic ray anomaly is present in wavenumber j we may see that

$$y_{i,j} >> y_{i,j'} \quad \text{for } 1 \leq j, j' \leq p,\ j \neq j'$$

and so we remove

$$y_{i,j} \quad \text{for } i = 1, ..., n.$$

We may also identify cosmic ray anomalies when viewing the estimated component spectra in a dataset if an extra peak is present. Whilst we often have over 1000 wavenumbers for each dataset and this process will only result in a small loss of data we may inadvertently remove important information such as peaks in the spectra which define a component chemical.

We therefore propose a process similar to Winsorising [28, 97]. Winsorising at 90% is defined for ordered data $\mathbf{x}$, with $x_1 \leq x_2 \leq ... \leq x_n$ as

$$\{x_i : 1 \leq i \leq (|0.05n| - 1)\} = x_{|0.05n|}$$
$$\{x_i : (|0.95n| + 1) \leq i \leq n\} = x_{|0.95n|}.$$

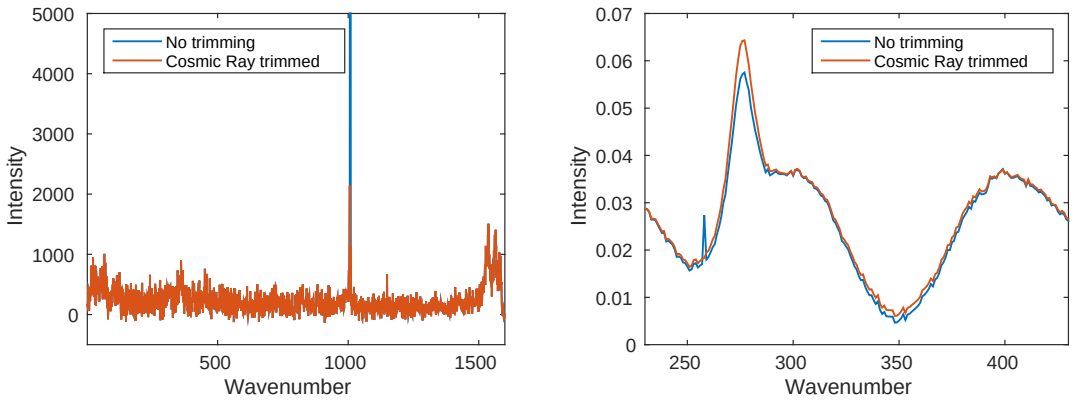
All values lying outside the upper and lower 5% of the ordered data are set equal to the 5^{th} and 95^{th} percentile values respectively [28].

We modify this mean to correct cosmic ray anomalies with the following algorithm,

Algorithm 3.3.1. Cosmic ray anomaly trimming

1. Select the $|\alpha np|$ greatest elements in the data matrix to trim. In testing we found $\alpha = 0.001$ is usually enough to remove anomalies whilst leaving the spectral peaks unaffected.

2. Identify the set, $\mathcal{C}$, of matrix indices, (i, j), corresponding to the largest $|\alpha np|$ elements in the dataset.



(a) Paracetamol Raman spectra

(b) Dentine Raman spectra between 230-440cm^{-1}

Figure 3.3: The effects of cosmic ray anomaly trimming with $\alpha = 0.001$ and $\beta = 0.05$ on an observation from the paracetamol data and spectra identified in the Human Tooth dataset. Both show a reduction in the amplitude of a cosmic ray anomaly.

3. For $m = 1, ..., |\mathcal{C}|$ replace each identified potential anomaly using

$$y_{m,j} = \frac{1}{|\mathcal{L}_j|} \sum_{l \in \mathcal{L}} y_{m,l} \tag{3.3.1}$$

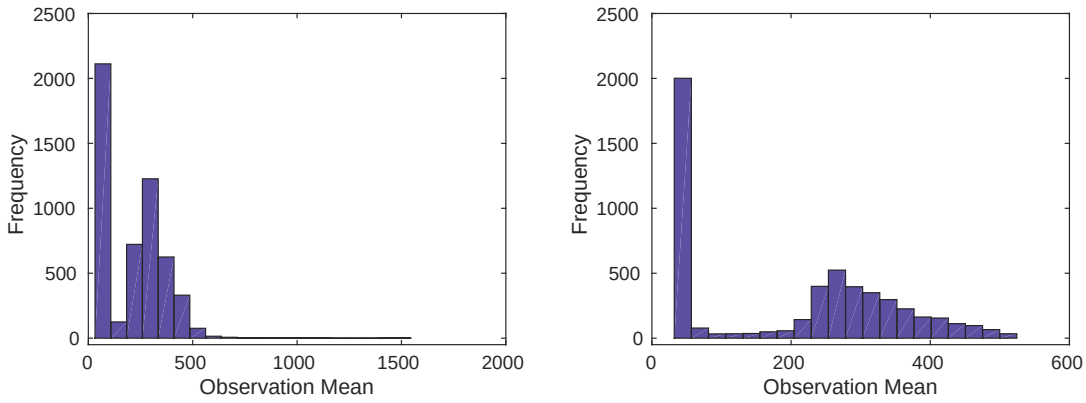
where $\mathcal{L}_j = \{l : j - |\beta p| \leq l \leq j + |\beta p|, l \neq j, y_{i,l} \notin \mathcal{C}\}$. The set $\mathcal{L}$ causes each $y_{i,j}$ to be replaced by the average of βp values either side, excluding those also identified as potential anomalies. In practice we found $\beta = 0.05$ adequately smooths out cosmic ray anomalies.

We see the results of this trimming in figure 3.3. In the raw paracetamol data a cosmic ray anomaly was reduced from over 38000. We can also see the effects on the results from solving (1.2.3) when applied to the human tooth data, §1.2.2. Here an anomalous peak in the dentine spectra was reduced resulting in a clearer overall spectra with a more pronounced peak where expected when compared to the true spectra [38, 105].

3.3.2 Burnt Pixel Anomaly Trimming

The second local anomaly we consider are burnt pixel anomalies. We also remove these using a method of smoothing similar to the Windsoring as used in §3.3.1. Our process for removing these anomalies is,

Algorithm 3.3.2. Burnt pixel trimming



(a) Untrimmed data. **(b)** Trimmed data.

Figure 3.4: Burnt pixel trimming results for the paracetamol data shown as histograms of the observation means (3.3.2). The trimming has successfully scaled all the anomalously high values.

1. Calculate the $n \times 1$ vector of observation means, $\tilde{\mathbf{y}}$, with

$$\tilde{y}_i = \frac{1}{p} \sum_{j=1}^{p} y_{i,j} \tag{3.3.2}$$

2. Identify the set, $\mathcal{B}$, of indices, i, of the αn largest means in vector $\tilde{\mathbf{y}}$. In testing we found $\alpha = 0.0001$ is enough to remove any burnt pixels.

3. For $i = 1, ..., |\mathcal{B}|$ replace each burnt pixel using

$$\mathbf{y}_i = \frac{1}{|\mathcal{M}_i|} \sum_{m \in \mathcal{M}_i} \mathbf{y}_m \tag{3.3.3}$$

   where $\mathcal{M}_i$ is the set of indices corresponding to pixels surrounding $\mathbf{y}_i$ in the Raman image. We exclude some pixels from set $\mathcal{M}$ if the pixel of interest, i, lies on the edge of the Raman image.

We can see results of this averaging in figure 3.4 where the pixels with a mean of over 500 have been scaled

3.4 Variance Scaling

The first global anomaly we investigate is an excess of noise in the dataset. Under the basic model

$$\mathbf{y}_i = \mathbf{S}\mathbf{c}_i + \mathbf{e}_i$$

with

$$\mathbf{e}_i \sim \mathcal{N}_p(\mathbf{0}, \sigma^2 \mathbf{I}_p) \quad \text{for } i = 1, ..., n$$

a high level of noise, $\mathbf{e}_i$, implies σ^2 is large. We therefore scale each observation in our dataset with some estimate of the variance, v_i,

$$\frac{\mathbf{y}_i}{\sqrt{v_i}} \tag{3.4.1}$$

in order to approach

$$\mathbf{e}_i \sim \mathcal{N}_p(\mathbf{0}, \mathbf{I}_p) \quad \text{for } i = 1, ..., n$$

for large n. This will emphasise the detected spectra in each observation [32] as the constraint that $\mathbf{s}_k^T \mathbf{s} = 1$ means

$$\frac{\mathbf{S}\mathbf{c}_i}{\sqrt{v_i}}$$

will only effect the concentrations, $\mathbf{c}_i$, as $\mathbf{S}$ cannot be scaled. Therefore if clear spectra are the intended result the data is scaled as the noise is reduced with no effect on the spectra. However if clear concentrations are the objective then we do not perform variance scaling as we also scale the estimated concentrations, losing detail.

3.4.1 Methods of Variance Scaling

In order to scale our data using (3.4.1) we must first estimate the variance, σ^2. The most common estimate, and the most sensible considering our assumption that the $\mathbf{e}_i$ are normally distributed, is the standard deviation

$$v_i = s_i^2 = \frac{1}{p-1} \sum_{j=1}^{p} \left(y_{i,j} - \bar{y}_i\right)^2 \tag{3.4.2}$$

where $\bar{y}_i = \frac{1}{p} \sum_{i=1}^{p} y_{i,j}$.

The second variance estimate we consider is Poisson scaling, also known as square root mean scaling, which was shown to be effective when used with mass spectra by Keenan and Kotula [57]. Use of this estimate assumes the noise in our observations is proportional to the mean of the observation. Therefore we calculate, for each observation $\mathbf{y}_i$,

$$v_i = \tilde{y}_i = \frac{1}{p} \sum_{j=1}^{p} y_{i,j}. \tag{3.4.3}$$

3.4.2 Variance Scaling Comparison

In comparing these two scaling methods we look at both the effects on **S** and **C** after solving (1.2.3). We also investigate the effects on the results of chemical rank analysis, described in §2.2. The effects on **S** and **C** were as we expected from the results of Ferraro et al. [32]. Without any scaling applied to the data we see in figure 3.5a that the initial results of (1.2.2) returned two very similar spectra when only one chemical component should be detected. The heatmap 3.5b for $\mathbf{c}_1$ however, when reshaped into the original Raman image dimensions as detailed in §1.2, gives a clear indication of the location of the chemical component in the tablet.

After standard deviation scaling, using (3.4.2) in (3.4.1), we now see in figure 3.5c that we have one clear paracetamol spectrum, with lower noise when compared to the unscaled estimates, and a second spectrum now representing only background signal. As expected though the heatmap of concentrations, 3.5d, now contains little detail aside from an outline of the tablet.

Poisson scaling, using (3.4.3) in (3.4.1), proved less successful with Raman spectra than it did for Keenan and Kotula [57] with mass spectra. This difference is presumably because mass spectroscopy uses electrons in the subject where Raman spectroscopy uses photons and the models used to decompose these datasets differ. Although the concentration heatmap, figure 3.5f, remains unaffected the spectra, shown in figure 3.5e, are less clear. When compared to the spectra in figure 3.5c we see our two estimates after Poisson scaling are a noisy combination of both the paracetamol spectrum and the background noise.

For our final comparison of scaling techniques we look at the eigenvalues of covariance matrix, $\frac{\mathsf{Y}^T\mathsf{Y}}{n}$, before and after correction to determine chemical rank. With eigenvalues, λ_j, such that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$ we calculate the vector, **l**, of cumulative sums

$$l_m = \sum_{j=1}^{m} \frac{\lambda_j}{\sum_{J=1}^{p} \lambda_J} \tag{3.4.4}$$

and, as described in §2.2, use the point at which this vector plateaus as our number of component spectra. Figure 3.6 shows l_m, (3.4.4) for $m = 1, ..., 5$, calculated using unscaled and scaled data. We see little difference between the three scaling techniques in terms of or chemical rank analysis results. Unscaled and Poisson scaled data appear to cause the second eigenvector to explain a little more variance however this is not enough for us to assume a second component spectra exists.

In this thesis we will use standard deviation scaling if only the estimated spectra are of interest to us, for example in Chapter 7 we judge our proposed new decomposition method based on the spectra it detects and therefore scale the data. However in Chapters 5 and 6

we focus on analysing the estimated concentrations and so do not scale the data.

3.5 Baseline Correction

The second global anomaly we investigate is an incorrect baseline in the pixels, $\mathbf{y}_i$, of our Raman image, as described in §3.2. With chemical identification via spectra performed by examining the peaks in the spectra, any structure present in the non-peak parts of a spectra may result in identification errors. We see an example of this in figure 3.7a where an observation from our paracetamol dataset is plotted against the wavenumbers. There have been numerous publications on the subject of baseline correction with a large number focused on chromatography and nuclear magnetic resonance spectroscopy [19, 20, 43]. More recently Zhang et al. [106] proposed a method using iteratively refined least squares where the data are weighted to discount the effect of peaks on the model fit. However we propose a more simple method which can be fit using any numerical optimisation routine.

We begin with the definition of an x^{th} order linear model fit to a pixel from a Raman image,

$$\mathbf{y}_i = \mathbf{Z}\boldsymbol{\theta}_i + \boldsymbol{\varepsilon}_i \tag{3.5.1}$$

for unknown parameters $\boldsymbol{\theta}_i$ and

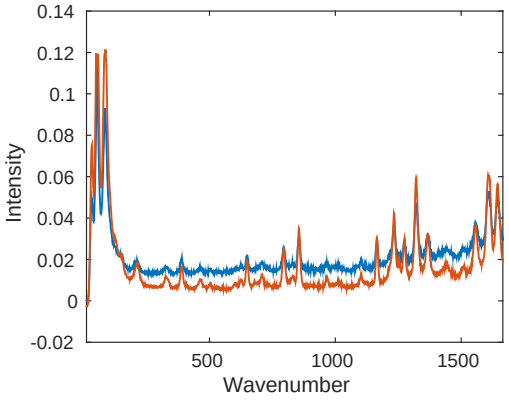
$$\mathbf{Z} = \begin{bmatrix} 1 & w_1 & \dots & w_1^x \\ \vdots & \vdots & \dots & \vdots \\ 1 & w_p & \dots & w_p^x \end{bmatrix}$$

where $\mathbf{w}$ is a vector of wavenumbers. This model will help us identify an incorrect baseline as any linear structure in the observations will cause the line fit by this model to have a non-zero gradient. We see this in figure 3.7a where the 3^{rd} order linear model fit has a clear positive gradient after the spectral peak.

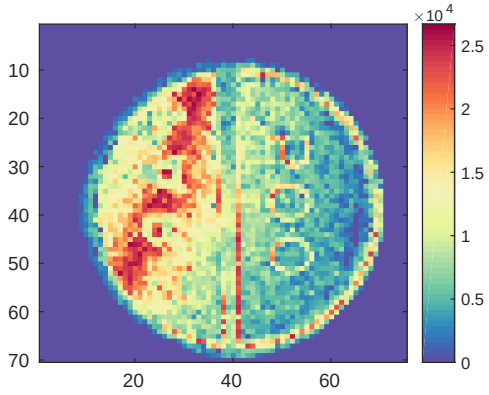
We may be able to remove this gradient by replacing each spectra with the residuals of the linear model,

$$\tilde{\mathbf{y}}_i = \boldsymbol{\varepsilon}_i = \mathbf{y}_i - \mathbf{Z}\boldsymbol{\theta}_i, \quad \text{for } \; i = 1, ..., n. \tag{3.5.2}$$

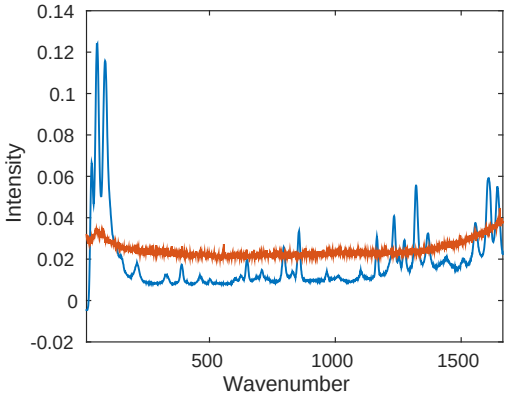
however in using these residuals we will cause a large proportion of the data to become negative. As discussed in §1.2 this would render our data physically implausible as the data represent light intensities and so cannot be negative. We must therefore take care to avoid introducing negativity into our data. To avoid this we propose a modification to our linear model to include a penalty term which promotes positive residuals whilst still minimising



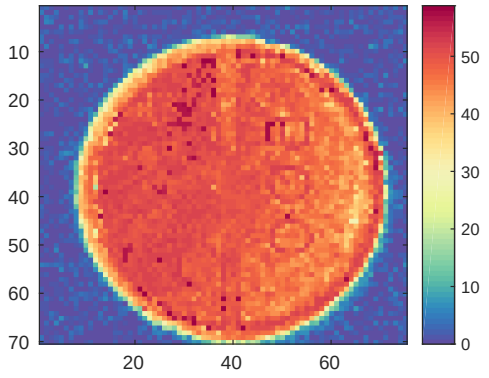
(a) Unscaled ALS derived spectra, $\mathbf{s}_1$ and $\mathbf{s}_2$.



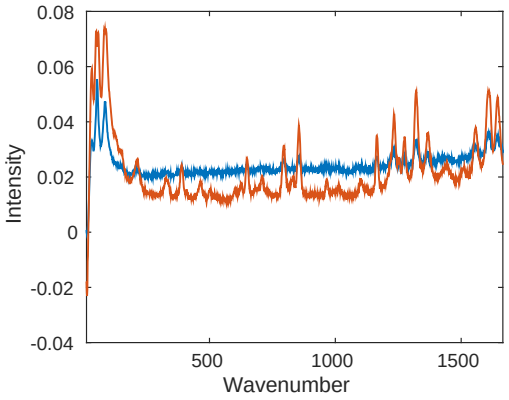
(b) Unscaled ALS derived concentrations for paracetamol, $\mathbf{c}_1$.



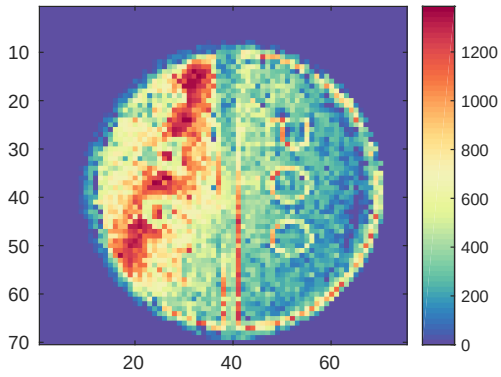
(c) Scaled ALS derived spectra, $\mathbf{s}_1$ and $\mathbf{s}_2$.



(d) Scaled ALS derived concentrations for paracetamol, $\mathbf{c}_1$.



(e) Poisson scaled ALS derived spectra, $\mathbf{s}_1$ and $\mathbf{s}_2$.



(f) Poisson scaled ALS derived concentrations for paracetamol, $\mathbf{c}_1$.

Figure 3.5: Comparison of scaling techniques on the bilinear model results for paracetamol data. Spectra, $\mathbf{s}_k$, are plotted against wavenumbers and concentrations, $\mathbf{c}_1$, are reshaped into the original Raman image dimensions.

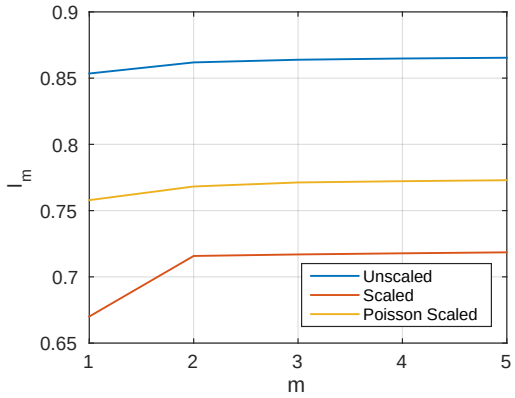


Figure 3.6: Comparison of (3.4.4) calculated using the eigenvectors of unscaled and scaled paracetamol data. By chemical rank analysis, §2.2, standard deviation scaled data indicates two component chemicals however with unscaled and Poisson scaled data we see the correct result of a single component chemical in the data.

the errors. Our model remains unchanged,

$$\mathbf{y}_i = \mathbf{Z}\boldsymbol{\theta}_i + \boldsymbol{\varepsilon}_i \tag{3.5.3}$$

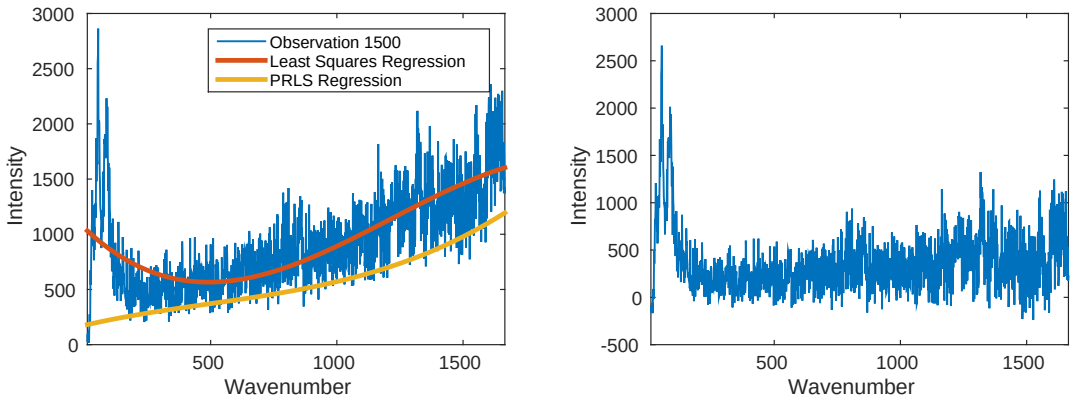
however is now subject to the constraint that

$$\boldsymbol{\varepsilon}_i = \mathbf{y}_i - \mathbf{Z}\boldsymbol{\theta}_i > 0$$

for Euclidean norm $\|\cdot\|$. We now define our objective function such that the standard least squares objective function is increased by the magnitude and quantity of negative residuals,

$$\sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{Z}\boldsymbol{\theta}_i\|^2 + \lambda \sum_{j:\varepsilon_{i,j}<0} |\varepsilon_{i,j}| \tag{3.5.4}$$

where λ controls how severely violations of the constraint are penalised, a technique known as a quadratic penalty method, detailed in §A.4. We then fit this model using a numerical optimisation method such as the simplex method [60], used here due to computation speed. We see the effects of this penalty term on a third order linear model of the form (3.5.1) in figure 3.7a. The underlying structure in the data is still detected by (3.5.3) however we now see that the residuals will be largely positive. Another effect of penalising negative residuals is a lessening of the effects of spectral peaks on the fitted line due to their contribution to negative residuals as seen in figure 3.7a. Replacing each received spectra in the data with the residuals of this model, as in (3.5.2), gives us spectra as seen in figure 3.7b. The incorrect baseline in the data is now removed as there is no obvious linear trend to the spectrum and the identifying peak of paracetamol remains.



(a) 3rd order linear model and penalised model.

(b) Penalised model residuals.

Figure 3.7: Comparison of linear models (3.5.1) and (3.5.4) when fit to observation 1500, $\mathbf{y}_{1500}$, from the paracetamol data. (3.5.4) is fit with $\lambda = 100$ using the simplex method. Results show an almost complete removal of the linear trend in the original spectrum.

3.6 Wavenumber Adjustment

The final method of pre-processing we investigate for Raman spectroscopy data is of use when datasets are to be compared and will be pooled before analysis, as we do in Chapter 5. Before each use a Raman spectrometer must be calibrated so each element in the detected spectrum of light intensities, **d** is associated with the correct value in a vector of wavenumbers, **w**. The spectrometer will detect the Raman spectrum at a pixel with no knowledge of the wavenumbers with which it corresponds, instead pairing **d** with a default vector of wavenumbers, **w**'. Therefore the user must calibrate the machine, altering **w**', so any peaks in the spectrum lie at the correct wavenumbers [50]. For example if we know spectrum **d** has a maximum peak at some wavenumber W we must adjust **w** so

$$w_j = W \quad \text{for } j = \max_j d_j$$

If the calibration is performed correctly we will have a vector of wavenumbers, **w**, which when paired with spectrum **d** assigns the correct wavenumbers to the features of the spectrum. However this alignment may not be identical for multiple scans due to the sensitivity of Raman spectrometers. Therefore for datasets with supposedly identical component spectra the wavenumbers with which they are associated may not be the same.

With most forms of our analysis, for example estimating the pure spectra by solving (1.2.3), this has no effect on our results. Providing the wavenumbers are close enough for us to identify peak locations this also has no effect on the interpretations a chemist may make.

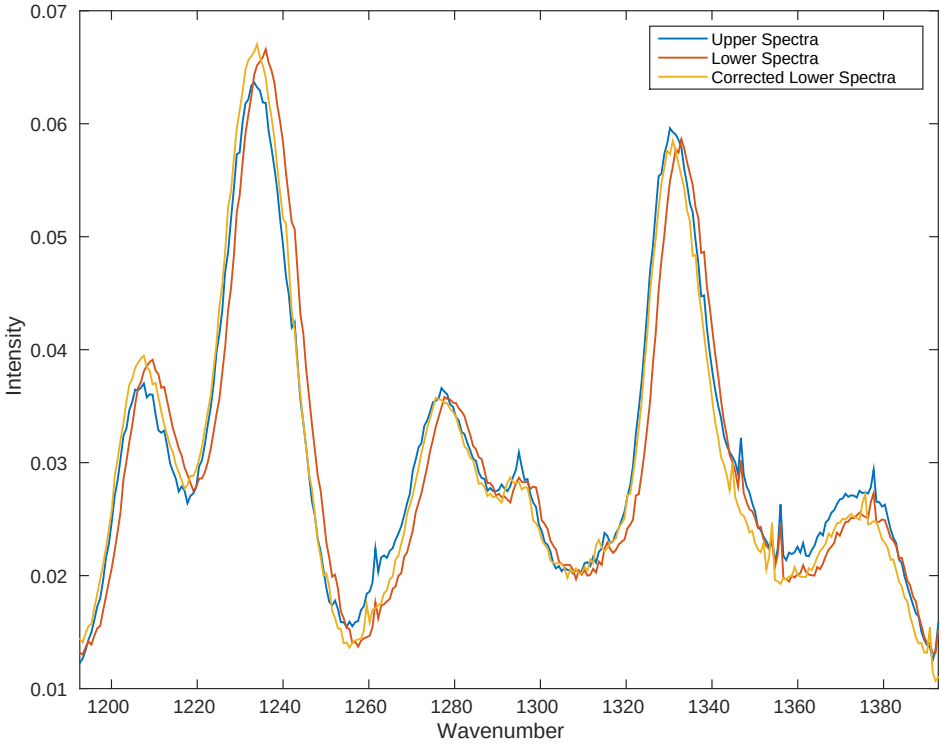


Figure 3.8: Bicalutamide spectra obtained from applying (1.2.2) to both sides of a bicalutamide tablet scan. Algorithm 3.6.1 is then applied to the lower tablet scan and (1.2.2) refit.

However when concatenating multiple datasets it is important the wavenumbers of each dataset match up to ensure the spectra we obtain are as clear as possible. In the worst case misaligned datasets may result in a single component spectra being identified as two near identical spectra due to a difference in peak location. We see an example of incorrectly aligned spectra in figure 3.8 where the blue spectrum appears to lie to the left of the red spectrum.

We propose a simple algorithm to align the wavenumber scales of two datasets. We can describe the process intuitively as fixing the wavenumber scale of the first dataset then moving the observations of the second dataset along this scale until the two datasets match up. A process seen in figure 3.9 where the black spectrum is fixed and the coloured spectra show the second dataset at various stages of being moved along the wavenumber scale. Our proposed algorithm for aligning the observations of datasets $\mathbf{Y}_1$ and $\mathbf{Y}_2$ is as follows,

Algorithm 3.6.1. Wavenumber adjustment

1. Select an observation from each dataset, $\mathbf{y}_1$ and $\mathbf{y}_2$, to use in estimating the shift amount (see *note* below).

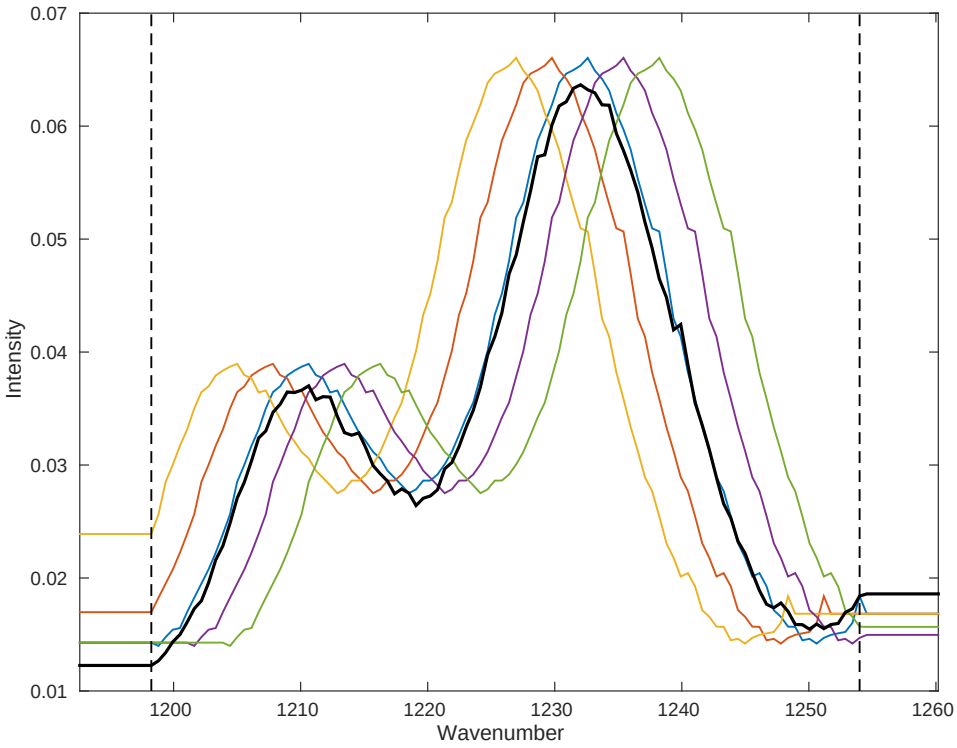


Figure 3.9: Visualisation of the process of wavenumber adjustment by moving one of the two reference spectra along the wavenumber scale.

2. Calculate the initial residual sum of squares, $(\mathbf{y}_1 - \mathbf{y}_2)^T(\mathbf{y}_1 - \mathbf{y}_2)$.

3. Extend $\mathbf{y}_1$ by repeating the first and final values $|\alpha p|$ times, for $\alpha < 1$, to give length $2|\alpha p| + p$ vector

$$\tilde{\mathbf{y}}_1 = [y_{1,1}, ..., y_{1,1}, y_{1,2}, ..., y_{1,p-1}, y_{1,p}, ..., y_{1,p}]^T .$$

   In practice we found $\alpha = 0.05$ to be sufficient to find the best alignment position between $\mathbf{y}_1$ and $\mathbf{y}_2$.

4. Calculate the residual sum of squares between $\tilde{\mathbf{y}}_1$ and all positions of $\mathbf{y}_2$ by pairing $\tilde{y}_{2,1}$ with $y_{1,l}$ for $l = 1, ..., 2|0.05p| + 1$. In calculating the residual sum of squares elements of $\tilde{\mathbf{y}}_1$ without a corresponding element in $\mathbf{y}_2$ must be discarded.

5. Select the position with the minimum residual sum of squares as the optimal shift value L.

6. Adjust dataset $\mathbf{Y}_2$

   - For $L \leq |0.05p|$ remove the first $|0.05p| - L + 1$ columns and replicate the final column $|0.05p| - L + 1$ times.
   - For $L = |0.05p| + 1$ no correction is necessary.
   - For $L \geq |0.05p| + 2$ remove the final $L - |0.05p| - 1$ columns and replicate the first column $L - |0.05p| - 1$ times.

Note: Care must be taken when selecting spectra $\mathbf{y}_1$ and $\mathbf{y}_2$ to ensure they show similar spectra, or linear combinations of spectra. For example if a tablet is the scan subject then two pixels from the centre of the tablet should meet these criteria.

During the final stage of the algorithm in practice we found replicating the first or final value to cause no problems during analysis as the required shift was minimal and the tails of the spectra contained no peaks. However if a peak is close to the edge of a spectrum a wavenumber shift may cause problems by repeating a part of the spectral peak obscuring the shape and peak location. In such a situation it may make more sense to modify step 6 to

6. Adjust datasets $\mathbf{Y}_1$ and $\mathbf{Y}_2$

   - For $L \leq |0.05p|$ remove the first $|0.05p| - L + 1$ columns of $\mathbf{Y}_2$ and the final $|0.05p| - L + 1$ columns of $\mathbf{Y}_1$.
   - For $L = |0.05p| + 1$ no correction is necessary.

- For $L \geq |0.05p| + 2$ remove the final $L - |0.05p| - 1$ columns of $\mathbf{Y}_2$ and the first $L - |0.05p| - 1$ columns of $\mathbf{Y}_1$.

Returning to figure 3.9 we see the cut regions defined in step 6 of algorithm 3.6.1 as dashed lines and the extended tails of the coloured spectra. To test algorithm 3.6.1 we applied it to data from both sides of a bicalutamide tablet, §1.2.2. Our results are shown in figure 3.8 where we have shifted the spectra detected on the lower tablet so peaks align with the upper tablet.

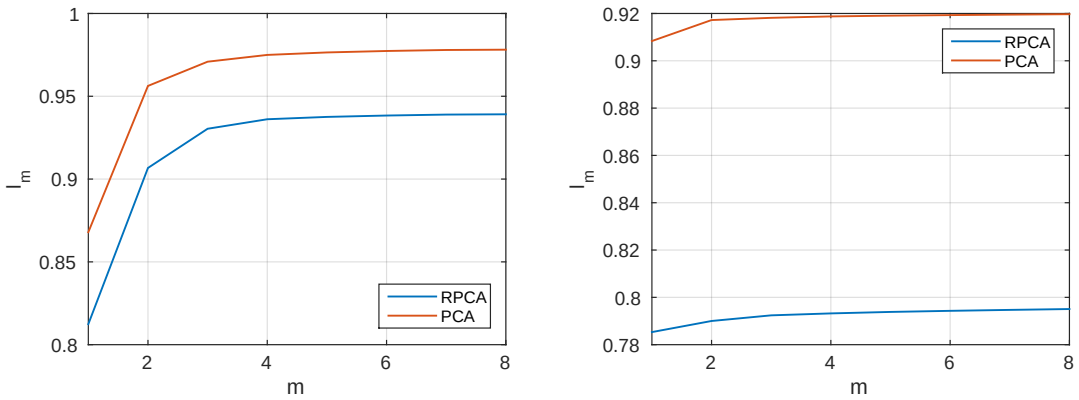
3.7 Chemical Rank Analysis using Robust Principal Components

Whilst not strictly pre-processing chemical rank analysis is an important method we use before spectral decomposition to estimate the number of component chemicals in a dataset, as discussed in §2.2. Conventional principal component rank analysis however is susceptible to anomalies in the dataset and may therefore prove unreliable if pre-processing is not performed or is not entirely successful. We therefore investigate the possibility of using the robust principal components analysis algorithm (RPCA), see §2.3.3. As with standard chemical rank analysis we use the robust PCA algorithm to estimate the eigenvalues of $\frac{\mathbf{Y}^T\mathbf{Y}}{n}$, for data matrix $\mathbf{Y}$. With eigenvalues, λ_j, such that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$ we calculate the vector, $\mathbf{l}$, of cumulative sums

$$l_m = \sum_{j=1}^{m} \frac{\lambda_j}{\sum_{J=1}^{p} \lambda_J} \tag{3.7.1}$$

to be used for identifying the number of component chemicals, as detailed in §2.2.

To compare chemical rank analysis using PCA and RPCA we use two datasets for testing. The first being the bicalutamide tablet dissolving over time and the second the scan of a human tooth, both described in §1.2.2. The former dataset contains three component spectra, the initial tablet extrudate, bicalutamide type I and bicalutamide type II. The latter contains only enamel and dentine, both of which have very similar Raman spectra. Applying chemical rank analysis to each dataset we expect the variance proportion of the eigenvectors to plateau after the first three and two eigenvalues respectively. We see in figure 3.10a that for the bicalutamide data we achieve this plateau after three eigenvalues using both methods. In figure 3.10b our results for the human tooth show evidence against using RPCA. Whilst both calculations of (3.7.1) plateaued after a single eigenvalue the variance proportion attributed to the eigenvectors by RPCA was substantially lower. We ideally want over 90% of variance to be explained by principal components before we identify



(a) Bicalutamide dataset **(b)** Human tooth dataset

Figure 3.10: Plots of (3.7.1) for $m = 1, ..., 8$, when the eigenvalues are calculated using both PCA, §2.3, and RPCA, §2.3.3. Chemical rank analysis indicates three spectra in the Bicalutamide dataset with both methods. The human tooth dataset however is shown to have either one or two spectra, with two being the correct value, giving evidence against RPCA.

that as the number of component spectra [99]. We also found that in applying the robust principal component analysis algorithm our computation time was dramatically increased.

3.8 Discussion

Our aim throughout this chapter was to derive techniques to be applied to Raman spectroscopy data before any spectral decomposition. Our first area of investigation was on the removal of specific local anomalies manifesting as unusually large elements or entire observations in the data matrix. To correct both of these types of anomalies we used techniques similar to Winsorising. In the case of an unusually large element, or cosmic ray anomaly, we remove the outlier by taking an average of a small group of values within the observation and around the anomaly. For an overly high observation, or burnt pixel, we use the spatial information contained in a Raman image and replace the entire pixel with an average of the surrounding pixels. Both of these techniques proved very successful in testing on our paracetamol and human tooth datasets and have the added benefit of being more simple to implement and requiring less time with the spectrometer than current methods [76, 83, 98].

We then turned our attention to global anomalies. The first of these, excessive levels of noise in the image, required the data to be scaled by the variance. We implemented scaling using both the standard deviation and the square root of the mean, equal to the variance of

a Poisson distribution. Whilst the latter of these proved unsuccessful in comparison to the results Keenan and Kotula [57] achieved with mass spectra, scaling by the standard deviation performed as we expected. Standard deviation scaling produced considerably clearer spectra obtained after solving (1.2.3) at the expense of detail in our estimated concentrations. The second global anomaly of interest was an incorrect baseline, or linear trend, in the data, potentially caused by degradation of the scan subject or interference from background lighting. A standard linear model proved effective in detecting the shape of the incorrect baseline however using the residuals of this model as corrected data resulted in a large proportion of the dataset becoming negative. We therefore modified our linear model to include a penalty term forcing the residuals to remain largely positive which in turn reduced the effect of spectral peaks on the model and so reduced any minimisation in peaks after correction. With this new model fit using numerical optimisation we were able to use the residuals as corrected data to great effect with very little of the incorrect baseline remaining. As with the cosmic ray anomaly removal method we proposed our baseline correction method was more simple to implement than currently published methods [20, 43, 106]. Future work on baseline correction could include investigations into a method which will still allow log transformation of the data, discussed in §1.2, due to complete avoidance of negative values. Whilst this is possible with our current method the extremely large penalisation parameter that would be required may adversely effect the spectral shape.

Our final two areas for investigation were methods applied to Raman spectroscopy data before spectral decomposition for reasons other than anomaly removal. We first looked at adjustments to the wavenumber scale which would make direct comparison of datasets possible. Although spectrometers are calibrated before each use to align the detected spectra to the correct wavenumber scale these are not always exact. Therefore two datasets with identical component chemicals may not show the same wavenumber for peaks in the spectra. Therefore the accuracy of results obtained from pooled datasets, to be used in Chapter 5, is drawn into question. We corrected this by moving a reference spectra along the wavenumber scale and calculating the Euclidean distance between that spectra and a sample from the other dataset until the minimum was found. This proved to be a fast and effective process with minimal modification to the data. Finally we looked at a possible improvement to the technique of chemical rank analysis, the standard method for estimating the number of component chemicals in a Raman image. Our proposed method used a robust principal component analysis algorithm in place of conventional PCA with the idea that it would be less effected by anomalies. However in testing RPCA proved to not only be slower but also caused a large decrease in the explained variance for each principal component, meaning we were unable to draw conclusions on chemical rank.

Chapter 4

Exploring the Solution Space

4.1 Objectives of Exploring the Solution Space

Our first form of analysis of a Raman image, $\overset{n\times p}{\mathbf{Y}}$, is usually the decomposition of the image into r component spectra, $\overset{p\times r}{\mathbf{S}}$, and their concentrations $\overset{n\times r}{\mathbf{C}}$. This is equivalent to directly finding solutions to

$$\mathbf{Y} = \mathbf{C}\mathbf{S}^T + \mathbf{E} \tag{1.2.2}$$

by minimising

$$\underset{\mathrm{S,C}}{\operatorname{argmin}} \left\|\mathbf{Y} - \mathbf{C}\mathbf{S}^T\right\|$$

subject to

$$\begin{array}{lll} 1. & \mathbf{s}_k^T\mathbf{s}_k = 1 & \text{for all } k = 1, ..., r \\ 2. & s_{j,k} \geq 0 & \text{for all } j = 1, ..., p \text{ and } k = 1, ..., r \\ 3. & c_{i,k} \geq 0 & \text{for all } i = 1, ..., n \text{ and } k = 1, ..., r \end{array} \tag{1.2.5}$$

or by first finding a solution to

$$\mathbf{Y} = \mathbf{G}\mathbf{H}^T + \mathbf{E} \tag{2.1.1}$$

by finding

$$\underset{\mathrm{H,G}}{\operatorname{argmin}} \left\|\mathbf{Y} - \mathbf{G}\mathbf{H}^T\right\|,$$

where, $\mathbf{H} = [\mathbf{h}_1 \ \ldots \ \mathbf{h}_r]$ with the constraint that

$$1. \qquad \mathbf{H}^T\mathbf{H} = \mathbf{I}_r \tag{2.1.2}$$

and transforming solutions $\mathbf{G}$ and $\mathbf{H}$ by some invertible $r \times r$ matrix $\mathbf{R}$ to give $\mathbf{C} = \mathbf{G}\mathbf{R}^{-1}$ and $\mathbf{S}^T = \mathbf{R}\mathbf{H}^T$ conforming to (1.2.5). This process is described in §1.2 and §2.4. We

also define $\|\cdot\|$ as the standard Euclidean norm, or an extension of the Euclidean norm to matrices,

$$\|\mathbf{X}\|^2 = \text{tr}\left(\mathbf{X}^T\mathbf{X}\right)$$

for row $\mathbf{x}_i^T$ of $\mathbf{X}$.

Due to the non-identifiability of (1.2.2) we have a range of possible solutions, all of which minimise the function however to a chemist their physical interpretation may vary. Therefore defining a solution space will allow us greater understanding of the solutions available from (1.2.2). For example the solution space will prove extremely useful in hypothesis testing data sets; an area investigated in Chapter 5.

One of the first techniques for solving (1.2.2) was Lawton's method of self modelling curve resolution (SMCR) [62], detailed in §2.4. The method was initially defined to only solve problems for two component spectra, $r = 2$, however we will expand upon this later in this chapter. Lawton's method involves two steps, the first to solve (2.1.1) with the second transforming these solutions such that they solve (1.2.2). We can write the two steps mathematically as,

1. Calculate the eigenvectors, $\mathbf{k}_1, ..., \mathbf{k}_p$, of the covariance matrix, $\frac{\mathbf{Y}^T\mathbf{Y}}{n}$.
2. Find linear combinations of the two primary eigenvectors, $\mathbf{k}_1$ and $\mathbf{k}_2$, to give two spectra

$$\mathbf{s}_1 = \xi_{1,1}\mathbf{k}_1 + \xi_{2,1}\mathbf{k}_2$$
$$\mathbf{s}_2 = \xi_{1,2}\mathbf{k}_1 + \xi_{2,2}\mathbf{k}_2$$

conforming to the constraints,

- $\mathbf{s}_{k,j} \geq 0$ for all $k \in [1, 2]$ and $j \in [1, p]$
- $\|\mathbf{s}_k\| = 1$
- $c_{i,1}$ and $c_{i,2}$ such that $\mathbf{y}_i = c_{i,1}\mathbf{s}_1 + c_{i,2}\mathbf{s}_2$ with $\mathbf{c}_{k,i} \geq 0$ for all $k \in [1, 2]$ and $i \in [1, n]$.

where $\|\cdot\|$ is the Euclidean norm. The values $\xi_{l,m}$ in step 2 of this process are chosen from an analytically defined solution space defined as

$$\mathcal{S} = \Big\{\mathbf{r} : \mathbf{R} = [\mathbf{r}_1 \dots \mathbf{r}_r],\ \mathbf{HR} = \mathbf{S},\ s_{j,k} \geq 0\ \forall\ j = 1, ..., p \text{ and } k = 1, ..., r, \dots$$
$$\dots\ \mathbf{C} = \mathbf{G}\left(\mathbf{R}^{-1}\right)^T,\ c_{i,k} \geq 0\ \forall\ i = 1, ..., n \text{ and } k = 1, ..., r\Big\}$$

which we will investigate later in this chapter. However we must first prove that the solution space defined by Lawton and Sylvestre [62] is actually the solution space for (1.2.2). To

prove this we must show that the eigenvectors, $\mathbf{k}_r$, used in step 1 are a least squares solution to (2.1.1), with $\mathbf{H} = [\mathbf{k}_1\ \mathbf{k}_2]$ and some $\mathbf{G}$ such that $\mathbf{Y} = \mathbf{G}\mathbf{H}^T$. Therefore after transforming $\mathbf{S}^T = \mathbf{R}^T\mathbf{H}^T$ and $\mathbf{C} = \mathbf{G}\left(\mathbf{R}^{-1}\right)^T$ the minimisation of (1.2.2) is achieved. Once we have confirmed that the solution space boundaries defined by Lawton apply to (1.2.2) we will build on them to define a solution space for datasets with more than two component spectra.

Tauler [95] and Garrido et al. [35] have worked on the exploration of the solution space using numerical optimisation. Their methods involve restricting the transformation matrix $\mathbf{R}$, in $\mathbf{Y} = \mathbf{C}\mathbf{R}\mathbf{R}^{-1}\mathbf{S}^T$ as in §1.2.1, to reduce the range of values it may take. Investigated restriction methods include forcing a unimodal spectrum, useful if only a single peak is present, and constraining specific regions of $\mathbf{C}$ to equal zero, useful if areas of the scan subject are comprised of only a single spectrum as the concentrations for the other spectra will equal zero. Whilst their results give a range of possible solutions to (1.2.2) they are more restricted than the method we propose in this chapter as they are based on specific features of the data. Our method will therefore be developed to explore the solution space of (1.2.2) regardless of the data.

4.2 Proof that Eigenvectors Minimise the Bilinear Model

As described in §4.1 in order to use the solution space defined by Lawton with (1.2.2) we must first prove that solutions obtained by Lawton's method are themselves least squares solutions and so could also be obtained by directly solving (1.2.2). For solutions from Lawton's method to be true minimisers of (1.2.2) the eigenvectors on which they are based must be minimisers of (2.1.1). We therefore first consider the least squares problem of the form (2.1.1) under constraints (2.1.2).

To show the first r principal components, corresponding to the greatest r eigenvalues, are the optimal solution to (2.1.1) we must first note that if $\mathbf{H}$ are taken to be principal component loadings then $\mathbf{G}$ will be equal to the principal component scores, $\mathbf{G} = \mathbf{Y}\mathbf{H}$, see §2.3. We now rewrite (2.1.1) as

$$\left\|\mathbf{Y} - \mathbf{G}\mathbf{H}^T\right\| = \left\|\mathbf{Y} - \mathbf{Y}\mathbf{H}\mathbf{H}^T\right\| = \|\mathbf{Y} - \mathbf{Y}\mathbf{P}\| = \|\mathbf{Y}(\mathbf{I}_p - \mathbf{P})\|$$

where $\mathbf{P} = \mathbf{H}\mathbf{H}^T$, and $\mathbf{H}^T\mathbf{H} = \mathbf{I}_r$, is a projection matrix for the new basis vectors $\mathbf{h}_r$. We

now square this to obtain

$$\begin{aligned}\|\mathbf{Y}(\mathbf{I}_p - \mathbf{P})\|^2 &= \text{tr}\left(\left[\mathbf{Y}\left(\mathbf{I}_p - \mathbf{P}\right)\right]^T\left[\mathbf{Y}\left(\mathbf{I}_p - \mathbf{P}\right)\right]\right)\\ &= \text{tr}\left(\left[\mathbf{I}_p - \mathbf{P}\right]^T\mathbf{Y}^T\mathbf{Y}\left[\mathbf{I}_p - \mathbf{P}\right]\right)\\ &= \text{tr}\left(\left[\mathbf{I}_p - \mathbf{P}\right]^T\mathbf{Y}^T\mathbf{Y}\right)\\ &= \text{tr}\left(\mathbf{Y}^T\mathbf{Y}\right) - \text{tr}\left(\mathbf{P}^T\mathbf{Y}^T\mathbf{Y}\right)\\ &= \text{tr}\left(\mathbf{Y}^T\mathbf{Y}\right) - \text{tr}\left(\mathbf{P}^T\mathbf{Y}^T\mathbf{Y}\mathbf{P}\right) \qquad (4.2.1)\end{aligned}$$

using the cyclic invariance property of trace,

$$\text{tr}\left(\mathbf{ABC}\right) = \text{tr}\left(\mathbf{BCA}\right) = \text{tr}\left(\mathbf{CAB}\right),$$

and the idempotency of

$$\mathbf{P}\mathbf{P}^T = \mathbf{H}\mathbf{H}^T\mathbf{H}\mathbf{H}^T = \mathbf{H}\mathbf{H}^T = \mathbf{P}$$

and

$$\begin{aligned}\left(\mathbf{I}_p - \mathbf{P}\right)^T\left(\mathbf{I}_p - \mathbf{P}\right) &= \mathbf{I}_p^T\mathbf{I}_p - \mathbf{I}_p\mathbf{P} - \mathbf{P}^T\mathbf{I}_p + \mathbf{P}^T\mathbf{P}\\ &= \mathbf{I}_p - \mathbf{P} - \mathbf{P} + \mathbf{P}\\ &= \mathbf{I}_p - \mathbf{P}.\end{aligned}$$

We can see this mimics ANOVA as it is of the form $\text{SS}_{\text{residual}} = \text{SS}_{\text{total}} - \text{SS}_{\text{explained}}$.

Now we calculate the principal components of $\mathbf{Y}$ which are the columns of $\mathbf{A}$ where $\mathbf{Y}^T\mathbf{Y} = \mathbf{A}\mathbf{D}\mathbf{A}^T$. We therefore have eigenvectors $\mathbf{A} = [\mathbf{a}_1\ ...\ \mathbf{a}_p]$ and eigenvalues $d_1 \geq d_2 \geq ... \geq d_p$ on the diagonal of $\mathbf{D}$. We can therefore minimise (4.2.1) by maximising

$$\begin{aligned}\max \text{tr}\left(\mathbf{P}^T\mathbf{Y}^T\mathbf{Y}\right) &= \max_{\mathbf{H}} \text{tr}\left(\mathbf{H}\mathbf{H}^T\mathbf{Y}^T\mathbf{Y}\mathbf{H}\mathbf{H}^T\right)\\ &= \max_{\mathbf{H}} \text{tr}\left(\mathbf{H}^T\mathbf{Y}^T\mathbf{Y}\mathbf{H}\right)\\ &= \max_{\mathbf{H}} \text{tr}\left(\mathbf{H}^T\mathbf{A}\mathbf{D}\mathbf{A}^T\mathbf{H}\right)\\ &= \max_{\mathbf{H}} \text{tr}\left(\mathbf{A}^T\mathbf{H}\mathbf{H}^T\mathbf{A}\mathbf{D}\right). \qquad (4.2.2)\end{aligned}$$

Now to maxmise (4.2.2) we note that, $\mathbf{D}$ is diagonal and $\mathbf{A}^T\mathbf{H}\mathbf{H}^T\mathbf{A}$ is square. Therefore the trace of (4.2.2) can be written as the sum of the diagonal elements of $\mathbf{A}^T\mathbf{H}\mathbf{H}^T\mathbf{A}$ multiplied by those of $\mathbf{D}$.

$$\text{tr}\left(\mathbf{H}^T\mathbf{A}\mathbf{D}\mathbf{A}^T\mathbf{H}\right) = \sum_{k=1}^{r}\left(\mathbf{A}^T\mathbf{H}\mathbf{H}^T\mathbf{A}\right)_{k,k} d_{k,k}. \qquad (4.2.3)$$

With both $\mathbf{A}$ and $\mathbf{H}$ comprised of orthonormal vectors we know $A_{j,j'} \leq 1$ and $H_{j,k} \leq 1$ for $j = 1, ..., p$ and $k = 1, ..., r$. We must therefore maximise the diagonal elements of $\left(\mathbf{H}^T\mathbf{A}\right)$

and $\left(\mathbf{A}^T\mathbf{H}\right)$ which equal $\mathbf{h}_k^T\mathbf{a}_k$, for $k = 1, ..., r$. As the $\mathbf{h}_k$ and $\mathbf{a}_k$ are both normalised vectors the maximum value of $\mathbf{h}_k^T\mathbf{a}_k$ is equal to 1 which is achieved when $\mathbf{h}_k = \mathbf{a}_k$ giving us

$$\mathbf{H} = \left[\begin{array}{ccc} \mathbf{a}_1 & \dots & \mathbf{a}_r \end{array} \right]$$

for eigenvectors $\mathbf{a}_j$ ordered corresponding to the size of the related eigenvalue as usual. With this we have shown that the r vectors, $\mathbf{h}_k$, needed to minimise (2.1.1) are equal to the r eigenvectors of $\mathbf{Y}^T\mathbf{Y}$, $\mathbf{a}_k$, corresponding to the r largest eigenvalues.

We can now select some invertible transformation matrix, $\mathbf{R}$, such that $\mathbf{S} = \mathbf{HR}$ and the minimisation of (2.1.1) is retained. We use this transformation to replace constraint (2.1.2) with (1.2.5) giving us physically plausible spectra. Solutions found using Lawton's method are therefore least squares solutions to (1.2.2) and we can use the solution space defined by Lawton with least squares based methods.

A very simple example of the relationship between (1.2.2) and (2.1.1) is the one-spectrum case where $r = 1$ in (1.2.2) and (2.1.1) leading to $r \times 1$ vectors $\mathbf{s}$ and $\mathbf{h}$ and scalars c and g. We define the models

$$\mathbf{y}_i = c\mathbf{s} + \varepsilon_i$$
$$\mathbf{y}_i = g\mathbf{h} + \varepsilon_i$$

to be fit subject to constraints (2.1.2) and (1.2.5) respectively. Through transformational ambiguity [12, 53], detailed in §1.2.1, there exists some transformation such that

$$\mathbf{y}_i = c\mathbf{s} = gt^{-1}t\mathbf{h} + \varepsilon_i$$

such that

$$\begin{aligned} \mathbf{s} &= \mathbf{h}t^T \\ c &= gt^{-1}. \end{aligned} \tag{4.2.4}$$

We now see that $t = 1$, i.e. this model is identifiable, using the constraint

$$\|\mathbf{s}\| = 1$$

as any $t \neq 1$ would result in $\|\mathbf{s}\| = t$. We therefore have $\mathbf{s} = \mathbf{h}$, our single component spectra is equal to the first principal component of $\mathbf{Y}^T\mathbf{Y}$, and $c = g$.

We can also see the relationship between between (1.2.2) and (2.1.1) from the ALS, §2.6.1, algorithm for solving (1.2.2). We can show that the alternating steps of this algorithm, in the most basic case with no thresholding of estimated values below zero, only depend on

the initial estimate of $\mathbf{C}$ and the data $\mathbf{Y}$ with the following iterative formulae:

$$\begin{aligned}\mathbf{C}_{(l)}^{T} = \mathbf{C}_0^T\mathbf{C}_0\left(\mathbf{C}_0^T\mathbf{U}\mathbf{D}^1\mathbf{U}^T\mathbf{C}_0\right)^{-1}\mathbf{C}_0^T\mathbf{U}\mathbf{D}^2\mathbf{U}^T\mathbf{C}_0\left(\mathbf{C}_0^T\mathbf{U}\mathbf{D}^3\mathbf{U}^T\mathbf{C}_0\right)^{-1}\ldots \\ \ldots\ \mathbf{C}_0^T\mathbf{U}\mathbf{D}^{2l-2}\mathbf{U}^T\mathbf{C}_0\left(\mathbf{C}_0^T\mathbf{U}\mathbf{D}^{2l-1}\mathbf{U}^T\mathbf{C}_0\right)^{-1}\mathbf{C}_0^T\mathbf{U}\mathbf{D}^{l}\mathbf{U}^T\end{aligned} \tag{4.2.5}$$

and

$$\mathbf{S}_{(l)}^{T} = \begin{cases} \left(\mathbf{C}_0^T\mathbf{C}_0\right)^{-1}\mathbf{C}_0\mathbf{Y} & \text{if } l = 1 \\ \left(\mathbf{C}_0^T\mathbf{C}_0\right)^{-1}\mathbf{C}_0\mathbf{U}\mathbf{D}^1\mathbf{U}^T\mathbf{C}_0\left(\mathbf{C}_0^T\mathbf{U}\mathbf{D}^2\mathbf{U}^T\mathbf{C}_0\right)^{-1}\mathbf{C}_0^T\mathbf{U}\mathbf{D}^3\mathbf{U}^T\mathbf{C}_0\ \ldots & \\ \qquad \ldots\ \left(\mathbf{C}_0^T\mathbf{U}\mathbf{D}^{2l-2}\mathbf{U}^T\mathbf{C}_0\right)^{-1}\mathbf{C}_0^T\mathbf{U}\mathbf{D}^{l-1}\mathbf{U}^T\mathbf{Y} & \text{if } l > 1 \end{cases} \tag{4.2.6}$$

(4.2.7)

where (l) is the iteration, $\mathbf{C}_{(0)}$ is the initial value for $\mathbf{C}$, usually chosen to be random numbers [55], and $\mathbf{U}$ and $\mathbf{D}$ come from the singular value decomposition of $\mathbf{Y}\mathbf{Y}^T$, such that $\mathbf{D}$ is a diagonal matrix with eigenvalues $d_1 \geq d_2 \geq \ldots \geq d_n$ on the diagonal and

$$\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ \ldots\ \mathbf{u}_n]$$

where the $\mathbf{u}_i$ are orthonormal eigenvectors ordered corresponding to the d_i.

We can see a simple intuitive limit due to the diagonality of $\mathbf{D}$, the power to which it is raise and the property that eigenvalues represent the proportion of variance explained by the corresponding eigenvector [73]. As used in chemical rank analysis, detailed in §2.2, a sample with r component spectra will have the largest r eigenvalues accounting for the majority of the variance. Therefore for increasing powers of $\mathbf{D}$ the larger eigenvalues will dominate the matrix giving greater weight in the final solution to the r eigenvectors corresponding to the r largest eigenvectors. We therefore see that the solution obtained by any alternating least squares based method finds a solution based on the same r eigenvectors as used by methods which first solve (2.1.1) such as Lawton's original method, §2.4.

4.3 Lawton's Solution Space Bounds

We now investigate the boundaries of the solution space for $\xi_{l,m}$ used in step 2 of Lawton's algorithm, §4.1 [62]. We will first summarise the boundaries defined by Lawton and Sylvestre [62] and then show that

- Boundaries always exist from within which we select $\xi_{l,m}$ such that $s_{j,k} \geq 0$ for all $j \in [1, p]$ and $k \in [1, r]$.

- Boundaries always exist from within which we select $\xi_{l,m}$ such that $c_{i,k} \geq 0$ for all $i \in [1, n]$ and $k \in [1, r]$.
- A region of $\xi_{l,m}$ exists such that both the previous constraints are met.
- This region is bounded so exploration will be viable

for

$$\mathbf{s}_1 = \xi_{1,1}\mathbf{k}_1 + \xi_{2,1}\mathbf{k}_2$$
$$\mathbf{s}_2 = \xi_{1,2}\mathbf{k}_1 + \xi_{2,2}\mathbf{k}_2$$

and

$$\mathbf{y}_i = c_{i,1}\mathbf{s}_1 + c_{i,2}\mathbf{s}_2 + \boldsymbol{\varepsilon}_i$$

with $\mathbf{k}_k$ being the eigenvectors of $\frac{\mathrm{Y}^T\mathrm{Y}}{n}$ with the largest eigenvalues, $\xi_{l,m}$ linear combination coefficients and $c_{i,k}$ concentrations corresponding to the spectra.

4.3.1 Non-negative Spectra

Lawton's first solution space bound corresponds to the constraint that a spectrum contains no negative elements. Lawton began with the definition of a spectrum as a linear combination of primary eigenvectors $\mathbf{k}_1$ and $\mathbf{k}_2$,

$$\mathbf{s}_k = \xi_{1,k}\mathbf{k}_1 + \xi_{2,k}\mathbf{k}_2 \quad \text{for } k \in [1, 2]. \tag{4.3.1}$$

To ensure every element in vector $\mathbf{s}_k$ is non-negative Lawton restricted the choice of $\xi_{1,k}$ and $\xi_{2,k}$ from the (ξ_1, ξ_2) plane such that all linear combinations produce non-negative vectors. The boundaries of this region were obtained using the knowledge that the first eigenvector, $\mathbf{k}_1$, has elements of the same sign, see §2.3.1, and as a result of this the second eigenvector, $\mathbf{k}_2$ will contain a mixture of positive and negative elements. This is clear if we assume $\mathbf{k}_2$ also has elements of the same sign as we see that

$$\sum_{j=1}^{p} k_{1,j}k_{2,j} > 0$$

and $\mathbf{k}_2$ is therefore not an eigenvector. With this Lawton noted that $\mathbf{k}_1$ will have a positive linear combination coefficient, ξ_1, otherwise $\xi_1\mathbf{k}_1 + \xi_2\mathbf{k}_2$ would contain negative elements due to an entirely negative $\xi_1\mathbf{k}_1$ and partially negative $\xi_2\mathbf{k}_2$.

With this Lawton defined that for a given value of ξ_1 the greatest negative value ξ_2 may take will be equal to

$$-\min_{j \in K^+} \left| \frac{k_{1,j}}{k_{2,j}} \right| \xi_1$$

for $K^+ = \{j : \mathbf{k}_{2,j} > 0\}$. Likewise for a given ξ_1 that the greatest positive value ξ_2 can take is equal to

$$\min_{j \in K^-} \left| \frac{k_{1,j}}{k_{2,j}} \right| \xi_1$$

for $K^- = \{j : \mathbf{k}_{2,j} \leq 0\}$. This leads to the boundaries

$$\xi_1 \geq 0 \tag{4.3.2}$$

and

$$\begin{aligned} \xi_2 &\geq \zeta \xi_1 \\ \xi_2 &\leq \tau \xi_1 \end{aligned} \tag{4.3.3}$$

where

$$\begin{aligned} \zeta &= -\min_{j \in K^+} \left| \frac{k_{1,j}}{k_{2,j}} \right| \\ \tau &= \min_{j \in K^-} \left| \frac{k_{1,j}}{k_{2,j}} \right|. \end{aligned} \tag{4.3.4}$$

The constraints (4.3.3) equate to two straight lines on the (ξ_1, ξ_2) plane, originating from the origin covering only the positive ξ_1 axis.

We now show that these boundaries cannot encompass the entire $\xi_1 \geq 0$ region as this implies

$$\zeta \to \infty$$
$$\tau \to \infty$$

This limit would require $k_{2,j} = 0$ for some $j = 1, ..., p$ which requires linearly dependent columns in $\mathbf{Y}$. As our data contains random noise linear dependence between columns will happen with probability zero. Therefore $-\infty < \zeta$ and $\tau < \infty$. For this same reason the gradients are also unable to equal zero as this would require some $k_{1,j} = 0$. Examples of these constraints are the blue region boundaries in figure 2.3a.

4.3.2 Non-negative Concentrations

Lawton derived the set of boundaries for the solution space from the assumption that the concentration matrix $\mathbf{C}$ contains no negative elements. Lawton began by establishing a link between the linear combination coefficients, $(\xi_{1,k}, \xi_{2,k})$, for spectra and the concentrations by defining

$$\mathbf{S} = \mathbf{K}\Xi$$

for

$$\mathbf{S} = [\mathbf{s}_1\ \mathbf{s}_2], \quad \mathbf{K} = [\mathbf{k}_1\ \mathbf{k}_2], \quad \Xi = \begin{bmatrix} \xi_{1,1} & \xi_{1,2} \\ \xi_{2,1} & \xi_{2,2} \end{bmatrix}.$$

This gives

$$\begin{aligned} & \mathbf{y}_i = \mathbf{S}\mathbf{c}_i = \mathbf{K}\Xi\mathbf{c}_i \\ \Rightarrow\ & \Xi^{-1}\mathbf{K}^T\mathbf{y}_i = \mathbf{c}_i \\ \Rightarrow\ & \frac{1}{|\Xi|}\begin{bmatrix} \xi_{2,2} & -\xi_{1,2} \\ -\xi_{2,1} & \xi_{1,1} \end{bmatrix}\begin{bmatrix} \mathbf{k}_1^T\mathbf{y}_i \\ \mathbf{k}_2^T\mathbf{y}_i \end{bmatrix} = \mathbf{c}_i \\ \Rightarrow\ & \begin{bmatrix} c_{i,1} \\ c_{i,2} \end{bmatrix} = \frac{1}{|\Xi|}\begin{bmatrix} \xi_{2,2}\mathbf{k}_1^T\mathbf{y}_i - \xi_{1,2}\mathbf{k}_2^T\mathbf{y}_i \\ -\xi_{2,1}\mathbf{k}_1^T\mathbf{y}_i + \xi_{1,1}\mathbf{k}_2^T\mathbf{y}_i \end{bmatrix} = \frac{1}{|\Xi|}\begin{bmatrix} \xi_{2,2}\kappa_{i,1} - \xi_{1,2}\kappa_{i,2} \\ -\xi_{2,1}\kappa_{i,1} + \xi_{1,1}\kappa_{i,2} \end{bmatrix} \end{aligned} \tag{4.3.5}$$

Now from (4.3.5) a single coordinate on the solution space, $(\xi_{1,k}, \xi_{2,k})$, gives separate concentration values

$$\begin{aligned} c_{i,1} &= \frac{1}{|\Xi|}\left(-\xi_{1,2}\kappa_{i,2} + \xi_{2,2}\kappa_{i,1}\right) \\ c_{i,2} &= \frac{1}{|\Xi|}\left(\xi_{1,1}\kappa_{i,2} - \xi_{2,1}\kappa_{i,1}\right). \end{aligned}$$

From (4.3.5) it is clear that $\kappa_{i,1}$ is positive as it is the product of two entirely positive vectors,

$$\kappa_{i,1} = \mathbf{k}_1\mathbf{y}_i.$$

With this positivity Lawton fixed $\xi_{1,k}$ to limit the values $\xi_{2,k}$ may take using the ratio $\frac{\kappa_{i,2}}{\kappa_{i,1}}$. This led to the solution space bounds,

$$\begin{aligned} \xi_2 &\geq \max_{1\leq i\leq n}\left[\frac{\kappa_{i,2}}{\kappa_{i,1}}\right]\xi_1 \\ \xi_2 &\leq \min_{1\leq i\leq n}\left[\frac{\kappa_{i,2}}{\kappa_{i,1}}\right]\xi_1 \end{aligned} \tag{4.3.6}$$

Finally we show that these bounds will always exist and lie within the boundaries defined in §4.3.1, i.e.

$$\begin{aligned} \max_{1\leq i\leq n}\left[\frac{\kappa_{i,2}}{\kappa_{i,1}}\right] &< \min_{j\in K^-}\left|\frac{k_{1,j}}{k_{2,j}}\right| \\ \min_{1\leq i\leq n}\left[\frac{\kappa_{i,2}}{\kappa_{i,1}}\right] &< -\min_{j\in K^+}\left|\frac{k_{1,j}}{k_{2,j}}\right|. \end{aligned}$$

We see this by noting that the boundaries (4.3.4) simply give the region of coefficients from which all linear combinations of the eigenvectors result in entirely positive vectors. As the data points, $\mathbf{y}_i$, are entirely positive and can be written as linear combinations of the

eigenvectors their coefficients must also lie within the region (4.3.4). Now as the $\kappa_{i,k}$ are the data projected onto the space spanned by $\mathbf{k}_1$ and $\mathbf{k}_2$ the boundaries (4.3.6) are lines drawn through two projected data points and will therefore lie within the previously defined region. Examples of these constraints are the black region boundaries in figure 2.3a.

4.3.3 Unit Area Under Spectra

The final constraint Lawton placed on the solution space came from the unit area restriction on the spectra, $\mathbf{s}_k$. It is important to note that whilst Lawton defined this by viewing the spectra and eigenvectors as functions, as in §2.4, this has been replaced by the unit norm constraint in all modern methods. However for completeness we will prove that this constraint is not trivial, as defined by Lawton, before reverting back to the unit norm constraint.

Lawton defined this final constraint as,

$$\int_{\Omega_1}^{\Omega_2} s_k(\omega) = 1. \tag{4.3.7}$$

where the $\Omega_1 \leq \omega \leq \Omega_2$. To obtain a solution space boundary from this Lawton and Sylvestre [62] used numerical integration to estimate this integral. By viewing the eigenvectors as functions over wavenumbers the areas

$$\int_{\Omega_1}^{\Omega_2} k_k(\omega) \quad \text{for } k = 1, 2$$

were first estimated with the trapezium rule,

$$\delta_k = \sum_{j=1}^{p} k_{j,k}\Delta_j, \qquad \text{for } k = 1, 2$$

with

$$\Delta_j = \begin{cases} |\omega_2 - \omega_1| & \text{if } j = 1 \\ |\omega_p - \omega_{p-1}| & \text{if } j = p \\ |\frac{\omega_k - \omega_{k-1}}{2}| & \text{if } 1 < j < p \end{cases},$$

and so for linear combination coefficients (ξ_1, ξ_2), for generating a spectrum, the unit area constraint, (4.3.7), becomes

$$\begin{aligned} 1 &= \delta_1\xi_1 + \delta_2\xi_2 \\ \Rightarrow \xi_2 &= \frac{1}{\delta_2} - \frac{\delta_1}{\delta_2}\xi_1 \\ &= d_1 + d_2\xi_1 \end{aligned} \tag{4.3.8}$$

for

$$\begin{aligned} d_1 &= \frac{1}{\delta_2} \\ d_2 &= -\frac{\delta_1}{\delta_2}. \end{aligned} \tag{4.3.9}$$

Existence of this non-empty subspace is clear unless we have entirely zero eigenvectors $\mathbf{k}_1$ and $\mathbf{k}_2$. We show an example of the intersection between the line defined by (4.3.8) and the previously defined regions (4.3.3) and (4.3.6) in figure 2.3b.

As the unit area constraint (4.3.7) uses a rudimentary numerical integration technique we replace it with a unit Euclidean norm as defined in (1.2.5), which replaces the line (4.3.8) with a unit arc,

$$\begin{aligned} \|\mathbf{s}_k\|^2 &= 1 \\ \Rightarrow \xi_{1,k}^2 + \xi_{2,k}^2 &= 1 \end{aligned}$$

for

$$\mathbf{s}_k = \xi_{1,k}\mathbf{k}_1 + \xi_{2,k}\mathbf{k}_2.$$

This ensures our estimated spectra will comply with the unit norm constraint with no risk of numerical estimation errors as may occur with (4.3.7). We are also assured that neither ξ_1 or ξ_2 will approach infinity as we will be selecting from an arc around the origin. This will prove useful later in this chapter when we explore the solution space with a random walk as we will know our walk will be bounded.

4.4 Lawton's Solution Space Bounds in Higher Dimensions

Having summarised the solution space defined by Lawton and Sylvestre [62] in §4.3, and proved that it will always exist for $r = 2$, we now investigate an extension of them to accommodate more than two spectra. We first consider the three spectrum case where the model now being fit is

$$\mathbf{y}_i = c_{i,1}\mathbf{s}_1 + c_{i,2}\mathbf{s}_2 + c_{i,3}\mathbf{s}_3 + \mathbf{e}_i, \qquad \text{for } i = 1, ..., n \tag{4.4.1}$$

with the same non-negativity and unit area constraints, (1.2.5). As defined in §4.3.1 and §4.3.2 the observations and the spectra are all linear combinations of the eigenvectors,

$$\begin{aligned} \mathbf{y}_i &= \kappa_{i,1}\mathbf{k}_1 + \kappa_{i,2}\mathbf{k}_2 + \kappa_{i,3}\mathbf{k}_3 \\ \mathbf{s}_1 &= \xi_{1,1}\mathbf{k}_1 + \xi_{2,1}\mathbf{k}_2 + \xi_{3,1}\mathbf{k}_3 \\ \mathbf{s}_2 &= \xi_{1,2}\mathbf{k}_1 + \xi_{2,2}\mathbf{k}_2 + \xi_{3,2}\mathbf{k}_3 \\ \mathbf{s}_3 &= \xi_{1,3}\mathbf{k}_1 + \xi_{2,3}\mathbf{k}_2 + \xi_{3,3}\mathbf{k}_3. \end{aligned} \tag{4.4.2}$$

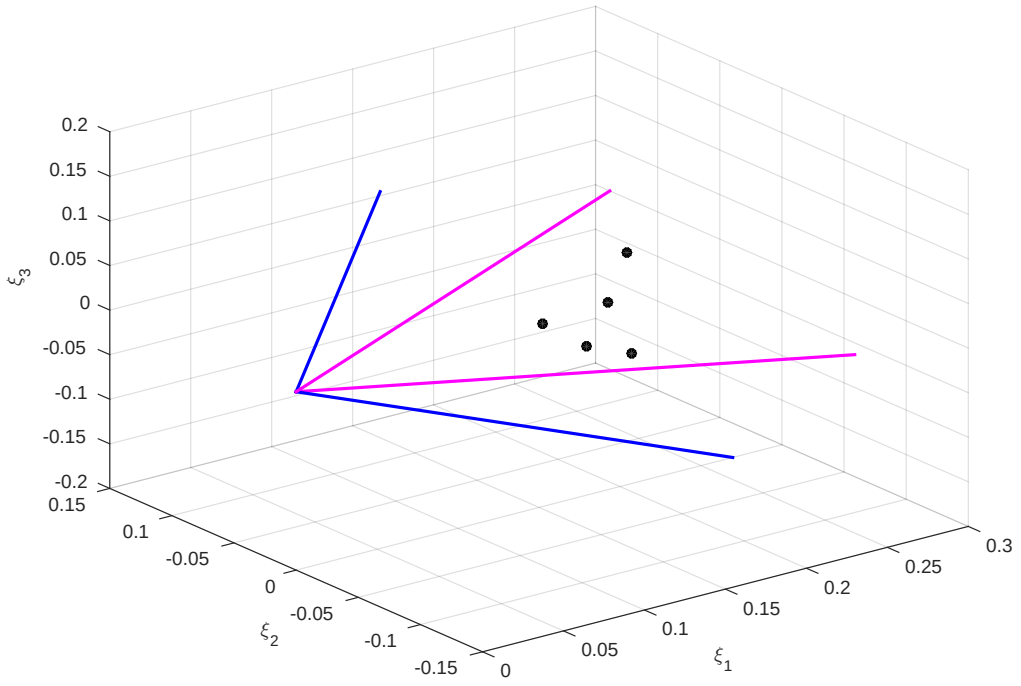


Figure 4.1: Produced using data from Lawton and Sylvestre [62] however now with $r = 3$ where Lawton used $r = 2$. The (ξ_1, ξ_2, ξ_3) space containing the solution volume in which the $\mathbf{y}_i$ are marked as five circles in terms of the $\kappa_{i,k}$ in (4.4.2). The 2-dimensional boundaries of the region resulting in non-negative $\mathbf{s}_k$, (4.4.4) assuming either $\xi_2 = 0$ or $\xi_3 = 0$, are marked by blue and magenta lines.

4.4.1 Unit Spectra

The most simple constraint to define as a boundary on our three spectra solution space, (ξ_1, ξ_2, ξ_3), is the unit norm constraint in (1.2.5). With spectra defined as

$$\mathbf{s}_k = \xi_{1,k}\mathbf{k}_1 + \xi_{2,k}\mathbf{k}_2 + \xi_{3,k}\mathbf{k}_3$$

our constraint is

$$\begin{aligned} \|\mathbf{s}_k\|^2 &= 1 \\ \Rightarrow \xi_{1,k}^2 + \xi_{2,k}^2 + \xi_{3,k}^2 &= 1 \end{aligned}$$

which is a unit sphere about the origin.

4.4.2 Non-negative Spectra

Defining the boundaries of the region from which we select our linear combination coefficients for forming spectra satisfying constraint 2 in (1.2.5) is more difficult. In the three spectra case, where $r = 3$, we have

$$\mathbf{S} = \mathbf{K}\Xi$$
$$\Rightarrow \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{k}_1 & \mathbf{k}_2 & \mathbf{k}_3 \end{bmatrix} \begin{bmatrix} \xi_{1,1} & \xi_{1,2} & \xi_{1,3} \\ \xi_{2,1} & \xi_{2,2} & \xi_{2,3} \\ \xi_{3,1} & \xi_{3,2} & \xi_{3,3} \end{bmatrix} \tag{4.4.3}$$

where each column of Ξ is a coordinate on the (ξ_1, ξ_2, ξ_3) volume. We therefore see that for a single $\mathbf{s}_k$ our constraint is equal to

$$\xi_1 k_{1,j} + \xi_2 k_{2,j} + \xi_3 k_{3,j} \geq 0 \quad \text{for } j = 1, ..., p. \tag{4.4.4}$$

The difficulty we encounter in defining the region boundaries comes from the inclusion of the third eigenvector. In the two spectrum case we could fix ξ_1 and calculate the possible values ξ_2 may take. However we now have a third coefficient, ξ_3, and so can no longer constrain the solution space in this way. Although we can still estimate parts of the region boundaries using the 2-dimensional formula to give us an indication of the solution space in the triangular shape we saw in figure 2.3. To obtain these we assume either $\xi_2 = 0$ or $\xi_3 = 0$ and use

$$\begin{aligned} \xi_1 &\geq 0 \\ \xi_2 &\geq \zeta_2 \xi_1 \\ \xi_2 &\leq \tau_2 \xi_1 \\ \xi_3 &\geq \zeta_3 \xi_1 \\ \xi_3 &\leq \tau_3 \xi_1 \end{aligned}$$

where

$$\begin{aligned} \zeta_2 &= -\min_{j \in K_2^+} \left| \frac{k_{1,j}}{k_{2,j}} \right| \\ \tau_2 &= \min_{j \in K_2^-} \left| \frac{k_{1,j}}{k_{2,j}} \right| \\ \zeta_3 &= -\min_{j \in K_3^+} \left| \frac{k_{1,j}}{k_{3,j}} \right| \\ \tau_3 &= \min_{j \in K_3^-} \left| \frac{k_{1,j}}{k_{3,j}} \right| \end{aligned} \tag{4.4.5}$$

where $K_l^+ = \{j : \mathbf{k}_{l,j} > 0\}$ and $K_l^- = \{j : \mathbf{k}_{l,j} > 0\}$. Calculating (4.4.5) using Lawton's data, described in §1.2.2, gives us the blue and magenta lines respectively in figure 4.1. When viewed from the perspective of $\xi_2 = 0$ or $\xi_3 = 0$ these are the 2-dimensional boundaries of a region of the same shape as that in figure 2.3.

With the analytical boundaries incalculable we turn to a numerical estimation of the region from which a spectrum will be entirely positive. For this we calculate the spectrum at each point of a 3-dimensional mesh, covering the (ξ_1, ξ_2, ξ_3) volume, with coordinate ranges

$$\begin{aligned}\xi_1 &= (0, 0.001, 0.002, ..., 0.399, 0.4)\\ \xi_2 &= (-0.15, -0.149, ..., 0.149, 0.15)\\ \xi_3 &= (-0.05, -0.049, ..., 0.149, 0.15).\end{aligned}$$

We then accept points from this mesh as lying in the solution space if the resulting $\mathbf{s}_k = \mathbf{K}\boldsymbol{\xi}$ has entirely positive elements. We see the results of this trial and error technique in figure 4.2 as layers of the accepted volume with the location of the layer indicated by a green plane on the 3-dimensional volume plot. The shape of the cross sections show the volume to be a more complicated multi-faceted structure with borders crossing through the calculable 2-dimensional boundaries.

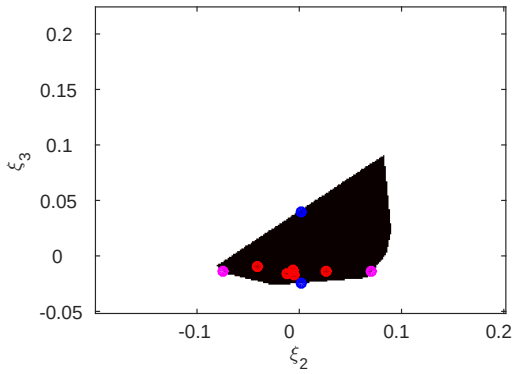
4.4.3 Non-negative Concentrations

In calculating the region which will result in non-negative concentrations we encounter the same problems as in §4.4.2 with the non-negative spectra boundaries. The inclusion of ξ_3 no longer allows us to fix ξ_1 and limit the remaining coefficient values. However we can still obtain the boundaries of the two spectrum case, assuming either $\xi_2 = 0$ or $\xi_3 = 0$, using

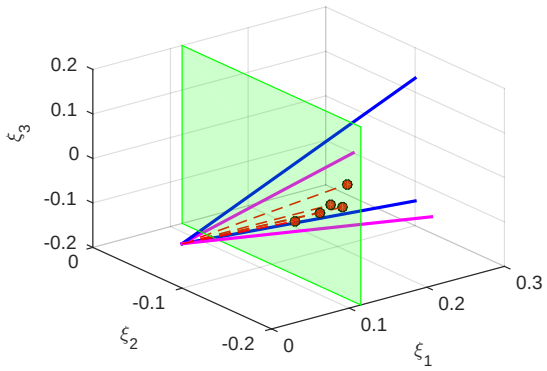
$$\begin{aligned}\xi_2 &\geq \max_{1\leq i\leq n}\left[\frac{\kappa_{i,2}}{\kappa_{i,1}}\right]\xi_1\\ \xi_2 &\leq \min_{1\leq i\leq n}\left[\frac{\kappa_{i,2}}{\kappa_{i,1}}\right]\xi_1\\ \xi_3 &\geq \max_{1\leq i\leq n}\left[\frac{\kappa_{i,3}}{\kappa_{i,1}}\right]\xi_1\\ \xi_3 &\leq \min_{1\leq i\leq n}\left[\frac{\kappa_{i,3}}{\kappa_{i,1}}\right]\xi_1\end{aligned} \tag{4.4.6}$$

where $\kappa_{i,k} = \mathbf{k}_k^T\mathbf{y}_i$. Using Lawton's data, described in §1.2.2, we show an example of these boundaries in figure 4.3 as the red and green lines respectively. When viewed from the perspective of $\xi_2 = 0$ or $\xi_3 = 0$ these are the 2-dimensional boundaries of a region of the same shape as that in figure 2.3.

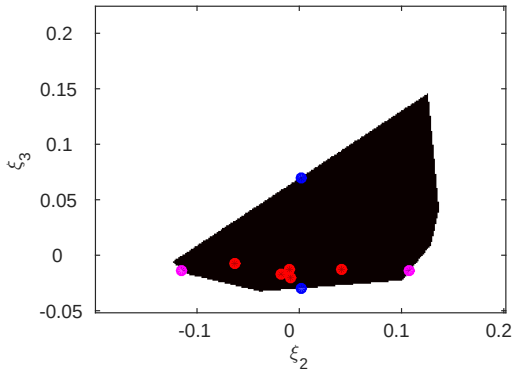
As in §4.4.2 we use a numerical method to estimate the solution space by finding points which lie within. However we come across a further problem when defining our trial and



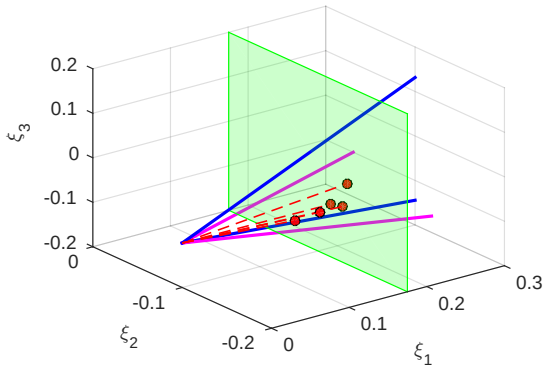
(a) Volume boundaries at $\xi_1 = 0.114$.



(b) Position of layer in volume.



(c) Volume boundaries at $\xi_1 = 0.174$.



(d) Position of layer in volume.

Figure 4.2: Figure produced using data from Lawton and Sylvestre [62], described in §1.2.2. Layers of the (ξ_1, ξ_2, ξ_3) volume in which the $\mathbf{y}_i$ are marked as red circles in terms of the $\kappa_{i,k}$, in (4.4.2). The 2-dimensional boundaries of the region resulting in non-negative $\mathbf{s}_k$, (4.4.4) assuming either $\xi_2 = 0$ or $\xi_3 = 0$, are marked by blue and magenta lines.

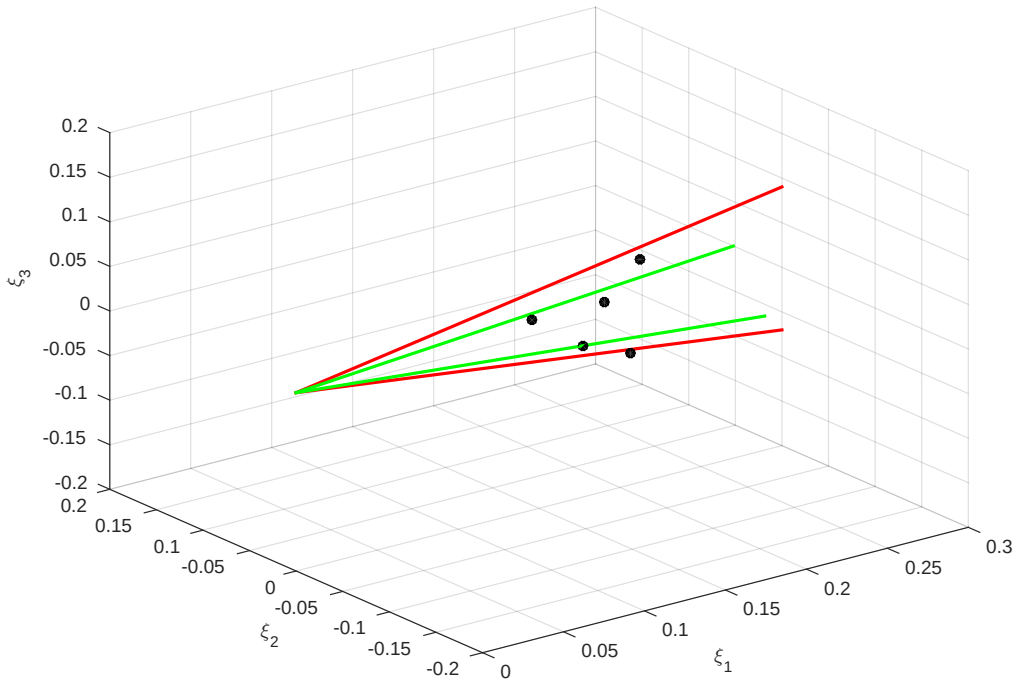


Figure 4.3: Produced using data from Lawton and Sylvestre [62] described in §1.2.2. The (ξ_1, ξ_2, ξ_3) space containing the volume in which the $\mathbf{y}_i$ are marked as five circles in terms of the $\kappa_{i,k}$, in (4.4.2). The 2-dimensional boundaries of the region resulting in non-negative $\mathbf{s}_k$, (4.4.4) assuming either $\xi_2 = 0$ or $\xi_3 = 0$, are marked by green and red lines.

error method for narrowing down the solution space. With the spectra it is possible for us to select a single point on the space, $\boldsymbol{\xi}$, and calculate a single spectrum, $\mathbf{s} = \mathbf{K}\boldsymbol{\xi}$, which we can check for negative elements. However it is not possible for us calculate concentrations, which we can check for negative elements, using a single point from the solution space. We see this in the definition of the model for a single observation,

$$\mathbf{y}_i = \mathbf{S}\mathbf{c}_i + \boldsymbol{\varepsilon}_i = \mathbf{K}\Xi\mathbf{c}_i + \boldsymbol{\varepsilon}_i$$

where to estimate the concentrations using

$$\mathbf{c}_i = \Xi^{-1}\mathbf{K}^T\mathbf{y}_i$$

we must have r points from the solution space to obtain Ξ^{-1}. Due to this checking for negative elements in the concentrations generated by each point of a 3-dimensional mesh, as in §4.4.2, is not possible. We would instead have to test all possible combinations of three points on the sub-mesh of non-negative spectra calculated in §4.4.2.

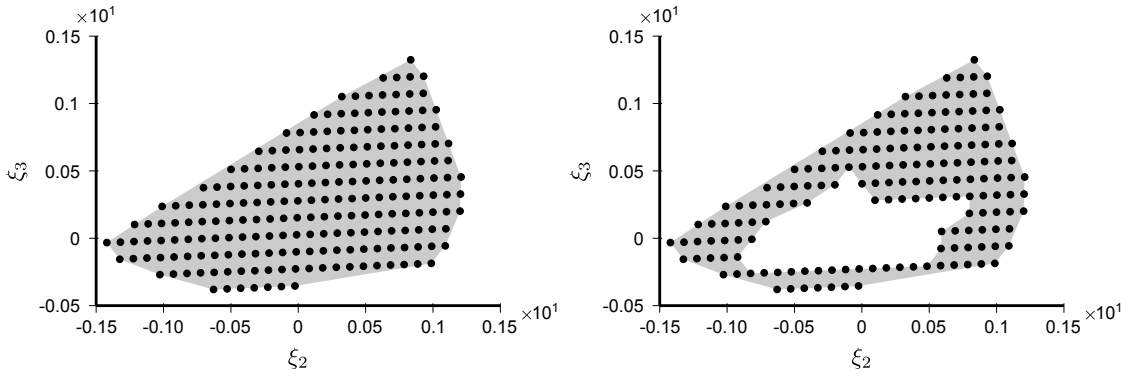
To calculate the region giving non-negative spectra in §4.4.2 for Lawton's original data [62], described in §1.2.2, we used the 3-dimensional mesh defined in §4.4.2 which gives 24 million points. To calculate the concentrations given by all combinations of three points from this mesh we would have 10^{21} point combinations, calculated with

$$\frac{24 \times 10^6!}{3!(24 \times 10^6 - 3)!}.$$

We reduce this by first restricting our points giving positive spectra to those also conforming to the unit area constraint, the plane segment shown in figure 4.4a. In doing so we reduce the number of combinations to test to 10^7. The resulting solution plane from this trial and error method for Lawton's data is shown in figures 4.4b and 4.5. Also shown in figure 4.5 as a red line is the 2-dimensional region calculated by the constraints detailed in §4.3 and seen in figure 2.3.

4.4.4 Further Extensions in Dimension

Expanding Lawton's method to find $r = 3$ spectra made clear the problems we will encounter when extending to further dimensions. Any dimension beyond two removes our ability to calculate the boundaries of the solution space. This intractable space then becomes extremely expensive to explore with the trial and error method we used in §4.4.2 and §4.4.3. As such we will consider a more efficient method of exploring this solution space using a random walk in §4.6.1.



(a) Non-negative **S**. **(b)** Non-negative **C**.

Figure 4.4: Produced using data from Lawton and Sylvestre [62] described in §1.2.2. The (ξ_1, ξ_2, ξ_3) volume and unit-S plane intercept generated using a brute force method comparing all possible combinations of $\boldsymbol{\xi}$ as described in §4.4.1, §4.4.2 and 4.4.3.

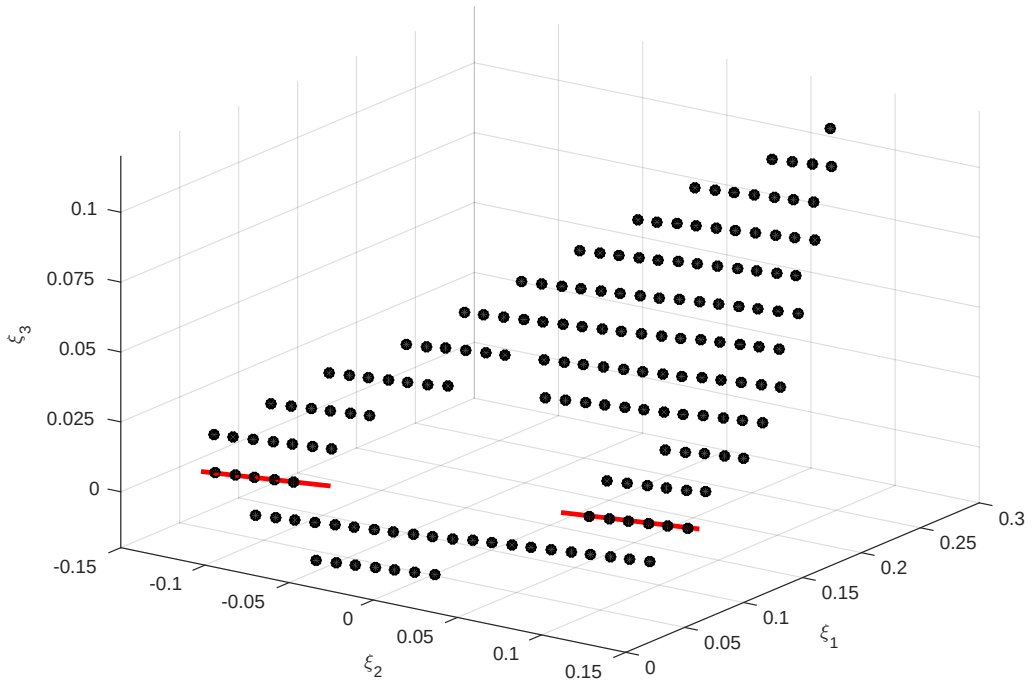


Figure 4.5: Produced using data from Lawton and Sylvestre [62] described in §1.2.2. The (ξ_1, ξ_2, ξ_3) volume and unit-S sphere intercept, viewed along the ξ_1 axis. Region generated using the trial and error method in §4.4.3 comparing all possible combinations of $\boldsymbol{\xi}$ as described in §4.4.1, §4.4.2 and 4.4.3. The red lines show the 2-dimensional solution space as defined in §4.3.

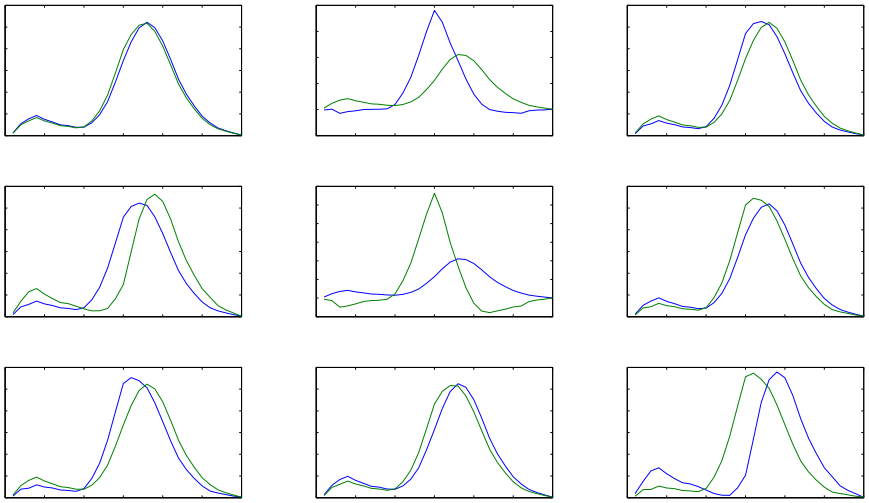


Figure 4.6: Sample results from two spectrum ALS, §2.7.1, using data from Lawton and Sylvestre [62], as described in §1.2.2, with the initial **C** generated as random integers from [1,10]. Whilst all the solutions are mathematically equal in terms of minimising (1.2.2) they are visibly very different which may cause issue when interpreting them as chemical spectra.

4.5 Variability of Results from the Solution Space

Due to the non-identifiability in (1.2.2), and the location of a solution given by any chosen algorithm depending on the initial values given for **S** and **C** as we show with (4.2.5), the results we obtain can vary. As a simple example of this we applied ALS, §2.7.1, with randomly generated starting values, to Lawton's original dataset [62]. We use random initial values under the assumption that for any starting point the algorithm will converge to equally good solutions which will differ due to the non-identifiability. By repeatedly solving (1.2.2) for two and three component spectra, $r = 2$ and $r = 3$ we obtained the estimated spectra shown in figures 4.6 and 4.7. where the instability in the results to (1.2.2) is clear.

We show a simple worst case scenario of the non-identifiability of (1.2.2) using our exploration results of Lawton's data [62] from §4.4. Figures 4.8 and 4.9 show two possible sets of solutions to (1.2.2). We see an example of physically plausible results in figure 4.8a where we have the two true spectra in Lawton's data [62] and a third which appears to be a combination of the other two and was only found because we searched for three components in data containing only two. Figure 4.9 however shows us the problem with selecting three points close together in the solution space. Here we have three spectra coming from the solution space of (1.2.2) with none looking like the true component spectra in the data.

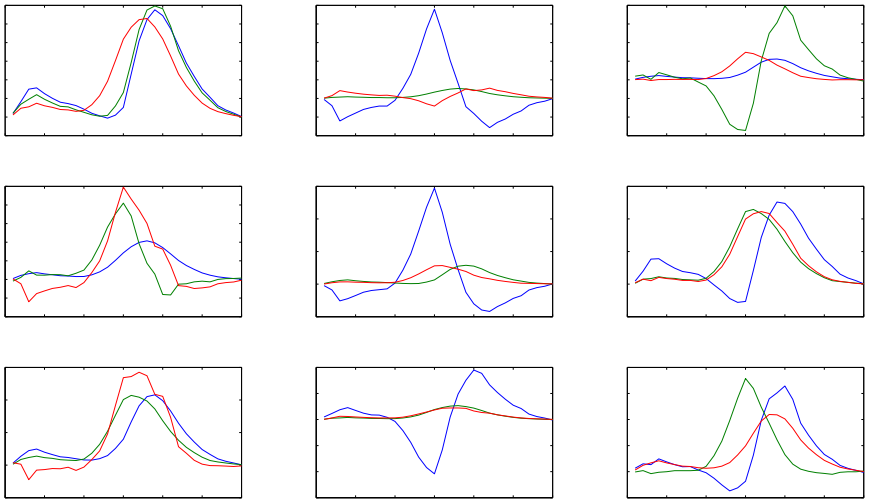
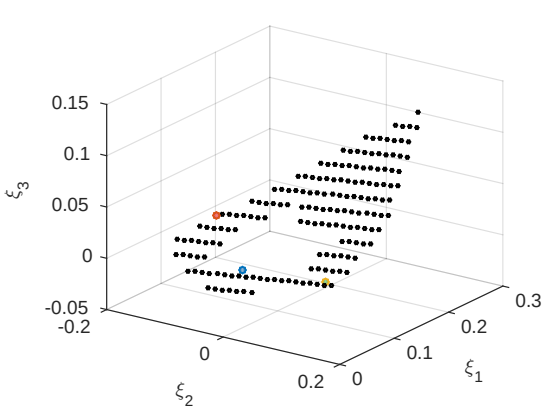


Figure 4.7: Sample results from three spectrum ALS, §2.7.1, using data from Lawton and Sylvestre [62], as described in §1.2.2, with the initial **C** generated as random integers from [1,10]. Whilst all the solutions are mathematically equal in terms of minimising (1.2.2) they are visibly very different which may cause issue when interpreting them as chemical spectra.

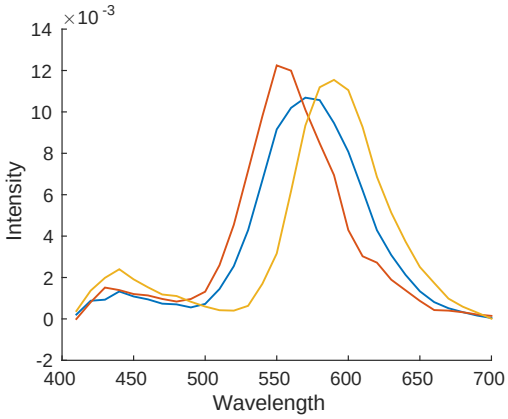
We will therefore derive an algorithm for exploring this solution space so we can obtain a range of results from the entire solution space instead of the random selection of results repeated solutions to (1.2.2) obtain.

4.6 Monte Carlo Random Walk Solution Space Sampling

As shown in §4.1 the boundaries of the solution space have no analytic solution when $r > 2$, a common situation with Raman images. The exploration method we use in §4.1 is also not a viable option due to computational difficulties in testing (1.2.5) for all points, and all combinations of points, on an r-dimensional mesh. Therefore we will employ Monte Carlo simulation techniques of random walks and rejection sampling to sample over the solution space.

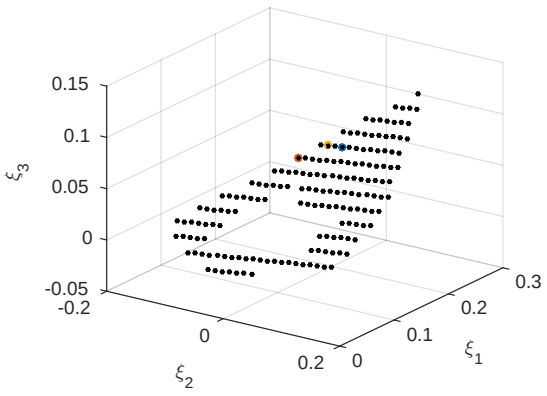


(a) 3-dimensional solution plane with three linear combination coefficients used to generate the three spectra in figure 4.8b.

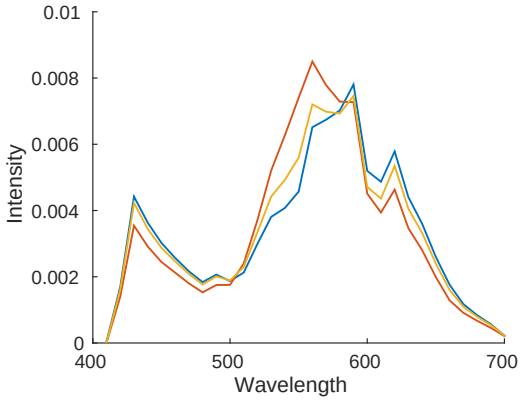


(b) Spectra corresponding to the coefficients in figure 4.8a.

Figure 4.8: Potential solutions to (1.2.2) chosen from the solution plane using data from Lawton and Sylvestre [62]. Spectra calculated using (4.4.3).



(a) 3-dimensional solution plane with three linear combination coefficients used to generate the three spectra in figure 4.9b.



(b) Spectra corresponding to the coefficients in figure 4.9a.

Figure 4.9: Potential solutions to (1.2.2) chosen from the solution plane using data from Lawton and Sylvestre [62]. Spectra calculated using (4.4.3).

4.6.1 Random Walk Algorithm

We propose a simple random walk algorithm for exploring the $(\xi_1, ..., \xi_r)$ $(r \geq 2)$ unit hypersphere, used in 3-dimensions in §4.4.1, defined by

$$\sum_{k=1}^{r} \xi_k^2 = 1 \tag{4.6.1}$$

where the coordinates on the sphere surface, $\boldsymbol{\xi}_k = (\xi_{k,1}, ..., \xi_{k,r})^T$, become the columns of the $r \times r$ transformation matrix

$$\Xi = [\boldsymbol{\xi}_1 \ \boldsymbol{\xi}_2 \ \dots \ \boldsymbol{\xi}_r]$$

initially defined in §2.4 and used in §4.4.2 and §4.4.3. We construct our random walk around this hypersphere as it will ensure the unit spectra constraint of (1.2.5) is met. To generate our step around the unit hypersphere (4.6.1) we take inspiration from the random walk defined by Hastings [47] and Kac [54] due to the simplicity with which it can be implemented. Future work in this area may involve a more efficient random walk in higher dimensions however as we rarely exceed three or four dimensions this causes us no problems for now. For a point on unit hypersphere (4.6.1) at iteration t, $\boldsymbol{\xi}^{(t)}$, we define our random walk by letting

$$\boldsymbol{\xi}^{(t+1)} = \mathbf{G}_{l,m}(\theta)\boldsymbol{\xi}^{(t)} \tag{4.6.2}$$

for rotation matrix

$$\overset{r\times r}{\mathbf{G}_{l,m}}(\theta) = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & \cos(\theta) & \dots & -\sin(\theta) & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & \sin(\theta) & \dots & \cos(\theta) & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \tag{4.6.3}$$

with angle θ randomly generated from $\theta \sim \mathcal{U}(0, 2\pi)$. The rotation matrix, $\mathbf{G}_{l,m}(\theta)$, is only different from the identity matrix at entries (l, l), (l, m), (m, l) and (m, m) where l and m are chosen such that $1 \leq l < m \leq r$. This random walk simply rotates $\boldsymbol{\xi}$ around the unit sphere to obtain a new set of coordinates, $\boldsymbol{\xi}'$ and therefore a new spectrum $\mathbf{s}' = \mathbf{K}\boldsymbol{\xi}'$.

To obtain our starting coordinate for the random walk, $\boldsymbol{\xi}^{(1)}$ on the unit hypersphere, we use any method for solving (1.2.2) to obtain estimates of the $\mathbf{s}_k$. With our estimates of the $\mathbf{s}_k$ we find some $\boldsymbol{\xi}$ such that $\mathbf{s}_k = \mathbf{K}\boldsymbol{\xi}$, where $\mathbf{K} = [\mathbf{k}_1 \ ... \ \mathbf{k}_r]$ are the r eigenvectors of $\frac{\mathrm{Y}^T\mathrm{Y}}{n}$ corresponding to the r largest eigenvalues.

We now propose two options for sampling the solution space of acceptable choices for $\boldsymbol{\xi}$. The first samples each coordinate point, $\boldsymbol{\xi}$, individually and calculates $\mathbf{s}^{(t)} = \mathbf{K}\boldsymbol{\xi}^{(t)}$ to check for negative elements in the spectrum. Sampling continues until the iteration limit is reached. A brute force method is then used testing all possible combinations of r coordinates for negative concentration values by calculating $\mathbf{C}$ such that

$$\begin{aligned} \mathbf{Y} &= \mathbf{K}\Xi\mathbf{C} \\ \Rightarrow\ \mathbf{K}^T\mathbf{Y} &= \Xi\mathbf{C} \\ \Rightarrow\ \mathbf{C} &= (\Xi)^{-1}\,\mathbf{K}^T\mathbf{Y}. \end{aligned} \tag{4.6.4}$$

In testing we greatly increased the speed of the brute force part by using parallel computing as each combination of r $\boldsymbol{\xi}$ to form Ξ may be tested independently of the others.

Algorithm 4.6.1. *r* **Spectra Solution Space Random Walk 1**

1. Solve (1.2.2) to obtain $\mathbf{S} = [\mathbf{s}_1\ ...\ \mathbf{s}_r]$ and find starting coordinate $\boldsymbol{\xi}^{(1)}$ such that $\mathbf{K}\boldsymbol{\xi}^{(1)} = \mathbf{s}_1$.

2. Sample θ from $\theta \sim \mathcal{U}(0, 2\pi)$, l and m from $\mathcal{U}(1, r)$ so all possible rotations are equally likely. Evaluate $\boldsymbol{\xi}^{(t+1)}$ from (4.6.2).

3. Check $\mathbf{s} = \mathbf{K}\boldsymbol{\xi}^{(t+1)}$ satisfies $s_j > 0$ for $j = (1, ..., p)$.

4. If accepted increase t by 1 and progress to point $\boldsymbol{\xi}^{(t+1)}$, else remain at $\boldsymbol{\xi}^{(t)}$, and return to step 2.

5. Repeat steps 2–4 until the predefined limit of t_{MAX} is reached.

6. Check all $\frac{t_{\text{MAX}}!}{r!(t_{\text{MAX}}-r)!}$ combinations of r sampled coordinates, used to form

$$\Xi = [\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_r]$$

for $\mathbf{C}_{i,k} > 0$ for $i = (1, ..., n)$ and $k = (1, ..., r)$. With $\mathbf{C}$ calculated using (4.6.4). We accept or reject each $\boldsymbol{\xi}^{(t)}$ as a point in the solution space using one of the following criteria,

   (a) Accept $\boldsymbol{\xi}^{(t)}$ if at least one of the $\frac{(t_{\text{MAX}}-1)!}{(r-1)!(t_{\text{MAX}}-r-2)!}$ versions of Ξ in which it appears generates non-negative concentrations

   (b) Accept $\boldsymbol{\xi}^{(t)}$ if all of the $\frac{(t_{\text{MAX}}-1)!}{(r-1)!(t_{\text{MAX}}-r-2)!}$ versions of Ξ in which it appears generates non-negative concentrations

In selecting θ, l and m for (4.6.3) from uniform distributions this random walk will be equally likely to select any point on the hypersphere. Therefore, for a large enough t_{MAX}, it will provide an even sampling of points from the volume representing constraints (1.2.5).

This algorithm initially generates t_{MAX} possible points on the solution space which are then reduced to T points after the rejection of those giving negative values in $\mathbf{C}$. In practice we use the less severe rejection criteria in step 6a as the more severe criteria resulted in very few accepted $\boldsymbol{\xi}^{(t)}$. With more powerful computation the more strict criteria could be used as a very large t_{MAX} would be possible resulting in more accepted points.

The second algorithm avoids step 6 of algorithm 4.6.1 and produces T points on the solution space but takes longer to complete. In avoiding step 6 of algorithm 4.6.1 this walk requires a full transformation matrix Ξ as a starting point. We acquire this using any solution to (1.2.2) and finding Ξ such that $\mathbf{S} = \mathbf{K}\Xi$.

Algorithm 4.6.2. *r* **Spectra Solution Space Random Walk 2**

1. Solve (1.2.2) to obtain $\mathbf{S}$ and find starting coordinates $\Xi^{(1)}$ such that $\mathbf{K}\Xi^{(1)} = \mathbf{S}$.
2. For each column of $\Xi^{(t)}$, $\boldsymbol{\xi}_k^{(t)}$ for $k = 1, ..., r$, sample θ from $\theta \sim \mathcal{U}(0, 2\pi)$, l and m from $\mathcal{U}(1, r)$ and evaluate $\boldsymbol{\xi}_k^{(t+1)}$ using (4.6.2).
3. Check $\mathbf{S} = \mathbf{K}\Xi^{(t+1)}$ and $\mathbf{C}$, calculated as in (4.6.4), satisfy $\mathbf{S}_{j,k} > 0$ and $\mathbf{C}_{i,k} > 0$ for $i = (1, ..., n)$, $j = (1, ..., p)$ and $k = (1, ..., r)$.
4. If accepted increase t by 1 and progress to $\Xi^{(t+1)}$, else remain at $\Xi^{(t)}$, and return to step 2.
5. Repeat steps 2–4 until the predefined limit of T is reached.

The acceptance criterion of this algorithm is the same as using 6b in algorithm 4.6.1 however we now also check the estimated concentrations for negative elements at the same time. As this algorithm is more strict in accepting a point as lying in the solution space we selected algorithm 4.6.1 for use in our testing due to computational speed. Algorithm 4.6.1 with acceptance criteria 6a also allows us to rank the accepted points. Rankings are calculated based on how many times a given point, $\boldsymbol{\xi}$, appears in an accepted matrix Ξ in step 6a. These rankings may be useful to weight the selection of points from the solution space so only those frequently giving non-negative concentrations are used.

Applying algorithm 4.6.1 to Lawton's data, detailed in §1.2.2, we obtained the plane estimate shown in figure 4.10. In this figure we also see an example of the point rankings where the size of the marker for each point corresponds to the number of times a Ξ in

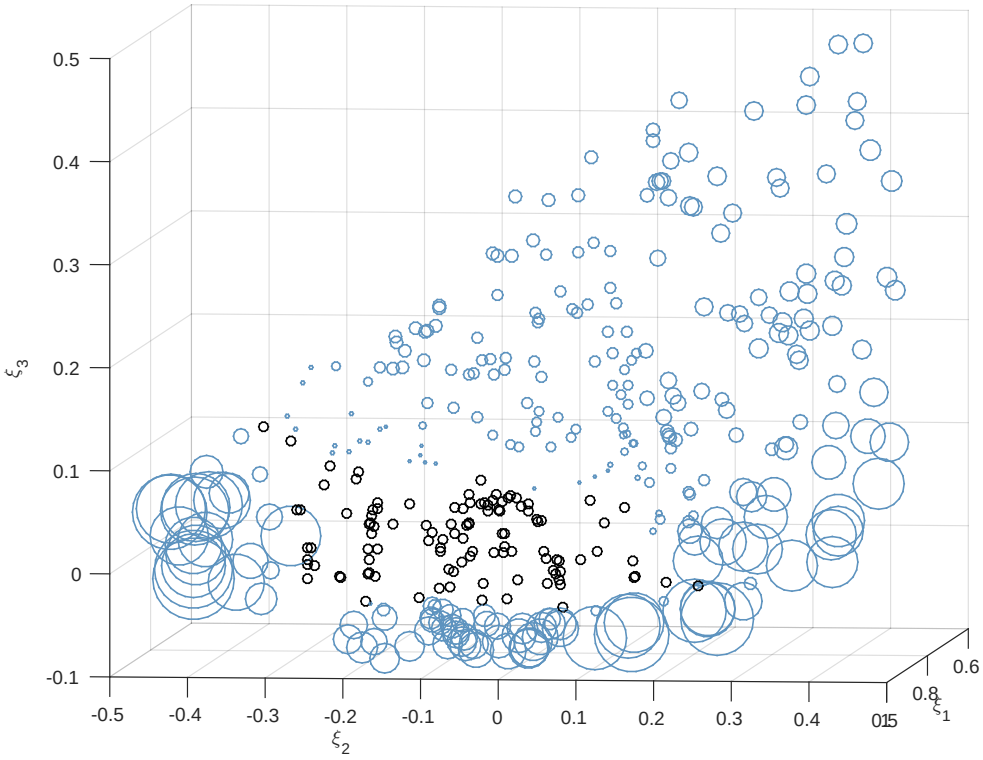
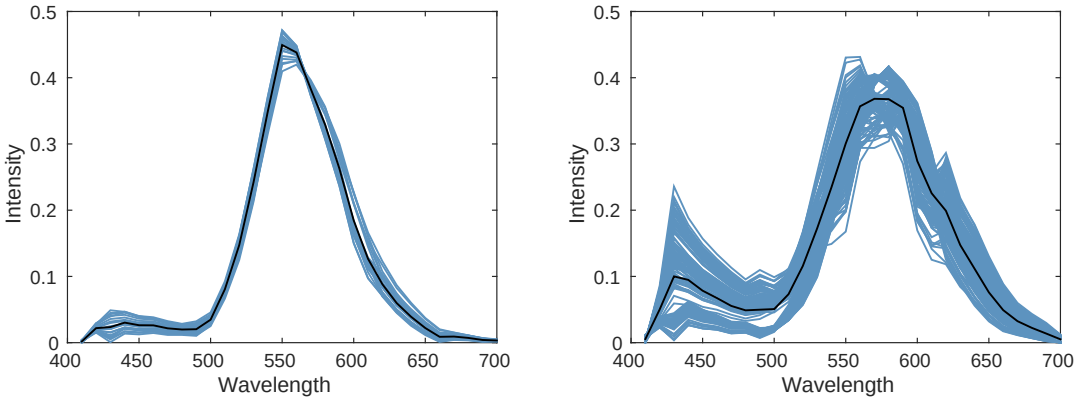


Figure 4.10: 400 $\boldsymbol{\xi}_i$ found using algorithm 4.6.1, with acceptance criteria 6a, on data from Lawton and Sylvestre [62]. Accepted points in blue with point size corresponding to the proportion of accepted combinations out of the total combinations in which they appear in step 6a. The bulk of frequently accepted points around $\xi_3 = 0$ is due to the third eigenvector, to which $\xi_3 = 0$ corresponds, representing background signal. Our random walk algorithm (4.6.1) returns a solution space very similar to that from §4.4.3 without the need for a large mesh of points.

which it featured generated non-negative concentrations. For this dataset the grouping of highly ranked points around $\xi_3 = 0$ is likely because the data contains only two component chemicals. Therefore the third principal component represents background signal and noise and so we have no guarantee the corresponding concentrations will be positive.

4.6.2 Clustering Spectra from the Solution Space

With T points on the solution space, generated by algorithm 4.6.1 or 4.6.2, we now look at creating r groups of points, $\boldsymbol{\xi}$, to represent the r true component spectra in the data. This will help us avoid situations such as that seen in figure 4.9b where all r estimated spectra resemble a single component spectrum as we can select a single spectrum from



(a) Cluster 1. **(b)** Cluster 2.

Figure 4.11: Results of UPGMA clustering the spectra generated by points on the solution space estimated for data from Lawton and Sylvestre [62]. Solution space estimated using algorithm 4.6.1 with acceptance criteria 6a. Cluster means shown in black are very close matches to the true spectra in the data [62].

each group.

With our three spectrum solution space for Lawton's data [62] we applied a variety of clustering techniques to the $\mathbf{s}_t = \mathbf{K}\boldsymbol{\xi}_t$, for $t = 1, ..., T$. These included k-means, §2.10, EMGM, §2.12.1, and the unweighted pair group method with arithmetic mean clustering (UPGMA) algorithm, §2.11. For this data the UPGMA method provided us with groups most closely representing the true component spectra however this will be largely data dependent. The two clusters of spectra, not including background signal, we found in our solution space for Lawton's data are shown in figure 4.11 along with their means.

We can also represent these clusters in our plot of the 3-dimensional volume by colouring the $\boldsymbol{\xi}_t$ corresponding to the $\mathbf{s}_t = \mathbf{K}\boldsymbol{\xi}_t$ in each cluster. For Lawton's data we give an example of this in figure 4.12. From this volume we can see that the locations of the clusters lie in areas where the coefficients of two eigenvectors are low, an unsurprising result when we consider that each eigenvector represents a component spectra as detailed in §2.3.

4.6.3 Applying to more complicated data

We now test random walk algorithm 4.6.1 with more complicated data. The dataset we select for this is one of the bicalutamide datasets detailed in §1.2.2. We select this dataset as the group of Raman images to which it belongs will feature heavily in Chapter 5 when we perform hypothesis testing incorporating random walk algorithm 4.6.1. After baseline correction and anomaly trimming, see chapter 3, approximately 5% of all $y_{i,j} \in \mathbf{Y}$ are

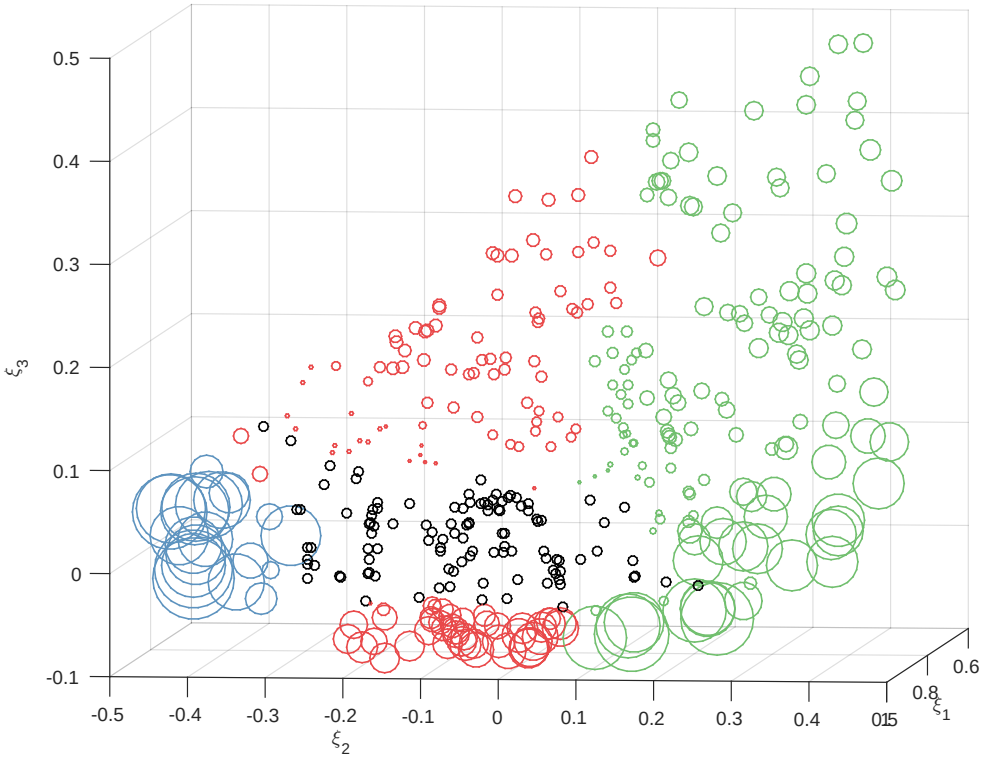


Figure 4.12: 400 $\boldsymbol{\xi}_i$ found using algorithm 4.6.1, with acceptance criteria 6a, on data from Lawton and Sylvestre [62]. Accepted points are coloured based on their group, detected with UPGMA clustering, with point size corresponding to the proportion of accepted combinations out of the total combinations in which they appear in step 6a. The blue and green groups, with the bulk of the highly ranked points, are around $\xi_3 = 0$ as expected with ξ_3 representing the concentration of background signal. We now have an even clearer picture of the solution space of (1.2.2) and why the estimated spectra can vary so much.

negative resulting in negative values in **S**, **C** or both. We must therefore account for these negative elements to ensure the solution space exists. To achieve this we relax constraints (1.2.5) by allowing a small number of negative values in steps 3 and 6 of algorithm 4.6.1. This gives us

Algorithm 4.6.3. Modified Steps for r Spectra Solution Space Random Walk 1

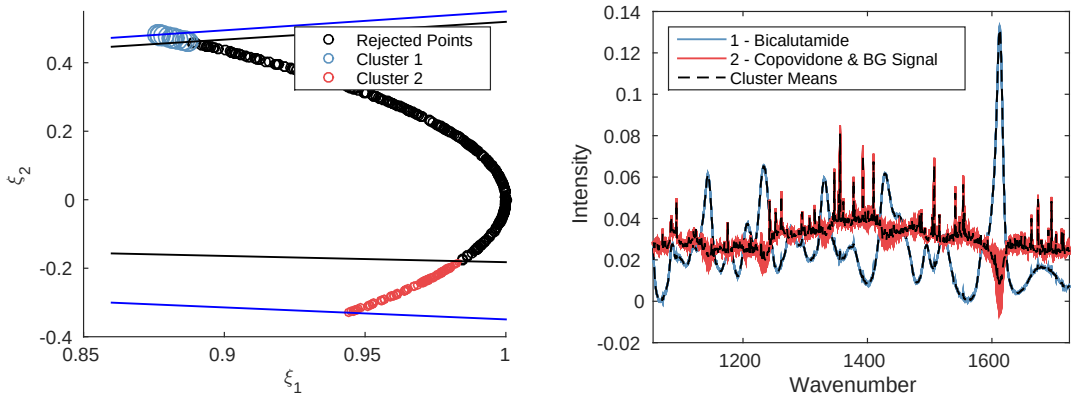
3. Check $\mathbf{s} = \mathbf{K}\boldsymbol{\xi}^{(t+1)}$ satisfies
$$\sum_{j=1}^{p} \mathbb{I}_{s_j>0} \geq p(1-\alpha_1)$$
for indicator function $\mathbb{I}$ and proportion of negative elements allowed per spectrum α_1.

6. Check all $\frac{t_{\text{MAX}}!}{r!(t_{\text{MAX}}-r)!}$ combinations of r sampled coordinates, used to form
$$\Xi = [\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_r]$$
for
$$\sum_{i=1}^{n}\sum_{k=1}^{r} \mathbb{I}_{c_{i,k}>0} \geq nr(1-\alpha_2)$$
with **C** calculated using (4.6.4). We accept or reject each $\boldsymbol{\xi}^{(t)}$ as a point in the solution space using one of the following criteria,
   (a) Accept $\boldsymbol{\xi}^{(t)}$ if at least one of the $\frac{(t_{\text{MAX}}-1)!}{(r-1)!(t_{\text{MAX}}-r-2)!}$ versions of Ξ in which it appears generates non-negative concentrations
   (b) Accept $\boldsymbol{\xi}^{(t)}$ if all of the $\frac{(t_{\text{MAX}}-1)!}{(r-1)!(t_{\text{MAX}}-r-2)!}$ versions of Ξ in which it appears generates non-negative concentrations

Applying this modified random walk to the bicalutamide data we found $\alpha_1 = 0.01$ and $\alpha_2 = 0.1$ to work well in giving us a good estimation of the solution space without requiring too great a t_{MAX}. Figure 4.13a shows the two spectrum solution plane for the bicalutamide data, with the two UPGMA derived clusters represented in red and blue. Three spectrum random walks for this data, performed with algorithm 4.6.3 and used to estimate bicalutamide, copovidone and background signal, will be used extensively in Chapter 5. We use the two spectra case here as a proof of concept only as it causes the copovidone and background signal spectra to be combined, as seen in figure 4.13b.

4.7 Discussion

With (1.2.2) being non-identifiable in the solutions to **S** and **C** we have a solution space from which we can select columns, $\boldsymbol{\xi}$, of Ξ such that $\mathbf{S} = \mathbf{K}\Xi$ as defined in §4.4.2. Whilst



(a) Solution space with points coloured corresponding to the cluster in which the generated spectra are assigned, §4.6.2.

(b) UPGMA, §2.11, clustered spectra generated by points on the solution space.

Figure 4.13: Random walk conducted using algorithm 4.6.1 on bicalutamide data, §1.2.2. For this data the variation in estimated spectra across the solution space appears minimal so using a single solution to (1.2.2) may provide consistent results.

not directly used in most modern methods this solution space was identified by Lawton and Sylvestre [62] in the first published chemometric method for solving (1.2.2). We therefore began our investigation with the solution space boundaries defined by Lawton. Before we used Lawton's constraints however we first proved that Lawton's solution to (1.2.2) was indeed a least squares solution. We established this by proving that under appropriate constraints for eigenvectors, (2.1.2), the eigenvectors provide a least squares solution to (2.1.1). With a solution to (2.1.1) we are able to transform $\mathbf{G}$ and $\mathbf{H}$ such that they comply with constraints (1.2.5) and therefore provide a solution to (1.2.2).

With the foundation of Lawton's solution space proven we went on to show that they will always exist for Raman spectroscopy data. With these shown we looked at calculating the boundaries for higher dimensional models with $r > 2$. However we were unable to obtain analytically calculable solution space boundaries as in the $r = 2$ case. We therefore turned our attention to Monte Carlo simulation techniques for estimating the shape and boundaries of the solution space to (1.2.2).

The Monte Carlo technique we used was a random walk which we implemented using a rotation matrix to move the linear combination coefficients $\boldsymbol{\xi}$, used to calculate $\mathbf{s} = \mathbf{K}\boldsymbol{\xi}$, about the solution space on a hypersphere whilst testing the constraints (1.2.5). We then modified this algorithm to allow a small quantity of negative elements in the estimated spectra and concentrations so noisy and pre-processed data can still be explored. Whilst effective at exploring the solution space this method suffered with lengthy computation

times. These were caused by the rotation of the $\boldsymbol{\xi}$ and the selection of any point on the hypersphere of which only a small region was our solution space. Beneficial future work on our random walk would focus on the area of calculating the rotation matrix, (4.6.3). Restrictions in the range of points to which it could move would provide an increase in speed especially if more constraints could be included in the step selection stage.

With our algorithm locating the solution space of both sample datasets we looked at clustering the range of possible solutions to (1.2.2), more specifically the spectra generated by the points, $\boldsymbol{\xi}$, on the solution space. This enabled us to separate the possible spectra into groups representing the component spectra in the dataset. Our motivation for this is that when selecting r points from the solution space we can ensure we obtain a single spectra for each chemical component, avoiding situations where nearly equal spectra are returned as in figure 4.9.

Using our random walk we will derive robust hypothesis tests in chapter 5 that take into account the non-identifiability of (1.2.2) and the arbitrary solution identified by methods for solving (1.2.2). In comparison to current publications on the non-identifiability of (1.2.2) [35, 95] our random walk can be applied to any Raman spectroscopic image with no requirements for specific features in either the scan subject or the component spectra.

Chapter 5

Multivariate Hypothesis Testing of Concentration Magnitudes

5.1 Objectives of Testing Concentration Magnitudes

Our initial analysis of a Raman image usually involves finding a solution to

$$\mathbf{Y} = \mathbf{C}\mathbf{S}^T + \mathbf{E} \tag{1.2.2}$$

my minimising

$$\underset{\mathrm{S,C}}{\operatorname{argmin}} \left\| \mathbf{Y} - \mathbf{C}\mathbf{S}^T \right\|$$

subject to

$$\begin{array}{lll} 1. & \mathbf{s}_k^T \mathbf{s}_k = 1 & \text{for all } k = 1, ..., r \\ 2. & s_{j,k} \geq 0 & \text{for all } j = 1, ..., p \text{ and } k = 1, ..., r \\ 3. & c_{i,k} \geq 0 & \text{for all } i = 1, ..., n \text{ and } k = 1, ..., r \end{array} \tag{1.2.5}$$

as we have seen in previous chapters. This model however is non-identifiable, a subject covered in §1.2.1 and extensively investigated in Chapter 4. It is this non-identifiability which makes further analysis of the results, for example estimates the concentration matrix $\mathbf{C}$, difficult as the matrix on which we will be performing hypothesis tests is one of an infinite number of possible solutions. These hypothesis tests focus on the means of two sets of concentrations and the spatial information contained within the concentrations.

When performing hypothesis tests on these concentrations our first area of interest is the magnitudes of the concentrations and whether these can be used to compare the Raman scans of two subjects. The primary question to be answered by this test is do the Raman images contain equal quantities of their component chemicals? For example in the

pharmaceutical industry this may lead us to ask, do all the tablets from a single production run contain equal quantities of the active ingredients? However as every Raman image is subject to noise we will never find exact quantities of component chemicals and so we will test whether the concentrations may have come from distributions with equal means.

The second area we investigate is the spatial information contained in the estimated concentrations. The question we ask here is how well mixed are the component chemicals in the subject? Another example from the pharmaceutical industry is the question, do amorphous production methods produce a more thoroughly mixed product than standard crystalline methods? (Amorphous tablets are mixed at a molecular level by melting the component chemicals, in theory producing a more thoroughly mixed product.)

Common methods for testing the mixing of a subject are applied during the production process. The methods of Koc et al. [58] and Coënt et al. [21] take photographs of the components during the mixing process, convert them to greyscale and count the number of pixels representing each gray level either in the entire image or in small sections of the image. These methods require the components to be different colours and so usually employ dyed powders as surrogate chemical components. They are also predominantly used to test the mixing capabilities of equipment on a production line rather than the final product such as a pharmaceutical tablet. A method close to those we propose in this chapter is by Lee and Lin [65]. This method uses Fourier-transform infra-red spectroscopy (FT-IR) to repeatedly scan the product during mixing in the same way our Raman images are gathered. The spectra are then estimated at each scan location and the uniformity of the mixing is calculated by comparing the peak intensities of the estimated spectra at each location.

A final question we must ask throughout all of our testing is what effect will the non-uniqueness of solutions to (1.2.2) have on our results? In answering this we turn to the solution space random walk, algorithm 4.6.3. This will allow us to obtain a number of solutions to our bilinear model, (1.2.2), and test each one, giving more confidence that our test conclusion is not the result of an anomalous model solution. This random walk component to our hypothesis testing introduces a second layer to the hypothesis test as each solution chosen from the solution space has a p-value associated with it. We therefore propose a method for summarising the results of these multiple p-values.

Note: In subsequent sections of this chapter the $\mathbf{C}$ we perform hypothesis tests on are estimates, $\tilde{\mathbf{C}}$, of the parameter $\mathbf{C}$ in (1.2.2). The tilde is dropped to simplify notation.

5.1.1 Concatenated Data Model

Before comparing datasets we modify them to ensure we are comparing concentrations of the same estimated component chemicals. For Raman images $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$, with n_1 and n_2 pixels respectively, in the usual form

$$\overset{n_l \times p}{\mathbf{Y}^{(l)}} = \begin{bmatrix} \mathbf{y}_1^{(l)T} \\ \vdots \\ \mathbf{y}_{n_l}^{(l)T} \end{bmatrix} \tag{5.1.1}$$

we combine these into a single larger image and define the model

$$\begin{bmatrix} \mathbf{Y}^{(1)} \\ \mathbf{Y}^{(2)} \end{bmatrix}^{(n_1+n_2)\times p} = \begin{bmatrix} \mathbf{C}^{(1)} \\ \mathbf{C}^{(2)} \end{bmatrix}^{(n_1+n_2)\times r} \mathbf{S}^{T\, r\times p} + \mathbf{E}. \tag{5.1.2}$$

Here we are assuming that both Raman images share estimated component spectra, $\mathbf{S} = [\mathbf{s}_1 \ \ldots \ \mathbf{s}_r]$, but have individual concentration matrices of the standard form

$$\mathbf{C}^{(l)} = \begin{bmatrix} \mathbf{c}_1^{(l)T} \\ \vdots \\ \mathbf{c}_{n_l}^{(l)T} \end{bmatrix}. \tag{5.1.3}$$

We fit this model by solving

$$\underset{\mathbf{S},\mathbf{C}^{(1)},\mathbf{C}^{(2)}}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{Y}^{(1)} \\ \mathbf{Y}^{(2)} \end{bmatrix} - \begin{bmatrix} \mathbf{C}^{(1)} \\ \mathbf{C}^{(2)} \end{bmatrix} \mathbf{S}^T \right\| \tag{5.1.4}$$

as with (1.2.2) where $\|\cdot\|$ is an extension of the Euclidean norm to matrices, (1.2.4) in §1.2.

5.1.2 Defining Submatrices of C

After estimating concentration matrices we may need to modify them further before testing. For example a Raman image is often not purely the scan subject and usually contains some background information. Testing the concentration magnitudes and including these regions where there was none of the subject is clearly going to affect the results. We therefore divide our estimated concentrations into submatrices representing the concentration magnitudes at small areas of the Raman image so we can test only parts which are sensible.

To define these submatrices, for some concentration matrix $\mathbf{C}^{(l)}$, we first define the columns of the concentration matrix representing the concentrations of chemical k at each pixel,

$$\mathbf{v}_k^{(l)} = \begin{bmatrix} c_{1,k}^{(l)} \\ \vdots \\ c_{n,k}^{(l)} \end{bmatrix}. \tag{5.1.5}$$

For the full heatmap we reshape this vector into a matrix with the dimensions of the original Raman image, say $x \times y$, giving

$$\overset{x\times y}{\mathbf{V}} = \begin{bmatrix} v_{k,1}^{(l)} & \cdots & v_{k,(y-1)x+1}^{(l)} \\ \vdots & \ddots & \vdots \\ v_{k,x}^{(l)} & \cdots & v_{k,n}^{(l)} \end{bmatrix}. \tag{5.1.6}$$

Therefore to define a submatrix we select specific elements of (5.1.5), $c_{j,k}^{(l)}$, such that when reshaped to form a heatmap, such as (5.1.6), we have only a subset of the full heatmap. We define this submatrix as

$$\mathbf{v}_{k,m}^{(l)} \quad \text{such that } j \in \mathcal{I}_m \tag{5.1.7}$$

where $\mathcal{I}_m$ is the set of indices, j, of pixels in submatrix m. For example if we were to reshape vector $\mathbf{x} = [1, 2, ..., 16]$ into matrix

$$\begin{bmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{bmatrix}$$

and divide this into four submatrices we would obtain sets

$$\begin{aligned} \mathcal{I}_1 &= \{1, 2, 5, 6\} \\ \mathcal{I}_2 &= \{9, 10, 13, 14\} \\ \mathcal{I}_3 &= \{3, 4, 7, 8\} \\ \mathcal{I}_4 &= \{11, 12, 15, 16\}. \end{aligned}$$

5.2 Established Statistical Tests

For some of the hypothesis testing we perform in this chapter we will use two established statistical testing methods. These are Hotelling's two-sample T^2 test [48, 73, Chapter 5] and bootstrap testing [24], which we will now outline.

5.2.1 Hotelling's Two-Sample T-squared Test

The two-sample T^2 test was proposed by Hotelling [48] to test whether two samples come from population distributions with a common mean. For two p-dimensional datasets,

$\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{n_1}\}$ and $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_{n_2}\}$, we assume

$$\mathbf{x}_i \sim \mathcal{N}_p(\boldsymbol{\mu}_x, \Sigma)$$
$$\mathbf{y}_i \sim \mathcal{N}_p(\boldsymbol{\mu}_y, \Sigma)$$

and consider the testing problem

$$H_0 : \boldsymbol{\mu}_x = \boldsymbol{\mu}_y \quad \text{vs.} \quad H_1 : \boldsymbol{\mu}_x \neq \boldsymbol{\mu}_y$$

The sample means are defined as

$$\overline{\mathbf{x}} = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbf{x}_i \qquad \text{and} \qquad \overline{\mathbf{y}} = \frac{1}{n_2}\sum_{i=1}^{n_2}\mathbf{y}_i \tag{5.2.1}$$

with the pooled sample covariance matrix

$$\mathbf{W} = \frac{\sum_{i=1}^{n_1}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T + \sum_{i=1}^{n_2}(\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})^T}{n_1 + n_2 - 2}. \tag{5.2.2}$$

The statistic

$$t^2 = \frac{n_1 n_2}{n_1 + n_2}(\overline{\mathbf{x}} - \overline{\mathbf{y}})^T \mathbf{W}^{-1}(\overline{\mathbf{x}} - \overline{\mathbf{y}}) \tag{5.2.3}$$

has a $T^2(p, n_1 + n_2 - 2)$ distribution under the null hypothesis [48] or equivalently [73, Section 3.6.1]

$$F = \frac{n_1 + n_2 - p - 1}{(n_1 + n_2 - 2)p} t^2 \sim F(p, n_1 + n_2 - 1 - p). \tag{5.2.4}$$

F will tend to be larger under H_1 as $\|\overline{\mathbf{x}} - \overline{\mathbf{y}}\|$ will be larger. Thus H_0 is rejected in favour of H_1 when the observed F statistic is sufficiently large. In particular H_0 is rejected at the $100(1-\alpha)\%$ level if $F > F_{1-\alpha,p,n_1+n_2-1-p}$, where $F_{1-\alpha,p,n_1+n_2-1-p}$ is the inverse cumulative distribution function of the F_{p,n_1+n_2-1-p} distribution.

Affine Invariance of the T-squared Test Statistic

The T^2 test statistic also has the property of affine invariance. This property means the value is unchanged for data modified by an invertible matrix $\mathbf{R}$ [84]. We demonstrate affine invariance for data matrices, $\mathbf{X}$ and $\mathbf{Y}$, of the form (5.1.1). By transforming our data

$$\mathbf{R}\mathbf{X}^T \quad \text{and} \quad \mathbf{R}\mathbf{Y}^T$$

we now write the pooled sample covariance matrix, (5.2.2), as

$$\mathbf{W} = \frac{\mathbf{R}\mathbf{X}_c^T\mathbf{X}_c\mathbf{R}^T + \mathbf{R}\mathbf{Y}_c^T\mathbf{Y}_c\mathbf{R}^T}{n_1 + n_2 - 2} = \frac{\mathbf{R}\left(\mathbf{X}_c^T\mathbf{X}_c + \mathbf{Y}_c^T\mathbf{Y}_c\right)\mathbf{R}^T}{n_1 + n_2 - 2} = \mathbf{R}\mathbf{W}\mathbf{R}^T,$$

where $\mathbf{X}_c$ and $\mathbf{Y}_c$ are the mean centred data matrices with each $\mathbf{y}_i$ and $\mathbf{x}_i$ replaced by $\mathbf{y}_i - \bar{\mathbf{y}}$ and $\mathbf{x}_i - \bar{\mathbf{x}}$ respectively using (5.2.1).

We therefore rewrite test statistic (5.2.3) as

$$\begin{aligned} t^2 &= \frac{n_1 n_2}{n_1 + n_2}(\mathbf{R}\bar{\mathbf{x}} - \mathbf{R}\bar{\mathbf{y}})^T \left(\mathbf{R}^{-1}\right)^T \mathbf{W}^{-1}\mathbf{R}^{-1}(\mathbf{R}\bar{\mathbf{x}} - \mathbf{R}\bar{\mathbf{y}}) \\ &= \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{x}} - \bar{\mathbf{y}})^T \mathbf{W}^{-1}(\bar{\mathbf{x}} - \bar{\mathbf{y}}) \end{aligned}$$

which is unchanged.

This affine invariance can also nullify the problem of non-unique solutions in (5.1.4) as when we transform the spectra and concentration estimates with some invertible matrix $\mathbf{R}$, such that constraints (1.2.5) hold, our model becomes

$$\begin{aligned} \begin{bmatrix} \mathbf{Y}^{(1)} \\ \mathbf{Y}^{(2)} \end{bmatrix} &= \begin{bmatrix} \mathbf{C}^{(1)} \\ \mathbf{C}^{(2)} \end{bmatrix} \mathbf{R}\mathbf{R}^{-1}\mathbf{S}^T + \mathbf{E} \\ &= \begin{bmatrix} \mathbf{C}^{(1)}\mathbf{R} \\ \mathbf{C}^{(2)}\mathbf{R} \end{bmatrix} \mathbf{R}^{-1}\mathbf{S}^T + \mathbf{E} \end{aligned}$$

where we see that both concentration matrices are transformed by the same matrix and so cancel when calculating the test statistic.

There is however a limitation that causes this property to be inapplicable with our data. For affine invariance to hold we must use all r columns of a concentration matrix $\mathbf{C}^{(l)}$, where the $l \in \{1, 2\}$ matrices correspond to the concentrations of the two images we are comparing. We show this by looking at the means being compared by the hypothesis test where we are testing $H_0 : \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$. Now if we transform each of our concentration matrices, $\mathbf{C}^{(l)}$, to obtain $\mathbf{C}^{(l)}\mathbf{R}$ we are simply testing whether the linear combination of the means, $\boldsymbol{\mu}_1\mathbf{R} = \boldsymbol{\mu}_2\mathbf{R}$, are equal. However if only the first element of each mean were of interest our hypothesis becomes $\mu_{1,1} = \mu_{2,1}$ which is clearly not equivalent to testing $(\boldsymbol{\mu}_1\mathbf{R})_1 = (\boldsymbol{\mu}_2\mathbf{R})_1$ as the first elements of the transformed means are now linear combinations incorporating removed dimensions.

Although seemingly illogical to test only a submatrix of $\mathbf{C}$ when this removes the affine invariance property, we see reason from the perspective of a chemist. When testing the magnitudes of estimated concentrations we are not interested in testing all linear combinations of the concentrations for all component spectra as that has no physical meaning, the reason we place constraints on $\mathbf{C}$ in (1.2.2). We are only interested in comparing a subset of the component chemical concentrations, for example the active ingredients ignoring the binding agent.

5.2.2 Bootstrap Sampling

Bootstrap sampling is a technique we use throughout this chapter. Bootstrapping was suggested by Efron [29] for making inferences about a population from a sample of that population. Bootstrapping is primarily recommended [2, 24] for use in situations where

1. The theoretical distribution of the test statistic is unknown.
2. The asymptotic distribution of the test statistic is well known but the sample size is too small for the asymptotic distribution to be accurate.
3. Power calculations for a test statistic need to be performed however only a small sample from the population distribution is available. Bootstrap sampling allows us to simulate multiple samples from the population distribution.

In bootstrapping we draw conclusions comparing our sample to random resamples, a process analogous with comparing samples from the true population to the true population itself. However as we are estimating the distribution of the test statistic under the null hypothesis we must take care that the bootstrap resampling is performed under the null hypothesis. This may not be true in some cases if only the data is resampled and if care is not taken it can lead to a test with poor power [44].

Although we can normally use Hotelling's T^2 test statistic with data that may not be multivariate normal, as the central limit theorem gives convergence to normal for a sufficient sample size, this law of large numbers may not be applicable to our estimated concentrations, $\mathbf{C}$. This is because the sample size of our estimated concentrations is related to dimension n in our model, (1.2.2). Therefore the central limit theorem of $n \to \infty$ corresponds not only with the increasing sample size of $\mathbf{C}$ but with an increase in the number of parameters in (1.2.2) which is defined as $n \times p \times r$.

We therefore define a simple algorithm for using the T^2 test statistic with no requirement for an asymptotic distribution, point 2 in the uses for bootstrapping detailed previously. Use of the variant of Hotelling's T^2 test in this algorithm also requires the assumption of equal variances between samples. For our data comprised of either pairs of subsets from a single Raman image or pairs of Raman images of the same subject from the same spectrometer this assumption should hold. However if the images under comparison were from different spectrometers or were of subject with different compositions an alternative test statistic may be required.

Algorithm 5.2.1. Hotelling's two-sample T^2 test for equal means by bootstrap

1. Calculate the initial test statistic, (5.2.4), for datasets $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ of dimension $n_1 \times r$ and $n_2 \times r$ respectively.
2. Calculate the residuals $\mathbf{E}^{(1)} = \mathbf{C}^{(1)} - \mathbf{1}_{n_1}\bar{\mathbf{c}}^{(1)T}$ and $\mathbf{E}^{(2)} = \mathbf{C}^{(2)} - \mathbf{1}_{n_2}\bar{\mathbf{c}}^{(2)T}$.
3. Calculate the pooled mean of the samples
$$\boldsymbol{\mu}_p = \frac{1}{n_1 + n_2}\left[\sum_{i=1}^{n_1}\mathbf{c}_i^{(1)} + \sum_{i=1}^{n_2}\mathbf{c}_i^{(2)}\right]. \tag{5.2.5}$$
4. Resample the residuals, $\mathbf{E}^{(1)*}$ and $\mathbf{E}^{(2)*}$, with replacement using
$$E_{i,k}^{(l)*} = E_{i',k'}^{(l)} \text{ for } i \in \{1, ..., n_l\}, k \in \{1, ..., r\}, i' \sim \mathcal{U}(1, n_l) \text{ and } k' \sim \mathcal{U}(1, r)$$
to give a matrix of the same size however with randomly selected elements.
5. Recalculate the test statistic, (5.2.4), using $\mathbf{C}^{(1)*} = \boldsymbol{\mu}_p + \mathbf{E}^{(1)*}$ and $\mathbf{C}^{(2)*} = \boldsymbol{\mu}_p + \mathbf{E}^{(2)*}$ to give $F^{(d)}$.
6. Repeat steps 4 and 5 for $d = 1, ..., N$.

We can now calculate the p-value of the T^2 test for equal means, $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$, with
$$\frac{1 + \sum_{d=1}^{N}\mathbb{I}(F^{(d)} > F)}{1 + N}. \tag{5.2.6}$$
where I is the indicator function. In using algorithm 5.2.1 we can now perform the T^2 test in spite of the knowledge that the sample distributions, the distributions of estimated concentrations, may not be multivariate normal.

5.3 Hypothesis Tests for Equal Means in Estimated Concentration Magnitudes

Our chosen test statistic for this section is Hotelling's T^2, §5.2.1, as it allows us to compare the mean concentration magnitudes between Raman images. As discussed in §5.1 the motivation for this is to compare the quantities of component chemicals in a number of subjects. That is, have the estimated concentrations come from distributions with equal means?

The data we will use in this section are two Raman images representing the upper and lower face of a single bicalutamide tablet. We therefore use the data pooling method described in §5.1.1 to combine these images and obtain estimates of concentrations for three component spectra; copovidone and bicalutamide which are in the tablet, and background signal from

the scan bed around the tablet. We also perform the concentration heatmap splitting as described in §5.1.2 to give the heatmaps in figure 5.1 with concentration subsections $\mathbf{v}_{k,m}^{(l)}$, where $m = 1, ..., 36$ are the image sections, $k = 1, ..., r$ are the chemical components and $l = 1, 2$ are the two tablets used in the concatenated data.

5.3.1 Testing a Single Estimate of Concentration Magnitudes

Our first area of investigation is hypothesis tests on concentrations estimated from a single solution to (5.1.4) which is equivalent to selecting a single solution from the solution space defined in Chapter 4. We first define the concentrations we will test using submatrices, (5.1.7),

$$\overset{n_t \times r'}{\mathbf{B}_t} = \begin{bmatrix} \mathbf{v}_{k_\alpha, m_\alpha}^{(l)} & \cdots & \mathbf{v}_{k_\beta, m_\alpha}^{(l)} \\ \vdots & \ddots & \vdots \\ \mathbf{v}_{k_\alpha, m_\beta}^{(l)} & \cdots & \mathbf{v}_{k_\beta, m_\beta}^{(l)} \end{bmatrix} \tag{5.3.1}$$

for component spectra $1 \leq k_\alpha \leq k \leq k_\beta \leq r$, giving a total of r' spectra, and concentration submatrices $m_\alpha \leq m \leq m_\beta$ defined in §5.1.2 with a total of n_t pixels. To perform a T^2 test on these concentration subsets we define the models

$$\mathbf{B}_1 = \mathbf{1}_{n_1} \boldsymbol{\mu}_1^T + \mathbf{E}_1 \tag{5.3.2}$$

$$\mathbf{B}_2 = \mathbf{1}_{n_2} \boldsymbol{\mu}_2^T + \mathbf{E}_2. \tag{5.3.3}$$

With these defined we now test the hypotheses

$$H_0 : \ \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 \qquad H_1 : \ \boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$$

using the method detailed in §5.2.1. Using the standard T^2 test however requires our data, $\mathbf{B}_t$ to either follow multivariate normal distributions $\mathcal{N}_r(\boldsymbol{\mu}_t, \Sigma)$ or be sufficient in sample size to asymptotically follow this distribution by the central limit theorem. As we cannot guarantee this to be the case when testing small submatrices of the estimated concentrations, using the bootstrap T^2 test, algorithm 5.2.1, may be more appropriate.

To implement these tests on our bicalutamide data, figure 5.1, we used 200 bootstrap resamples ($N = 200$) and selected four regions of the Raman image concentrations which are comprised of either a single one of the 36 submatrices, $\mathbf{v}_{k,m}^{(l)}$ from (5.1.7), or a collection of them joined to form a larger vector,

1. Inner 16 submatrices, upper vs. lower face. For $\mathbf{B}_1$
    - $l = 1$, upper face concentrations

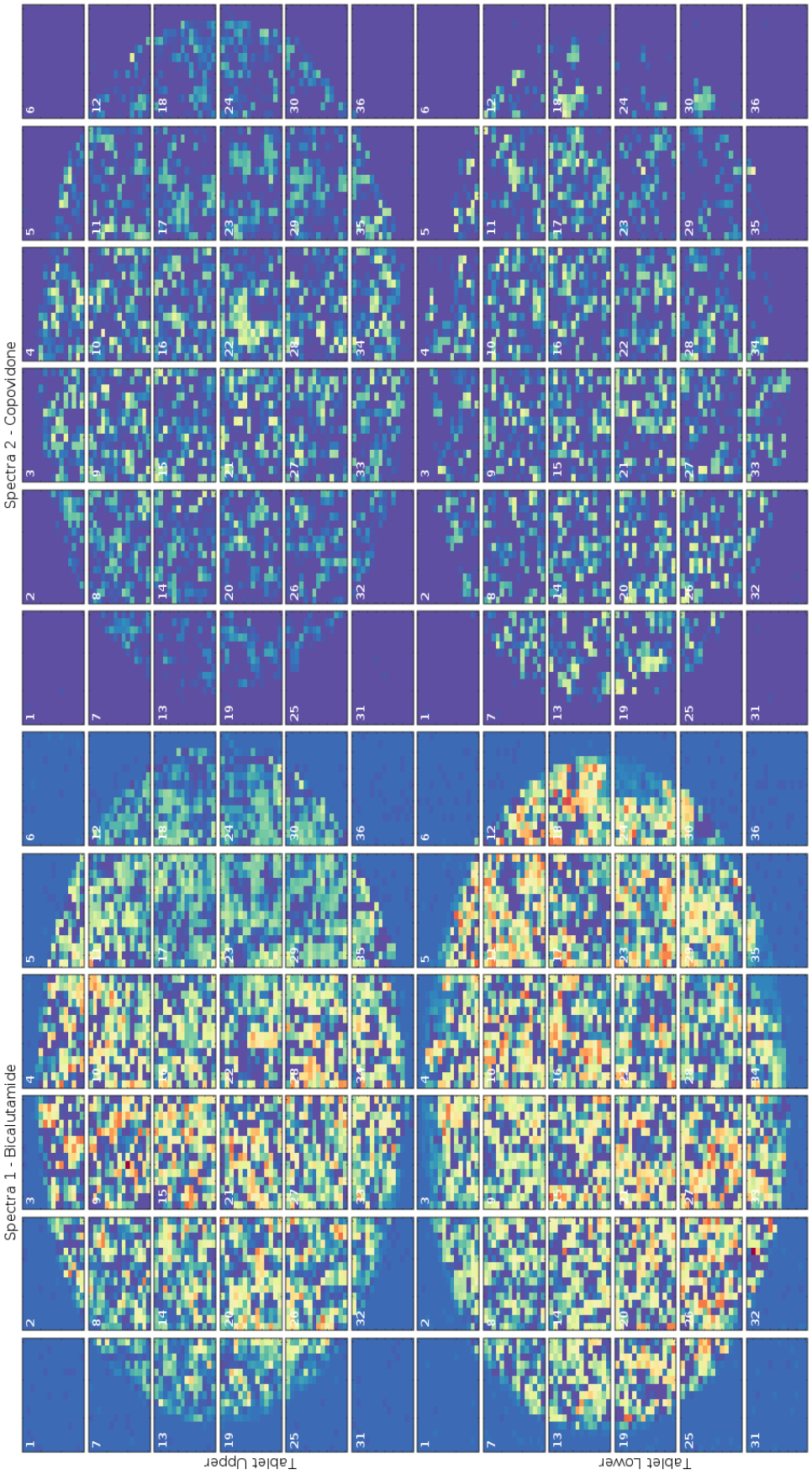


Figure 5.1: Concentrations, **C**, obtained using the ALS algorithm, (2.7.1), applied to pooled bicalutamide data. Concentration matrices are separated as described in §5.1.2.

- $k \in [1, 2]$, bicalutamide and copovidone
- $m \in \{8, 9, 10, 11, 14, 15, 16, 17, 20, 21, 22, 23, 26, 27, 28, 29\}$

for $\mathbf{B}_2$

- $l = 2$, lower face concentrations
- $k \in [1, 2]$, bicalutamide and copovidone
- $m \in \{8, 9, 10, 11, 14, 15, 16, 17, 20, 21, 22, 23, 26, 27, 28, 29\}$.

This is testing whether the concentrations estimated for the tablet's upper and lower faces came from distributions with equal means.

2. Upper vs. lower faces. Same data selection as 1 but with $m = 1, ..., 36$, therefore $\mathbf{B}_1$ is equal to $\mathbf{v}_1^{(1)}$, (5.1.5), reshaped into (5.1.6) and $\mathbf{B}_2$ is created using $\mathbf{v}_1^{(2)}$. Here we are testing the entire estimated concentrations from the upper and lower faces.

3. Submatrix 15 vs. 16, lower face. Set $l = 1$ and $k \in [1, 2]$ with $m = 15$ for $\mathbf{B}_1$ and $m = 16$ for $\mathbf{B}_2$. This is testing a small region of the estimated concentrations on a single tablet face.

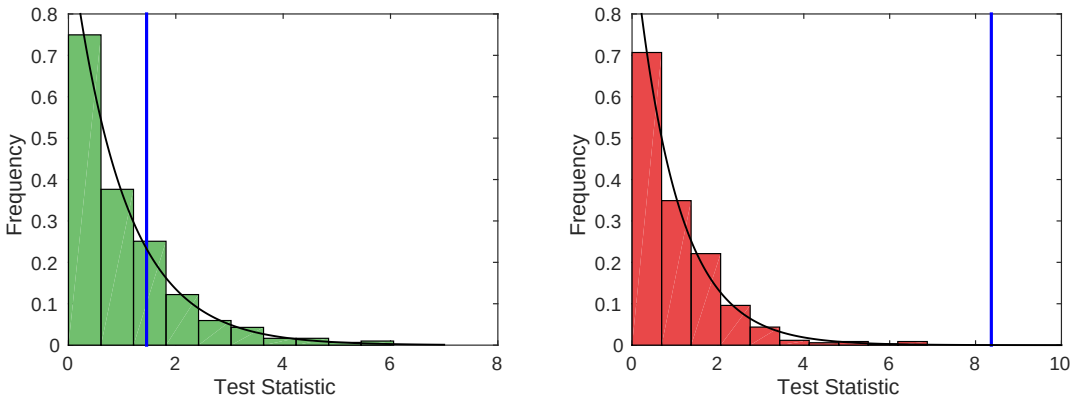
4. Submatrix 19 vs. 24, lower face. Set $l = 1$ and $k \in [1, 2]$ with $m = 19$ for $\mathbf{B}_1$ and $m = 24$ for $\mathbf{B}_2$. As with the previous region this is testing a different region of a single tablet face.

Our results from testing for equal means between these four concentration selections are shown in table 5.1. The first column, $F_{0.05}$, contains the p-values for a classical T^2 test calculated using (5.2.4). The second column, F_b, contains the p-values calculated using the bootstrap test algorithm 5.2.1. The conclusions from all the tests are not surprising when viewed with heatmap 5.1. We have strong evidence against equal means in concentration selections 1,2 and 4, and no evidence against equal means for selection 3. Histograms of the bootstrap sample F statistics, shown in figure 5.2, also show that the asymptotic distribution under H_0 was sufficiently accurate.

However whilst both tests performed exactly as expected we only tested one possible solution to (5.1.4) when there exists an infinite number of other estimates for the concentrations as shown by the solution space in Chapter 4. We therefore turn our attention in §5.3.3 to an identifiable model and a method of incorporating our solution space random walk, proposed in Chapter 4, with the hypothesis test.

Test area	$F_{0.05}$	F_b	$F_{0.05,PCA}$	$F_{b,PCA}$	$F_{b,RWBS}$
Inner 16 upper vs. lower	0	0.0010	0.00006	0.0010	0.0010
Upper vs. lower	0	0.0010	0.00006	0.0010	0.0010
Lower 15 vs. 16	0.2345	0.2655	0.37038	0.3513	0.2409
Lower 19 vs. 24	0.0003	0.0010	0.00006	0.0010	0.0511

Table 5.1: Hotelling's T^2 test theoretical and bootstrap p-values for equal population means, denoted $F_{0.05}$ and F_b respectively, with second subscripts denoting testing of PCA data and the random walk bootstrap test (algorithm 5.3.1)



(a) Lower section 15 vs. section 16. **(b)** Lower section 19 vs. section 24.

Figure 5.2: Histograms of bootstrap resamples, algorithm 5.2.1, with overlaid F-distributions (5.2.4). Blue lines show the observed F test statistics, (5.2.4), for the given concentration sections.

5.3.2 Principal Component Analysis Scores

As an alternative to testing estimated concentrations from (5.1.4) we also consider testing the $\mathbf{G}^{(l)}$ from

$$\underset{\mathrm{H,G}}{\operatorname{argmin}} \left\| \begin{bmatrix} \mathbf{Y}^{(1)} \\ \mathbf{Y}^{(2)} \end{bmatrix} - \begin{bmatrix} \mathbf{G}^{(1)} \\ \mathbf{G}^{(2)} \end{bmatrix} \mathbf{H}^T \right\| \tag{5.3.4}$$

where $\mathbf{H} = [\mathbf{h}_1 \ \ldots \ \mathbf{h}_r]$ with the constraints that

$$\begin{array}{lll} 1. & \|\mathbf{h}_k\| = 1 & \text{for all } k = 1, ..., r \\ 2. & \mathbf{h}_k^T \mathbf{h}_{k'} = 0 & \text{for all } k = 1, ..., r \text{ and } k' = 1, ..., r \text{ such that } k \neq k' \end{array} \tag{5.3.5}$$

which, as we show in §4.2, finds the identifiable solution of principal component analysis, §2.3, that can be transformed such that it solves (5.1.4).

With this model we consider the principal component loadings, $\mathbf{h}_k$, to be some linear combination of the component spectra, $\mathbf{S} = \mathbf{HR}$, as detailed in §4.2. Therefore the principal component scores, the $\mathbf{G}^{(l)}$, are of the form (5.1.3) and therefore equivalent to our concentration estimates. As such we calculate submatrices with (5.1.7) and perform hypothesis tests as in §5.3.1.

As in §5.3.1 we fit this model to our concatenated bicalutamide Raman images, with $r = 3$, and isolate the same submatrices from the $\mathbf{G}^{(l)}$. Our results from applying both a conventional T^2 test and a bootstrap test to the concentration subsets are shown in table 5.1 under columns $F_{0.05,PCA}$ and $F_{b,PCA}$ respectively. We see from these results that they concur with our testing of concentrations estimated with (5.1.4).

Whilst these results agree with our previous results and do not suffer from non-identifiability in the model from which they are calculated they are meaningless from a chemical standpoint. As we detail in §2.3.1 only the first principal component, $\mathbf{h}_1$ will always be entirely non-negative and therefore will be the only potentially correct spectrum estimate. The corresponding scores, equivalent to concentrations, may be physically plausible but the scores for the remaining principal components cannot be due to negativity.

5.3.3 Concentration Magnitudes from a Random Walk

As we show in §4.5 selecting a single solution to (1.2.2) can produce extremely varied results. Performing hypothesis tests on these results may therefore provide unreliable conclusions because they depend on a specific choice of solution. To combat this we propose an algorithm for combining our solution space random walk from §4.6.3 with the bootstrap

T^2 test. In doing so we will be testing a random selection of solutions from the solution space and so will obtain a range of conclusions to our hypothesis test, this will illustrate to us which of the conclusions are more common among the range of solutions. We propose a test of the hypotheses

$$H_0 : \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 \quad \text{vs.} \quad H_1 : \boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$$

by combining the standard bootstrapping, algorithm 5.2.1, with our random walk algorithm, 4.6.3. This combined algorithm will perform the bootstrap Hotelling's T^2 test on a variety of solutions to (1.2.2), giving us a p-value for each one. Our interpretation of these p-values is done by calculating the proportion of them which are below our significance level, for example 0.05, as this will tell us how many times our null hypothesis is likely to have been incorrect. It is important to note that there is no set way to interpret this result, unlike a single p-value, and so the user must decide upon an overall conclusion based upon how often the null hypothesis was accepted or rejected.

Algorithm 5.3.1. Random walk bootstrap T^2 test

1. If two Raman images are being compared concatenate the two data matrices as in (5.1.2).
2. For data, $\mathbf{Y}$, calculate the eigenvectors of $\mathbf{Y}^T\mathbf{Y}/n$ and set $\mathbf{K} = [\mathbf{k}_1 \ \ldots \ \mathbf{k}_r]$ where the $\mathbf{k}_k$ are the eigenvectors corresponding to the r largest eigenvalues.
3. Perform a random walk over the solution space in r-dimensions with algorithm 4.6.3 and cluster the results as in §4.6.2. This clustering step allows us some level or certainty that the r points we select from the solution space to form our new $\mathbf{S}$ will each represent a different component chemical.
4. Select r coordinates $\boldsymbol{\xi}_k$, each from a separate cluster, on the solution space to form
$$\Xi = [\boldsymbol{\xi}_1 \ \ldots \ \boldsymbol{\xi}_r]$$
and calculate spectra $\mathbf{S} = \mathbf{K}\Xi$.
5. Calculate $\mathbf{C}$ using (4.6.4)
$$\mathbf{C} = \Xi^{-1}\mathbf{K}^T\mathbf{Y}.$$
6. Separate the required concentrations from $\mathbf{C}$ as in §5.1.2 and combine into testing matrix (5.3.1).
7. Perform bootstrap T^2 test algorithm 5.2.1 and store the p-value, $p^{(t)}$.

8. Progress to iteration $t+1$, return to step (4) and repeat M times.

9. Calculate the proportion of p-values less than α to give an overall result for the test,

$$p = \frac{\sum_{t=1}^{M} \mathbb{I}_{p^{(t)}<\alpha}}{M}$$

where $\mathbb{I}$ is the indicator function. For example setting $\alpha = 0.05$ gives the proportion of tests which rejected the null hypothesis at a 5% level.

We apply this algorithm to the bicalutamide data, and concentration submatrices, introduced in §5.3.1. The first stages of the algorithm return us the solution plane shown in figure 5.3 where the point colour represents the cluster to which they are assigned based on the spectra they generate. These spectra are shown in figure 5.4 where we see the Raman spectra for bicalutamide, copovidone and the background signal. In testing the four concentration subsections used in §5.3.1 we obtain the p-values shown in the final column of table 5.1, $F_{b,RWBS}$. The first three of these p-values concur with all previous methods of testing however it is in comparing sections 19 and 24 on the lower face of the tablet where we see a difference, with a 50 times increase in the p-value when compared to the bootstrap test for a single solution to (5.1.4). We see in figure 5.5 the reason for these changes in p-value as we show histograms of all 500 p-values estimated during the random walk bootstrap test. The range of p-values obtained from testing section 19 against 24 shows that certain solutions to (5.1.4) have concentrations which are deemed to be similar while the others produce significantly different concentrations.

5.4 Testing the Uniformity of Component Mixing in the Subject of a Raman Image

Our second area of interest for testing is the spatial information contained in the estimated concentrations, leading us to the question we posed in §5.1. How uniformly mixed is the subject? For example do amorphous production methods produce a more thoroughly mixed product than standard crystalline methods?

The mixing of a tablet is of great interest to the pharmaceutical industry because it has a large effect on the dissolution of the drug, and a consistently dissolving drug provides the most efficient delivery. We detail current pre and post production mixing testing in §5.1 however we investigate a potential alternative where we analyse the mixture of the chemical components from a spectroscopic scan. From this analysis we will be able to detect poor mixing, which will have a detrimental effect on the dissolution. We also note

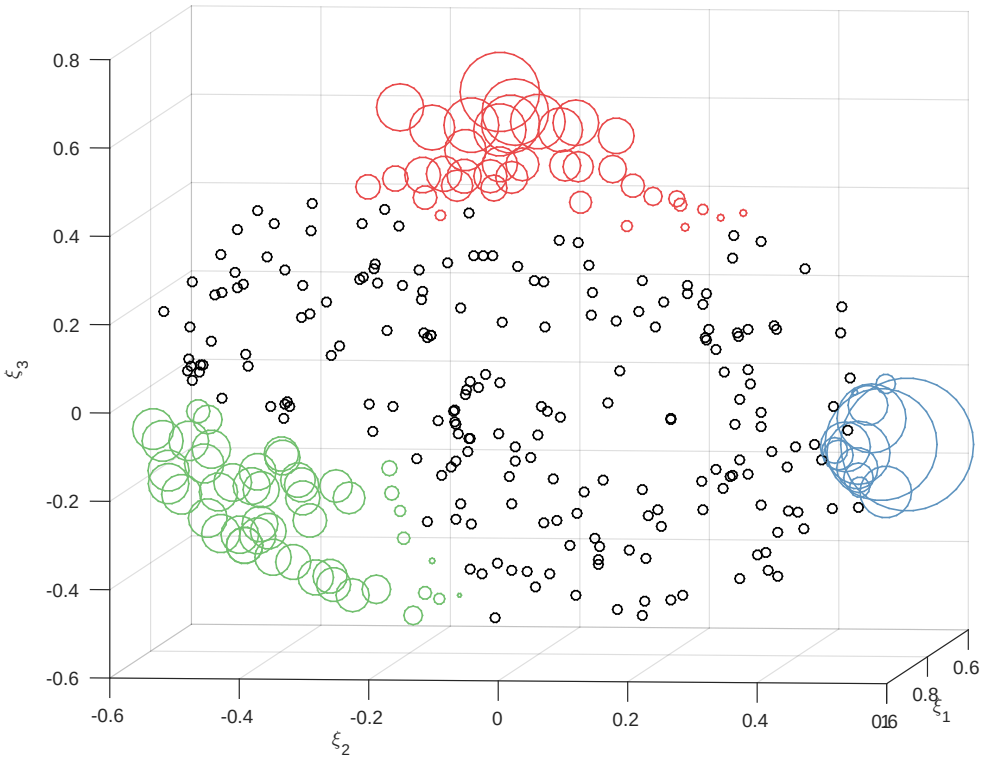
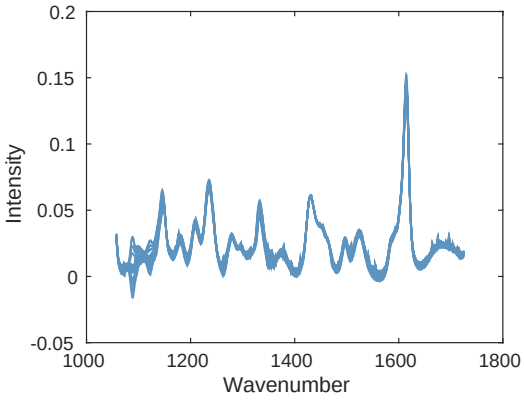
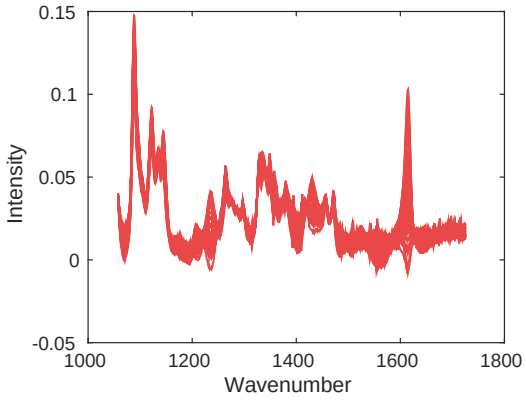


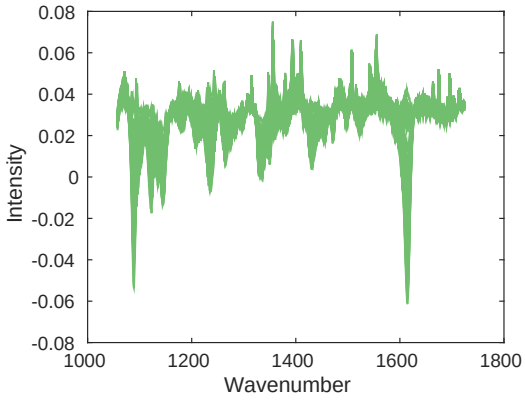
Figure 5.3: Random walk space for bicalutamide data with coordinates resulting in accepted spectra, non-negative, unit **S**, with non-negative **C**. Coordinates coloured by applying UPGMA clustering, §2.11, to the corresponding spectra, detailed in §4.6.1. Clusters allow us to select a single one of each component when selecting a solution to (1.2.2).



(a) Bicalutamide

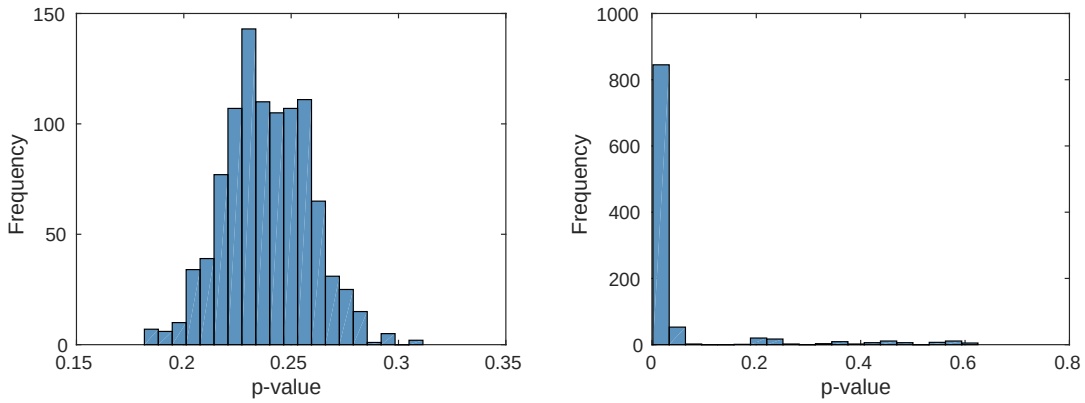


(b) Copovidone



(c) Background signal

Figure 5.4: A selection of each component spectra obtained by random walk on the solution space for the bicalutamide data, §4.6.3, clustered by UPGMA. Although similar in shape the spectra can clearly change depending on their position on the solution space, this causes different concentration estimates and therefore potentially different hypothesis test conclusions.



(a) Lower section 15 vs. section 16. **(b)** Lower section 19 vs. section 24.

Figure 5.5: Histograms of p-values from random walk bootstrap testing. Testing 15 against 16 gives a consistent rejection of H_1, the test comparing 19 to 24 however shows potential for rejecting either H_0 or H_1.

that our testing is less likely to suffer from the poor repeatability of the current techniques [21, 58, 65] as the spectroscopic scan does not involve any flow of the chemicals. This flow will likely vary between different tablets due to the movement of the solvent where as a Raman image is static.

5.4.1 Algorithm for Testing the Hypothesis of Perfect Mixing

The method with which we will test our hypothesis of perfect mixing takes into consideration the bilinear model non-identifiability and is based on algorithm 5.3.1 for testing the means of concentrations in §5.3. We also note that perfect mixing may appear similar to uniform i.i.d. values, as all concentration levels of all possible chemical components are equally likely, or as an image of equal valued pixels if a single chemical component is present. In this latter case and in the context of our model the concentrations are estimated subject to random error and will therefore appear similar to random values.

Our data in this test is in the form $\mathbf{v}_k^{(l)}$, (5.1.5), a vector of concentrations of a single chemical component at each pixel, reshaped into $x \times y$ matrix $\mathbf{V}$, (5.1.6), which we can visualise as a heatmap of concentration levels. In the context of this chapter therefore our "image" is simply a matrix which we will analyse with image analysis techniques. We assume $\mathbf{V}$ is a random variable from some unknown distribution and define our hypothesis test as,

$$H_0 : F\left(\mathrm{E}\left[\mathbf{V}\right]\right) = F\left(\mathrm{E}\left[\mathbf{V}^*\right]\right) \quad \text{vs.} \quad H_1 : F\left(\mathrm{E}\left[\mathbf{V}\right]\right) \neq F\left(\mathrm{E}\left[\mathbf{V}^*\right]\right) \tag{5.4.1}$$

using test statistic F which quantifies how randomly distributed the pixels are in an image.

Here $\mathbf{V}$ is our original concentration matrix and $\mathbf{V}^*$ is a random permutation of our data calculated with

$$V^*_{q,t} = V_{q',t'} \quad \text{for } q \in \{1,...,x\}, t \in \{1,...,y\}, q' \in \{1,x\} \text{ and } t' \in \{1,y\} \tag{5.4.2}$$

where elements are removed from the sets $\{1,x\}$ and $\{1,y\}$ so that each element of $\mathbf{V}$ appears only once in $\mathbf{V}^*$ in a random location. In using this random permutation of our matrix $\mathbf{V}$ any structures or patterns in the original data will be removed and so the test statistic will change indicating poor mixing.

The following algorithm combines the random walk used in §5.3.3 with the previously defined permutation test for uniform mixing of elements in a matrix however now uses a test statistic quantifying any structures in the image $\mathbf{V}$. These test statistics will be defined in §5.4.3.

Algorithm 5.4.1. Test for perfect mixing

1. For data, $\mathbf{Y}$, calculate the eigenvectors of $\mathbf{Y}^T\mathbf{Y}/n$ and set $\mathbf{K} = [\mathbf{k}_1 \ ... \ \mathbf{k}_r]$ where the $\mathbf{k}_k$ are the eigenvectors corresponding to the r largest eigenvalues.

2. Perform a random walk over the solution space in r-dimensions with algorithm 4.6.3 and cluster the results as in §4.6.2.

3. Select r coordinates $\boldsymbol{\xi}_k$, each from a separate cluster, on the solution space to form

$$\Xi = [\boldsymbol{\xi}_1 \ ... \ \boldsymbol{\xi}_r]$$

   and calculate spectra $\mathbf{S} = \mathbf{K}\Xi$.

4. Calculate $\mathbf{C}$ using (4.6.4)

$$\mathbf{C} = \Xi^{-1}\mathbf{K}^T\mathbf{Y}.$$

5. Select the required concentrations, $\mathbf{v}_k$, as in (5.1.5), and reshape into $\mathbf{V}$ with (5.1.6) to obtain heatmap.

6. Perform a permutation test of perfect mixing using the chosen test statistic F,

   (a) Calculate $F(\mathbf{V})$

   (b) Randomly sample $\mathbf{V}$ without replacement, using (5.4.2), to obtain $\mathbf{V}^*$ and calculate $F(\mathbf{V}^*) = F^{(m)}$

   (c) Repeat step (b) for $m = 1, ..., M$

(d) Calculate the p-value using the one-sided formulae

$$p_l = \frac{1 + \sum_{m=1}^{M} \mathbb{I}(F^{(m)} > F)}{1 + M} \tag{5.4.3}$$

or

$$p_l = \frac{1 + \sum_{m=1}^{M} \mathbb{I}(F^{(m)} < F)}{1 + M} \tag{5.4.4}$$

or two-sided formula

$$p_l = \frac{1 + 2\min\left(\sum_{m=1}^{M} \mathbb{I}(F^{(m)} < F), \sum_{m=1}^{M} \mathbb{I}(F^{(m)} > F)\right)}{1 + M} \tag{5.4.5}$$

where appropriate for the chosen test statistic.

7. Return to step (3) and repeat N times to obtain N p-values.

8. Calculate the proportion of p-values less than α to give an overall result for the test,

$$p = \frac{\sum_{l=1}^{N} \mathbb{I}_{p_l < \alpha}}{N}$$

where $\mathbb{I}$ is the indicator function. Again α represents the significance level at which we wish the null hypothesis to be rejected and so p is the proportion of tests satisfying that criteria.

As with algorithm 5.3.1 this gives us a proportion of p-values below our significance level giving us an indication of how often the null hypothesis was rejected.

Before we can implement this algorithm we must first define our test statistic $F(\cdot)$ which we will do using some existing image and shape analysis techniques for defining features. A feature extraction method is a function or process such that we quantify elements in an image, for example regions of similar colour or dependencies between areas of the image.

5.4.2 Feature Extraction Methods

In devising our test statistics for hypothesis (5.4.1) we will consider three methods of quantifying how well mixed an image is when the image is in the form of matrix (5.1.6) which we visualise as a heatmap. The first two of these are standard image and shape analysis techniques with a choice of summary statistics available for each. For these methods we will need to combine them in such a way as to obtain a single test statistic. The third method we consider is less widely used in image analysis and involves the correlation between rows or columns of a matrix. We must therefore devise novel methods of forming this calculation into a test statistic.

Minkowski Functionals

The first features we investigate are Minkowski functionals which are functions for describing distance in a linear space. Certain 2-dimensional Minkowski functionals are commonly used in image analysis [46] for quantifying features of a binary image such as the perimeter and area of regions of equal valued pixels, in our interpretation these are poorly mixed regions comprised of a single component chemical. Three of the possible functionals for any $x \times y$ 2-dimensional binary image, $\mathbf{X}$, are defined by Legland et al. [66]

1. Area - the sum of the pixels in the image, $A(\mathbf{X}) = \sum_{i,j} x_{i,j}$.

2. Perimeter - calculated using a discretised version of the Crofton formula, a formula used to calculate the length of a curve [23]. This results in a formula requiring the number of pixels connected along orthogonal and diagonal lines by pixels of the same value. We calculate the number of horizontally connected pixels using

$$P_h = \sum_{i,j} x_{i,j} - \sum_{i,j} \mathbb{I}\left(x_{i,j} \neq x_{i+1,j}\right) \tag{5.4.6}$$

   which is easily modified to check for vertical and diagonal pairs of pixels by changing the indices in the indicator function. Thus the perimeter is equal to

$$P = \frac{\pi}{4}\left(P_h + P_v + P_{d_1} + P_{d_2}\right)$$

   where P_v, P_{d_1} and P_{d_2} are the numbers of vertically and diagonally connected pixels calculated using (5.4.6) with altered indices.

3. Euler-Poincaré characteristic - the standard Euler characteristic is $\chi(\mathbf{X}) = V - E + F$ where V, E and F are the numbers of vertices, edges and faces respectively [30]. This function is used to describe a shape regardless of it's form, for example the Euler characteristic of a 3-dimensional cube is the same as that cube represented as a 2-dimensional template. Whilst our images are strictly 2-dimensional it may still prove useful in forming a test statistic $F(\cdot)$.

   Estimates of this function for binary 2-dimensional images require the level of connectivity to be set, this is either 4 or 8 where 4 implies just horizontal and vertical lines and square faces are used and 8 also includes diagonal lines. For the simple 4-connectivity case the three characteristic elements are

   - Vertices - same as the Area functional, $V = \sum_{i,j} x_{i,j}$
   - Edges - number of vertical and horizontal pairs of pixels which are equal, $E = \sum_{i,j}\left[\mathbb{I}\left(x_{i,j} = x_{i+1,j}\right) + \mathbb{I}\left(x_{i,j} = x_{i,j+1}\right)\right]$

- Faces - number of groups of 4 pixels all equal,
  $F = \sum_{i,j} \mathbb{I}\left(x_{i,j} = x_{i+1,j} = x_{i,j+1} = x_{i+1,j+1}\right)$.

  As before we can modify this to estimate the Euler-Poincaré characteristic under the assumption that $\mathbf{X}$ is not the full image. In this case we set $\tilde{\chi}(\mathbf{X}) = \chi(\mathbf{X}) - \chi_i(\mathbf{X}) - \chi_j(\mathbf{X}) + \chi_{ij}(\mathbf{X})$ where χ_i and χ_j are the average Euler-Poincaré values calculated for the first and last rows and columns separately and χ_{ij} is the average value for the corners. The result of this is that the effect of the image border is removed from the Euler-Poincaré characteristic.

With the Minkowski functionals defined we must now consider how they apply to continuous data. As they require binary data it is necessary for us to convert our continuous data, $\mathbf{V}$, into a binary vector. This is most easily achieved by selecting a threshold value, u, and setting

$$V_{i,j} = \begin{cases} 1 & \text{if } V_{i,j} > u \\ 0 & \text{if } V_{i,j} \leq u \end{cases} . \tag{5.4.7}$$

Our second method for converting $\mathbf{V}$ to binary is to use a clustering technique such as EMGM, detailed in §2.12.1, and replacing each $V_{i,j}$ with the assigned cluster index, 0 or 1. Using a clustering method to obtain binary data is preferable as it will group based on the chemical concentrations in the scan subject whereas a simple threshold requires user input in deciding the threshold value, u.

Grey-level Co-occurence Matrices

The second feature extraction method of interest to us uses a grey-level co-occurance matrix (GLCM). The advantage of this over the Minkowski functional based test statistic is that we require greyscale data instead of binary. This introduction of levels may provide us with a test statistic that is more descriptive of the structure of the image than that of §5.4.2 as the levels will allow for more detailed structures within the image. Accuracy should improve as binary data only allows us two levels of chemical concentration and so may miss areas of high or low concentrations within one of the two groups. Creating a binary image is a similar process to the thresholding introduced for the Minkowski functional.

For some threshold values $u_1 < u_2 < ... < u_{l-1}$ we replace the $v_{i,j}$ with,

$$V_{i,j} = \begin{cases} 1 & \text{if } V_{i,j} \leq u_1 \\ 2 & \text{if } u_1 < V_{i,j} \leq u_2 \\ \vdots & \\ l-1 & \text{if } u_{l-1} < V_{i,j} \leq u_{l-1} \\ l & \text{if } V_{i,j} > u_{l-1} \end{cases}.$$

A grey-level co-occurrence matrix takes the form of an $l \times l$ matrix where l is the number of grey levels in the image to be analysed. Entries in a GLCM indicate the frequency with which grey levels appear next to each other in a given direction. For example in an east GLCM if a level one appears to the left of a level two the value in the GLCM entry $(1, 2)$ increases by one. Alongside controlling the direction of the comparison, we can set the distance from the reference pixel to the neighbouring pixel, known as the offset. The East-West GLCM with offset one of the simple three level 4×4 matrix in figure 5.6 is shown in table 5.2. Here, for example, we see that a level one pixel appears directly to the left or right of a level three pixel four times.

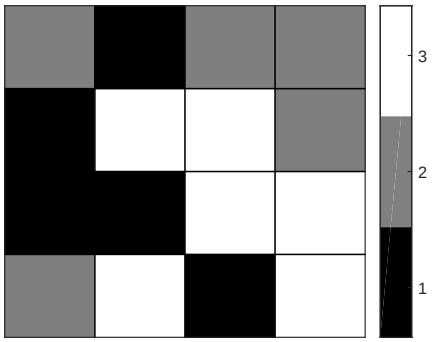


Figure 5.6: 4×4 greyscale matrix.

		Neighbour Level		
		1	2	3
Reference Level	1	2	2	4
	2	2	2	2
	3	4	2	4

Table 5.2: East-West Grey-level Co-occurrence Matrix for figure 5.6.

Testing for uniformity of mixing using GLCMs is not immediately simple as a direct comparison of GLCMs is difficult to interpret. We will therefore use summary statistics, or texture measures, of the GLCM to create our test statistic. We have numerous measures available to us to describe the information contained in a GLCM. A large number of GLCM properties are detailed by Albregtsen [4], of which we consider eight in this investigation. For these properties the GLCM, $\mathbf{G}$, must be normalised, meaning each element is divided by the total sum of the matrix.

1. *Energy*, defined by
$$\sum_{i,j} g(i,j)^2,$$
measures the homogeneity of the image. A homogeneous image will have only a few grey levels appearing frequently resulting in a high sum of squares for the GLCM. However an inhomogeneous image will have almost equal GLCM entries and a lower sum of squares. For the following definitions μ_i, μ_j, μ, σ_i and σ_j are the row, column and overall means and standard deviations respectively.

2. *Contrast*, defined by
$$\sum_{i,j}(i-j)^2 g(i,j),$$
measures local intensity variation, upscaling off diagonal entries of G.

3. *Local Homogeneity*, defined by
$$\sum_{i,j} \frac{1}{1+(i-j)^2} g(i,j),$$
is similar in principal to energy however uses the scaling factor $(i-j)$ to reduce the effects of inhomogeneous areas.

4. *Entropy*, defined by
$$-\sum_{i,j} g(i,j)\log(g(i,j)),$$
is another measure of homogeneity where an inhomogeneous image will result in small values in the GLCM and thus a higher entropy, i.e. smaller negative number.

5. *Correlation*, defined by
$$\sum_{i,j} \frac{ijg(i,j)-\mu_i\mu_j}{\sigma_i\sigma_j},$$
measures the dependence between pixels using their positions relative to each other.

6. *Variance*, defined by
$$\sum_{i,j}(i-\mu)^2 g(i,j),$$
puts higher weights on the elements of G that differ from the average.

7. *Cluster Shade*, defined by
$$\sum_{i,j}(i+j-\mu_x-\mu_y)^3 g(i,j),$$
weights elements differing from the mean, like variance, however includes spatial information by incororating the coordinates (i,j) of pixel $g(i,j)$ in the formula.

8. *Cluster Prominence*, defined by

$$\sum_{i,j}(i+j-\mu_i-\mu_j)^4 g(i,j),$$

uses an alternate weighting measure to cluster shade in order to increase contributions from elements of G differing from the spatial means.

Neighbour Correlations

The final method we consider for quantifying pixel mixing involves calculating the correlation between neighbouring columns of an image, $\mathbf{V}$. The idea behind this is that an image with little structure to the pixels will have low correlation between the columns. For example in the context of our tablet data the matrix $\mathbf{V}$ will represent the concentrations of a particular component chemical at each pixel of the Raman image. Therefore a pattern in these concentrations, for example a region of consistently high concentration, will appear as correlated columns.

Our motivation for this method was to derive a test statistic requiring no thresholds or grouping of data into levels. This allows us to totally remove the arbitrary choice of levels from our testing and fully take advantage of the continuous nature of our data. To describe the structure of the data will use Pearson's correlation coefficient, ρ, as comparisons between two vectors are simplified due to the limits of -1 and 1 and there is a known link with Student's t-distribution [73]. This relationship with Student's t-distribution is possible if, under the null hypothesis of $\rho = 0$ the two vectors come from an uncorrelated bivariate normal distribution, for a large enough sample this may be an asymptotic distribution by the central limit theorem. Therefore under the null hypothesis

$$r\sqrt{\frac{n-2}{1-r^2}} \sim t_{n-2} \tag{5.4.8}$$

where n is the length of our vectors.

We define our method for an image, $\mathbf{V}$ as in (5.1.6), in the form of a matrix of width x and height y with columns defined as

$$\mathbf{p}_i = \begin{bmatrix} V_{1,i} \\ \vdots \\ V_{y,i} \end{bmatrix}.$$

We calculate the correlation between each matrix column and the immediate neighbour to the right, working our way across the matrix $\mathbf{V}$ from left to right, using the two step process

Algorithm 5.4.2. 1-step Neighbour Correlation

1. Create two vectors, $\mathbf{q}_1$ and $\mathbf{q}_2$, by stacking columns $(1, ..., x-1)$ and $(2, ..., x)$ respectively,

$$\mathbf{q}_1 = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{x-1} \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_x \end{bmatrix}.$$

   Therefore when we calculate correlation the columns $\mathbf{p}_1$ and $\mathbf{p}_2$ are paired, then $\mathbf{p}_2$ and $\mathbf{p}_3$ and so on until the penultimate and final columns. This gives us the correlation between each column and it's immediate neighbour.

2. Calculate correlation, ρ using

$$\rho = \frac{(\mathbf{q}_1 - \bar{q}_1\mathbf{1}_{x-1})^T(\mathbf{q}_2 - \bar{q}_2\mathbf{1}_{x-1})}{\sqrt{(\mathbf{q}_1 - \bar{q}_1\mathbf{1}_{x-1})^T(\mathbf{q}_1 - \bar{q}_1\mathbf{1}_{x-1})}\sqrt{(\mathbf{q}_2 - \bar{q}_2\mathbf{1}_{x-1})^T(\mathbf{q}_2 - \bar{q}_2\mathbf{1}_{x-1})}}. \tag{5.4.9}$$

Whilst simple to implement and fast to compute we note that this correlation only involves immediate neighbours. Taking motivation from the offset of the GLCM method in §5.4.2 we modify our calculations to include comparisons of all possible combinations of columns in matrix $\mathbf{V}$. Results of these calculations are stored in the $x \times p$ strictly upper triangular matrix $\mathbf{N}$, with entries calculated using

Algorithm 5.4.3. Multi-step Neighbour Correlation

1. Select offset, i, from set $\{1, ..., x-1\}$.

2. Create two vectors, $\mathbf{q}_1$ and $\mathbf{q}_2$, concatenating reference columns and i^{th} neighbour columns from image $\mathbf{V}$,

$$\mathbf{q}_1 = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{x-i} \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} \mathbf{p}_{i+1} \\ \vdots \\ \mathbf{p}_x \end{bmatrix}. \tag{5.4.10}$$

3. Calculate correlation ρ_i using (5.4.9), with $\mathbf{1}_{x-1}$ replaced by $\mathbf{1}_{x-i}$.

4. Return to step 1 until all offsets have been calculated and vector $\boldsymbol{\rho}$ is complete.

Whilst this calculates the correlation of all possible combinations of columns in the image we see that the length, $(x-i)y$, of our vectors, $\mathbf{q}$, decreases as the distance between neighbours, i, increases. An obvious occurrence when at the maximum distance we are comparing only the leftmost column with the rightmost column. This difference in vector

length for each offset value means we are unable to reliably compare the values of (5.4.8) for each ρ_i. This is because the bivariate normal distribution assumption of the t-distribution may only be an asymptotic result, therefore differing sample sizes may affect the extent to which our data follows the required distribution.

We avoid this by defining (5.4.10) as in the previous algorithm but then randomly selecting y pairs of elements to be used in calculating the correlation. This gives equal sample sizes for all correlation calculations and gives us our final algorithm.

Algorithm 5.4.4. Multi-step Neighbour Correlation with Fixed Sample Size

1. Select offset, i, from set $(1, ..., x-1)$.
2. Create two vectors, $\mathbf{q}_1$ and $\mathbf{q}_2$, concatenating reference columns and i^{th} neighbour columns from image $\mathbf{V}$,
$$\mathbf{q}_1 = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{x-i} \end{bmatrix}, \quad \mathbf{q}_2 = \begin{bmatrix} \mathbf{p}_{i+1} \\ \vdots \\ \mathbf{p}_x \end{bmatrix}.$$
3. Randomly select y pairs from vectors $\mathbf{q}_1$ and $\mathbf{q}_2$ to give $\mathbf{q}_1'$ and $\mathbf{q}_2'$.
4. Calculate correlation ρ_i using (5.4.9), with $\mathbf{1}_{x-1}$ replaced by $\mathbf{1}_y$
5. Return to step 1 until all offsets have been calculated and vector $\boldsymbol{\rho}$ is complete.

The result of this algorithm is a vector indicating how the correlation between columns of a matrix, or image, changes as the gap between columns increases. A perfectly mixed image should have low correlations throughout however spikes or trends in the correlations will indicate patterns in the data.

A final and important feature of this method we note is how it will be affected by data displaying a vertically striped pattern. Due to the method working across columns we will miss within column correlation, for example a single column of equal valued pixels will not be detected providing it is uncorrelated with the other columns. To combat this we can transpose our matrix image $\mathbf{V}$ as this does nothing to the structure of the image however columns become rows and vice versa.

5.4.3 Deriving a Test Statistic from Feature Extraction Methods

Before we can apply our feature extraction methods in algorithm 5.4.1 we must first form test statistics, F, capable of distinguishing between perfectly and poorly mixed data in our

hypothesis (5.4.1). For the Minkowski functionals and GLCM properties this will involve finding the optimal linear combination of the functions for distinguishing between perfectly and poorly mixed data. To this end we will use linear discriminant analysis (LDA).

The purpose of LDA is to find a linear combination of measures which best characterise groups of data [73]. For g groups of data with common covariance, Σ, the groups, $(\mathbf{X}_1, ..., \mathbf{X}_g)$, are defined as

$$\mathbf{X}_h = \begin{bmatrix} \mathbf{x}_{h,1}^T \\ \vdots \\ \mathbf{x}_{h,n_h}^T \end{bmatrix}$$

where the columns represent our measures and the rows our observations. We calculate the group means

$$\boldsymbol{\mu}_h = \frac{1}{n_h} \sum_{i=1}^{n_h} \mathbf{x}_{h,i}$$

and define

$$\Sigma_b = \frac{1}{g} \sum_{h=1}^{g} (\boldsymbol{\mu}_h - \boldsymbol{\mu})(\boldsymbol{\mu}_h - \boldsymbol{\mu})^T$$

where

$$\boldsymbol{\mu} = \frac{1}{\sum_h n_h} \sum_{h=1}^{g} \sum_{i=1}^{n_h} \mathbf{x}_{h,i}$$

is the overall mean of all the groups combined. We now calculate our linear combination of features by finding $\mathbf{K}$ and Λ such that

$$\Sigma^{-1}\Sigma_b = \mathbf{K}\Lambda\mathbf{K}^T \tag{5.4.11}$$

where the columns of $\mathbf{K}$, $\mathbf{k}_j$, are eigenvectors and the diagonal entries of Λ follow $\lambda_1 > \lambda_2 > ... > \lambda_p$. We now select $\mathbf{k}_1$, the eigenvector of $\Sigma^{-1}\Sigma_b$ corresponding to the largest eigenvalue, which gives us the coefficients of our linear combination of measures for best separating a point of data from our g groups. As Σ is often unknown we can use the standard sample covariance estimate. The result of this algorithm is similar to that of principal component analysis, §2.3, as it gives us a new basis for our data. However now this basis is found such that it maximises the explained variance between different sets of data, instead of within a single dataset, it gives us the linear combination of measures which best describe each of our datasets.

To devise these test statistics we simulate data to represent perfectly and poorly mixed images.

- Perfectly mixed simulated data: 50 matrices with dimension 50×50 and $\mathcal{U}(0,1)$ entries.
- Poorly mixed: 50 2-dimensional multivariate normal PDFs calculated on a 50×50 mesh for poorly mixed data. Images, $\mathbf{G}$, generated by

$$G_{i,j} = f\left((i,j)|\boldsymbol{\mu}, \Sigma\right) \quad \text{for } i = \{1, ..., 50\},\ j = \{1, ..., 50\}$$

  where $f\left(\mathbf{x}|\boldsymbol{\mu}, \Sigma\right)$ is the probability distribution function of a $\mathcal{N}_2\left(\boldsymbol{\mu}, \Sigma\right)$ distribution. Therefore for each combination of coordinate points, (i,j), we obtain the probability distribution function value, for the 2-dimensional Normal distribution, at that pixel. This probability becomes the value of pixel (i,j) in our image matrix $\mathbf{G}$. We then add $\mathcal{N}(0,1)$ noise to the GRF data to more closely simulate real world data.

Examples of these two datasets are shown as heatmaps in figures 5.7b and 5.7d.

Minkowski Functional Test Statistic

The first test statistic we form uses the Minkowski functionals and EMGM for obtaining a binary image. For each of our 50 simulated uniform and Gaussian datasets, converted to binary using EMGM groups, we calculate the three Minkowski functionals. This gives us two 50×3 matrices to use in our linear discriminant analysis for finding the best linear combination of the functionals for distinguishing between perfectly and poorly mixed data. The result of our linear discriminant analysis, (5.4.11), was the eigenvector $\mathbf{k}_1 = (-0.1077, 0.9916, -0.0712)$ giving us our new basis vector, or alternatively the coefficients of our linear combination. We now define our final test statistic as

$$F_{mink} = -0.1077 * A + 0.9916 * P - 0.0712 * E \tag{5.4.12}$$

for A, P and E the area, perimeter and Euler-Poincaré characteristics respectively. To use this in algorithm 5.4.1 we must also take care in selecting the appropriate p-value for the test statistic, one or two sided. We can see a reason why perimeter dominates the test statistic, with a coefficient of 0.9916, when we consider that perfect mixing implies each pixel contributes one to the perimeter whereas the inner pixels of a group of equal valued pixels add nothing to the total. As perfect mixing will increase the test statistic value we will therefore use the one sided p-value, (5.4.4), in the permutation test. The comparatively low values in the linear combination of area and Euler-Poincaré may allow us to completely exclude them from the test statistic however calculation is simple and fast so including them for completeness is simple.

Grey-Level Co-occurrence Matrix Test Statistic

Our second test statistic combines the texture measures of a GLCM. As with the Minkowski functionals we select the optimum combination for detecting perfectly mixed data using LDA. However now we include the offset levels in the LDA to give a more complete representation of the image as looking at an offset of one may miss larger features in the image. These offset levels are detailed in §5.4.2 and describe the distance from the reference pixel we look in calculating our GLCM. We calculate the 8 GLCM texture measures over offsets $1, ..., 5$ for each of our simulated datasets. This gives us two $50 \times (8 * 5)$ matrices, representing the perfectly mixed and poorly mixed data.

In calculating all eight GLCM properties at offsets of one to five the result from our LDA was a vector, $\mathbf{k}$, of length 40. To reduce this and avoid including unnecessary properties which would contribute little to the test statistic we limited this to values greater than 0.05 as this left us a computationally fast, and more easily interpretable, five measures for our linear combination. The remaining five measures indicated that in distinguishing between the datasets energy was important at offsets of 2, 4 and 5 whilst entropy was important at offsets of 1 and 2. With our optimum linear combination of texture measures we now define our test statistic,

$$F_{GLCM} = 0.3520 * E_1 + 0.0548 * E_2 + 0.3976 * N_2 + 0.3340 * N_4 + 0.0576 * N_5 \tag{5.4.13}$$

where N_o and E_o are energy and entropy, at offset o, respectively. For a perfectly mixed image energy will return it's lowest value, for example a GLCM with every element equal will have a lower energy (the sum of the squares of all entries) than a GLCM with all entries except one equal to zero. The converse of this is true for entropy as a poorly mixed image will produce a large negative value for entropy getting less negative with better mixing. Therefore as energy and entropy return lower and higher values respectively for well mixed images we have no clear upper or lower bound for perfectly mixed data and thus do not know whether rejecting H_0 corresponds to a small or large value of $F(\cdot)$. Therefore we use the two sided p value formula, (5.4.5), in algorithm 5.4.1. When using this test statistic however we must bear in mind that we only used offsets 1,...,5. For a large Raman image an offset of 5 may not be enough to accurately measure patterns in the pixels.

Neighbour Correlation Test Statistic

Unlike our test statistics based on the Minkowski functionals and GLCM texture measures, we have no clear features of the neighbour correlation calculations on which to base our test

statistic. Our data from this feature extraction method is in the form of a vector which we can see as a function of the distance, i, between columns. An example for this is in figure 5.7 where we have plotted our vectors, $\boldsymbol{\rho}$, against the distance between columns, i, for a $\mathcal{U}(0,1)$ matrix and a $\mathcal{N}_2(\boldsymbol{\mu}, \Sigma)$ matrix as defined in §5.4.3. We see clearly here that the covariance between the two elements of the Gaussian distribution has introduced a strong relationship between elements of the matrix and thus pixels of the heatmap. Whereas the near perfectly mixed dataset has little correlation between columns of the matrix illustrated by the random pixel values in the heatmap.

To use these values we have devised three possibilities for obtaining a single test statistic from our vector of correlations, $\boldsymbol{\rho}$. Two of our proposed test statistics use the link between correlation coefficient ρ and Student's t-distribution defined as (5.4.8). Using (5.4.8) and given a matrix of dimensions $x \times y$ we can calculate the critical values of ρ for the hypothesis test that $\rho = 0$ [33, 34]. We will therefore have the range of values ρ may take yet still allow us to accept the null hypothesis. We obtain this range by rearranging (5.4.8) to give

$$\pm \frac{t_{y-2,0.95}}{y - 2 + t^2_{y-2,0.95}}. \tag{5.4.14}$$

The first test statistic we define for use in algorithm 5.4.1 is the minimum distance between columns until the first estimated correlation falls within the 95% acceptance region

$$F_{accept} = \min\{i | \rho_i \in \mathrm{A}\}. \tag{5.4.15}$$

We see an example of this in figure 5.7a at a distance of around 12. Our second test statistic is the number of estimated correlations lying within the confidence region (5.4.14)

$$F_{total} = \sum_{i=1}^{x-1} \mathbb{I}_{\rho_i \in \mathrm{A}}. \tag{5.4.16}$$

where x is the heatmap width and length of $\boldsymbol{\rho}$. The final test statistic we consider is the difference between the maximum and minimum estimated correlations

$$F_{diff} = \left| \max_i(\rho_i) - \min_i(\rho_i) \right|. \tag{5.4.17}$$

For these test statistics both one sided p-value formulae, (5.4.3) and (5.4.4) are required. The distance between columns value, F_{accept}, takes values greater than or equal to one with one corresponding to perfect mixing as immediate neighbours are uncorrelated. We therefore reject hypotheses of perfect mixing in the case of a high test statistic so use (5.4.3) to see how likely we are to obtain a value higher than our test statistic value. We use this same p-value formula for the difference between the maximum and minimum correlations, F_{diff}. This test statistic takes values greater than or equal to zero where zero

implies perfect mixing as the correlation between columns across the image is constant. The total acceptances statistic, F_{total}, however is limited above at $(x-1)$, therefore we must use (5.4.4), since a low value of the test statistic compared to it's values under the null distribution are evidence to reject the hypothesis of perfect mixing.

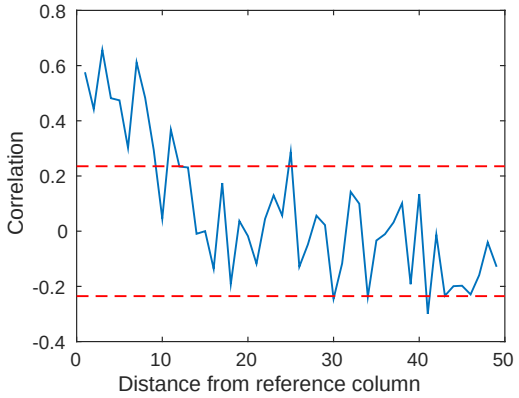
With these three test statistics we must also take into consideration that two are discrete, n_{accept} and n_{total}, and the other continuous, n_{diff}. Guerra et al. [41, 42] and Lloyd [68] investigated permutation testing on discrete values and concluded that whilst it provided acceptable results the results could be improved by kernel smoothing the values to simulate resampling from a continuous distribution. Including this smoothing kernel into algorithm 5.4.1 could overcomplicate the test with further variables so n_{diff} may prove the most reliable statistic after testing.

5.4.4 Error Rate Estimation

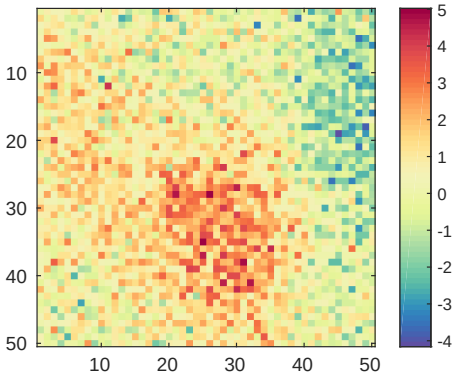
With our test statistics defined we can now use step 6 of algorithm 5.4.1 to estimate the Type I and II errors for our hypothesis (5.4.1). In order to estimate error rates we require data generated under the null and alternative hypotheses, therefore we once again use the $\mathcal{U}(0,1)$ and $\mathcal{N}_2(\boldsymbol{\mu}, \Sigma)$ data described in §5.4.3. To estimate the error rates we used the following method.

Algorithm 5.4.5. Type I and II Error Rate Estimation

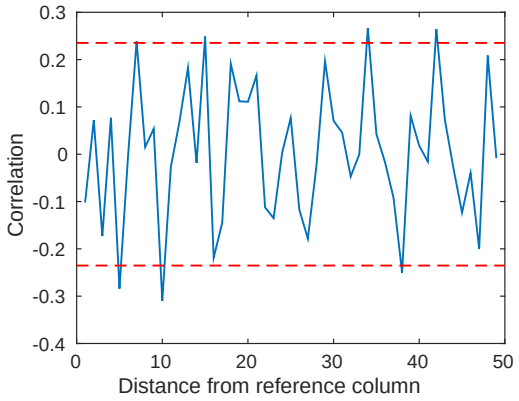
1. Select the test statistic F from §5.4.3.
2. Perform the permutation test of perfect mixing, step 6 of algorithm 5.4.1, for each $\mathcal{U}(0,1)$ dataset to obtain a vector, $\mathbf{p}_0$, of p-values calculated under H_0.
3. Perform the permutation test of perfect mixing, step 6 of algorithm 5.4.1, for each $\mathcal{N}_2(\boldsymbol{\mu}, \Sigma)$ dataset to obtain a vector, $\mathbf{p}_1$, of p-values calculated under H_1.
4. Calculate the 10% Type I error rate, probability of rejecting H_0 at the 10% level when it is correct, using
$$T_I = \sum_{i=1}^{50} \mathbb{I}_{p_{0,i}<0.1}$$
5. Calculate the 10% Type II error rate, probability of accepting H_0 at the 10% level when it is incorrect, using
$$T_{II} = \sum_{i=1}^{50} \mathbb{I}_{p_{1,i}>0.1}$$



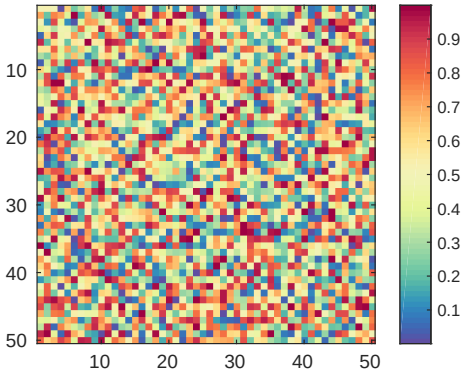
(a) GRF Data Correlations



(b) GRF Data Heatmap



(c) $\mathcal{U}(0,1)$ Data Correlations



(d) $\mathcal{U}(0,1)$ Data Heatmap

Figure 5.7: Neighbour correlations over distance, $\boldsymbol{\rho}$, for simulated data as described in §5.4.2. $\mathcal{N}_2(\boldsymbol{\mu}, \Sigma)$ and $\mathcal{U}(0,1)$ data, represented here as heatmaps, used for poorly and perfectly mixed data respectively as in §5.4. The structure of the Gaussian field has had a clear affect on the correlations over distance as close neighbours are correlated however over larger distances the covariance of the bivariate Gaussian distribution has less of an effect.

		Type I	Type II
F_{mink}		0.1	0
F_{GLCM}		0.02	0
Neighbour Correlation	F_{accept}	0.46	0.06
	F_{total}	0.22	0.02
	F_{diff}	0.08	0.12

Table 5.3: Type I and II error rates at a 10% level for our test statistics estimated by algorithm 5.4.5 using our simulated $\mathcal{U}(0,1)$ and $\mathcal{N}_2(\boldsymbol{\mu}, \Sigma)$ data, detailed in §5.4.3

Performing this for each of our test statistics gave us the results shown in table 5.3. We see from these results that both the Minkowski and GLCM based tests are extremely strict in their classification of random mixing. Both reject all GRF datasets and, in the case of the GLCM test, rarely reject a $\mathcal{U}(0,1)$ dataset in error. The neighbour correlation statistics produced type II error rates closer to our expectations, incorrectly accepting the null hypothesis between 2-12% of the time. The type I error rates however give us a clear indication of which test statistic to use. Both F_{accept} and F_{total} rejected our hypothesis of perfect mixing considerably more than the 10% level we desire. Our best performing test statistic is F_{diff} which gives us a type I error rate of 8%.

5.5 Applying the Test Statistics to Real Data

Having calculated the Type I and II error rates of our test statistics we now apply algorithm 5.4.1 to the bicalutamide data used in §5.3. The section we test is the inner 16 submatrices described in §5.3.1. We limit the test area to these submatrices as this excludes the tablet border with the background, a feature which would likely be detected as poor mixing by our test statistics. In figure 5.8 we show the region of the concentration heatmap to be tested and the results of the clustering and thresholding required for the Minkowski and GLCM based tests.

We applied algorithm 5.4.1 to our bicalutamide data, to each of 50 possible solutions identified by the random walk, and obtained the results shown in table 5.4. As we expected from the the Type II error rates, showing acceptance of only perfectly mixed data, they both reject the null hypothesis of a well mixed tablet 100% of the time. This is likely due to the levels in the image causing even minor shapes within the heatmap to be penalised alongside far more distinct homogeneous patches leading to a very strict test statistic.

Our final test statistics based on correlations allow us to conduct a test with no requirement

		Bicalutamide	
		Upper	Lower
F_{mink}		1	1
F_{GLCM}		1	1
Neighbour Correlation	F_{accept}	0.84	0.78
	F_{total}	0.24	0.16
	F_{diff}	0.36	0.24

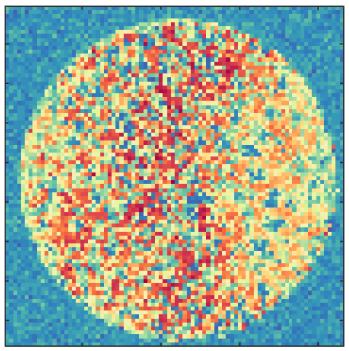
Table 5.4: Proportion of tests rejecting the null hypothesis of perfect mixing at a 10% level. To obtain these results algorithm 5.4.1 was applied to 50 randomly selected concentration matrices from the solution space of bilinear model 1.2.2 when applied to the bicalutamide data used in §5.3.1. Before testing the concentration matrices were cut down to only the inner 16 sections as described in §5.3.1 to exclude the tablet edge as the border between the tablet and the background would be detected by the test statistics as poor mixing.

for thresholding of the concentration, seen in figure 5.8d. The first two of these test statistics, F_{accept} and F_{total}, however performed poorly in type I error estimation, seen in table 5.3, and as discussed in §5.4.3 may cause problems in the bootstrapping due to their discrete nature. Without kernel smoothing incorporated into algorithm 5.4.1 the results obtained using either F_{accept} or F_{total} may be unreliable [41, 42, 68]. We are therefore left with F_{diff} as our most successfully performing test statistic when taking into consideration the nature of the data under testing, the performance in error rate estimation and structure we see in the heatmaps in figure 5.8 (bearing in mind these are only a single example of the results available in the solution space).

Comparing Amorphous and Standard Data

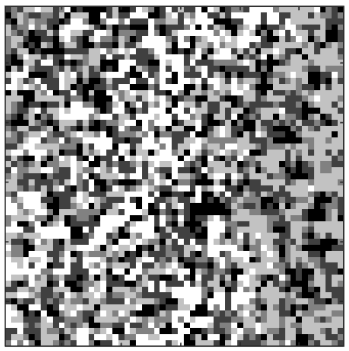
With our best performing test statistic identified, we return to our original question; do amorphous production methods produce a more thoroughly mixed product than standard crystalline methods? To test this we used our random walk, §4.6.1, to identify the solution spaces of six of our bicalutamide datasets, described in §1.2.2. All six scans were of the upper face of the tablets with three produced using the amorphous method and three using the standard crystalline method.

We applied algorithm 5.4.1 to each of our 6 datasets using the F_{diff} test statistic and $N = 50$, 50 locations on the solution space. We selected the concentrations corresponding to the bicalutamide spectra and created submatrices with (5.1.6) corresponding to the inner 16 sections described in §5.3.1. Our results show a clear improvement in the quality of

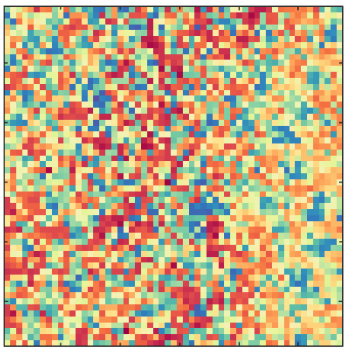


(a) Full upper face heatmap

(b) Upper face inner submatrix converted to binary with EMGM



(c) Upper face inner submatrix with 5 levels



(d) Upper face inner submatrix

Figure 5.8: Examples of the bicalutamide concentrations being tested for well-mixedness. All are calculated from a vector, $\mathbf{v}_k$, of concentrations corresponding to a single spectra as defined in (5.1.5). This is then reshaped into an 86×86 matrix, with the inner submatrices cut down to 57×57 as defined in §5.1.2.

	Amorphous			Crystalline		
	1	2	3	1	2	3
F_{diff}	0.2	0.24	0.22	0.08	0.14	0.12

Table 5.5: Proportion of tests rejecting the null hypothesis of perfect mixing at a 10% level. To obtain these results algorithm 5.4.1 was applied to 50 randomly selected concentration matrices from the solution space of bilinear model 1.2.2 when applied to all our bicalutamide datasets as described in §1.2.2. Before testing the concentration matrices were cut down to only the inner 16 sections as described in §5.3.1 to exclude the tablet edge as the border between the tablet and the background would be detected by the test statistics as poor mixing.

mixing in the crystalline production method, with an average of 11.3% of all tests rejecting the null hypothesis of perfect mixing at a 10% level in comparison to 22% for the amorphous method.

5.6 Discussion

Throughout this chapter we have investigated two key areas in the analysis of Raman spectroscopy data. To test the quantities of component chemicals in Raman images we applied Hotelling's T^2 test, using tabled values and a bootstrap test, and then incorporated into this our random walk, detailed in §4.6.1, to account for non-identifiability in the bilinear model 5.1.2. In testing how randomly mixed the component chemicals were we began with two existing image analysis techniques for describing features of an image. From these we created our own test statistic using linear discriminant analysis and simulated data. We also derived our own test statistic based on Pearson's correlation coefficient. These were then incorporated into a permutation test and combined with our random walk and applied to simulated and real datasets.

In implementing the classical T^2 test and the bootstrap test our conclusions were the same, with the results from the bootstrap test even showing that for our data the classical test assumptions held, or our sample was sufficiently large for an asymptotic distribution. However we had not accounted for the range of solutions our concentrations, $\mathbf{C}$, could take due to the non-identifiability of (1.2.2). Whilst we could perform a similar test with no issues of a non-identifiable solution by using principal component scores in place of concentrations, a relationship used in chapter 4 to define the solution space, we could not physically interpret the principal component scores as we could concentrations from (1.2.2).

We therefore combined our T^2 bootstrap test with our random walk of the solution space, algorithm 4.6.3. In using this bootstrap test on a variety of solutions we obtained similar results to those from the standard bootstrap and classical tests however we also saw how using only a single solution to (1.2.2) can give opposing p-values to our T^2 test, most clearly seen in figure 5.5b.

Our second area of interest in this chapter was testing how well mixed the concentrations are in a Raman image. Having formed an algorithm incorporating our random walk and permutation test we required a test statistic capable of extracting and quantifying the uniformity of mixing in the pixels of an image. The feature extraction methods we considered began with the Minkowski functionals, a set of three functions for quantifying shapes in a binary image. We therefore had to convert our images to binary for which we used expectation maximisation of a Gaussian mixture model clustering. The second method used eight texture properties of a grey-level co-occurrence matrix. This gave us a potential improvement over the Minkowski functionals as it allowed for multiple levels of grey in the image and so incorporated more of the continuous information in the concentrations. The final method we considered involved Pearson's correlation coefficient which we applied to the original concentrations with no need for clustering or thresholding. For this method we calculated the correlation between columns of the image with an increasing distance between them, giving us a vector of correlations as seen in figure 5.7. From this vector we defined our three summary statistics, detailed in §5.4.2, two of which use the known Student's t-distribution of the correlation coefficient.

With our three feature extraction methods defined we created test statistics. For the Minkowski functional and GLCM property based methods we used a linear discriminant analysis to select the best combination of functionals and properties for distinguishing between perfectly mixed $\mathcal{U}(0,1)$ datasets and poorly mixed $\mathcal{N}_2(\boldsymbol{\mu}, \Sigma)$ datasets. The final feature extraction methods using Pearson's correlation coefficient already returned values we could use as a test statistic in our algorithm. Initial investigation of our test statistics involved the $\mathcal{U}(0,1)$ and $\mathcal{N}_2(\boldsymbol{\mu}, \Sigma)$ data used previously. From this we estimated the Type I and II error values, where the Minkowski functional and GLCM based approaches had very low Type II error which we attributed to their penalisation of even minor homogeneous areas of the image as if they were clear shapes. Our neighbour correlation based approaches returned Type II error rates between 2-12%, indicating a less strict test statistic yet a low false negative range. However only one performed at the 10% Type I level we required. Our final investigation involved applying the test to Raman spectroscopy data. Here the Minkowski functional and GLCM based methods proved to be as strict in detecting structure in the images as we anticipated. We finally applied the best performing test statistic, F_{diff}

(5.4.17), to data which allowed us to test our initial question of whether amorphous or crystalline production methods produce a better mixed product. The results from this led us to conclude that, for our data at least, the crystalline method produced a more thoroughly mixed tablet. Future work in this area may include an investigation into the use of ellipsoid shape analysis methods where ellipsoids are repeatedly placed at random locations in an image and expanded until they include pixels too dissimilar from those already contained. The size of these ellipsoids provides an indication of the number of structures within the image.

In comparison to the current works of Coënt et al. [21], Koc et al. [58], Lee and Lin [65] our method has several benefits. Two of the previous methods [21, 58] involve counting the number of pixels per grey level. Not only does this require a greyscale image like our GLCM based method but it may miss a large group of pixels comprised of a single chemical component if the surrounding pixels are the other components to even out the quantities. Our better performing test statistic, F_{diff}, does not require greyscale data and would easily locate a region displaying a group of homogeneous pixels. The third method using spectroscopy [65] compares more closely with ours however uses the estimated spectra to judge the quality of mixing. Whilst comparing the spectral peaks between sections of the product will indicate whether each component is present it says nothing about the spread of the component chemicals. For example in a two component product each component could be present but inspection of the estimated concentrations may reveal the components are in two large separate clusters, a feature our method using neighbour correlations would detect.

Chapter 6

Analysis of a Spectral Concentrations through Time

6.1 Objectives of Change Point Analysis

Raman images can be captured over time by leaving the scan subject on the scan bed, whilst it dissolves for example, and scanning at set intervals. Sequential scans are a particularly useful area to the pharmaceutical industry as they allow us to perform a detailed analysis of the dissolution of a tablet over time, an area discussed by Strachan et al. [93]. This is a vital aspect of the industry, discussed by Bai and Wang [8], as the dissolution of a tablet has a major effect on the absorption of the chemicals by the body. The current method of post-production dissolution testing involves dissolving a tablet at body temperature, 37°C, and monitoring the spread of the tablet through a fixed quantity of liquid. This method however is susceptible to significant error and test failures [8].

Uneven dissolution of a tablet will have a negative effect on the efficacy of the tablet. A potential cause of uneven dissolution may be poor mixing causing a faster dissolving component to be prominent in one area. Whilst we may be able to pre-empt this cause of uneven dissolution by examining the uniformity of mixing in a tablet, a topic we considered in the previous chapter, this is only an examination of the tablet surface. We can therefore use sequentially gathered Raman images to conduct a more detailed investigation into how a tablet dissolves. We can perform this investigation by examining the concentrations of component chemicals at individual pixels as they increase or decrease over a sequence of time points. We therefore ask ourselves, can we accurately fit a model to these dissolution patterns? With a suitable model we go on to ask, can this model be used to display continuous component chemical levels throughout the dissolution process in place of the discrete scan times?

Using spectroscopy to image subjects changing over time is a relatively new field of research with very little involving Raman spectroscopy. Chan and Kazarian [18] used Fourier transform infra-red spectroscopy (FT-IR) to analyse the flow of different chemicals. The data used however was still in the form of a single static spectroscopic image through which the chemical was flowing. FT-IR was also used by Kazarian and van der Weerd [56] to gather images of the same form as our caffeine and bicalutamide datasets, §1.2.2, where a dissolving tablet was scanned at set time intervals. Their analysis of this data involved using least squares solutions to (1.2.2) and mapping the edge of the dissolving tablet over time. Finally Raman spectroscopy has been used in time analysis by [100, 101] where the data is in the form to be used in this chapter. In [100] investigation of the dissolution was simply performed by visually examining and comparing heatmaps of the estimated concentrations for two datasets corresponding to two production methods.

It is in [101] where the analysis begins to resemble what we propose in this chapter as to perform their analysis Windbergs et al. [101] initially use heatmaps of estimated concentrations estimated using the same methods we use in this thesis. Windbergs et al. [101] also use scanning electron micrographs of the dissolving tablet to visually describe the dissolution process. They then plot the total concentrations of the component chemicals and their changes over time. There has been no published research yet involving fitting models to the estimated concentrations of time related Raman images. It is in the model fitting that our methods differ as we use estimates of the times at which pixels of the Raman image change component chemical to map the dissolution rather than the total concentrations.

6.2 Sequential Scan Data

Dissolution scans are performed using the same process as a standard scan at discrete times for an image with n pixels over p wavenumbers. However, where before the $n \times p$ data matrix $\mathbf{Y}$ was of the form

$$\mathbf{Y} = \begin{pmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_n^T \end{pmatrix}, \tag{6.2.1}$$

a data matrix for m Raman images captured over time is of the form

$$\overset{nm\times p}{\mathbf{Y}} = \begin{pmatrix} \mathbf{Y}^{(1)} \\ \vdots \\ \mathbf{Y}^{(m)} \end{pmatrix}, \tag{6.2.2}$$

where each $\mathbf{Y}^{(l)}$ is of the form (6.2.1). We see this is simply m stacked Raman images. We also analyse this large data matrix by again finding solutions to

$$\mathbf{Y} - \mathbf{C}\mathbf{S}^T \tag{1.2.2}$$

by minimising

$$\underset{\mathrm{S,C}}{\operatorname{argmin}} \left\| \mathbf{Y} - \mathbf{C}\mathbf{S}^T \right\|$$

subject to the same constraints as before for a single image, namely

$$\begin{array}{lll} 1. & \mathbf{s}_k^T \mathbf{s}_k = 1 & \text{for all } k = 1, ..., r \\ 2. & s_{j,k} \geq 0 & \text{for all } j = 1, ..., p \text{ and } k = 1, ..., r \\ 3. & c_{i,k} \geq 0 & \text{for all } i = 1, ..., n \text{ and } k = 1, ..., r \end{array} \tag{1.2.5}$$

As p and r are the same for each of our Raman images in the sequence, in using (1.2.2) we get a single matrix of spectra, $\mathbf{S} = [\mathbf{s}_1 \ ... \ \mathbf{s}_r]$, for all of our images. However we now get an $nm \times r$ concentration matrix

$$\overset{nm \times r}{\mathbf{C}} = \begin{pmatrix} \mathbf{C}^{(1)} \\ \vdots \\ \mathbf{C}^{(m)} \end{pmatrix}. \tag{6.2.3}$$

where each $\mathbf{C}^{(l)}$ is a standard concentration matrix

$$\mathbf{C}^{(l)} = \begin{bmatrix} \mathbf{c}_1^{(l)T} \\ \vdots \\ \mathbf{c}_n^{(l)T} \end{bmatrix}$$

of the form described in §1.2.

From this matrix of concentrations, $\mathbf{C}$, we can perform one of the following two transformations, with these processes shown in figure 6.1.

1. Separate $nm \times r$ matrix, $\mathbf{C}$, into m matrices, $\mathbf{C}^{(l)}$, of dimension $n \times r$. These contain the concentrations of the component spectra at each time point l.

2. To obtain a heatmap of chemical k at time l,

   (a) Select the required submatrix, $\mathbf{C}^{(l)}$.

   (b) Select the k^{th} column of $\mathbf{C}^{(l)}$,

$$\mathbf{c}_k^{(l)} = \begin{bmatrix} c_{1,k}^{(l)} \\ \vdots \\ c_{n,k}^{(l)} \end{bmatrix}.$$

(c) Reshape column $\mathbf{c}_k^{(l)}$ into a matrix with the dimensions of the original Raman image, say $x \times y$, giving

$$\mathbf{P} = \begin{bmatrix} c_{1,k}^{(l)} & \cdots & c_{(y-1)x+1,k}^{(l)} \\ \vdots & \ddots & \vdots \\ c_{x,k}^{(l)} & \cdots & c_{n,k}^{(l)} \end{bmatrix} \tag{6.2.4}$$

from which we can create a heatmap of the concentrations.

3. To obtain the concentrations at each pixel of a given chemical at all time points,

(a) For chemical of interest, k, select column k of full concentration matrix $\mathbf{C}$

$$\mathbf{c}_k = \begin{bmatrix} c_{1,k} \\ \vdots \\ c_{mn,k} \end{bmatrix}.$$

(b) Reshape $\mathbf{c}_k$ into matrix

$$\underset{nr \times m}{\Delta_k} = \begin{bmatrix} c_{1,k} & \cdots & c_{nm-n,k} \\ \vdots & \ddots & \vdots \\ c_{n,k} & \cdots & c_{nm,k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\delta}_{1,k}^T \\ \vdots \\ \boldsymbol{\delta}_{n,k}^T \end{bmatrix} \tag{6.2.5}$$

where the $\boldsymbol{\delta}_{i,k}$ are the concentrations at pixel i of chemical component k over all m time points.

(c) Plot $\boldsymbol{\delta}_{i,k}$ against the m time points to see how the concentrations for chemical k change at pixel i. Figure 6.1 shows an example of Raman scans of a tablet with decreasing concentrations over time.

6.2.1 Sequential Scan Datasets

For the investigations conducted in this chapter we will use two datasets as they demonstrate two levels of complexity in dissolution patterns. The first is a caffeine tablet dissolving in water. These Raman images were collected over 211 minutes with nine individual scans. During this time the caffeine combines with the water to form caffeine hydrate, giving us one chemical component with decreasing concentration levels and a second component with increasing levels. Our second dataset is a bicalutamide tablet dissolving in water. This dataset is the more complicated of the two with the tablet extrudate, the initial substance, converting to bicalutamide type I either directly or first becoming bicalutamide type II, as shown in figure 6.2. Therefore this gives us a chemical component with decreasing concentrations, a component with increasing concentrations and a third that first increases

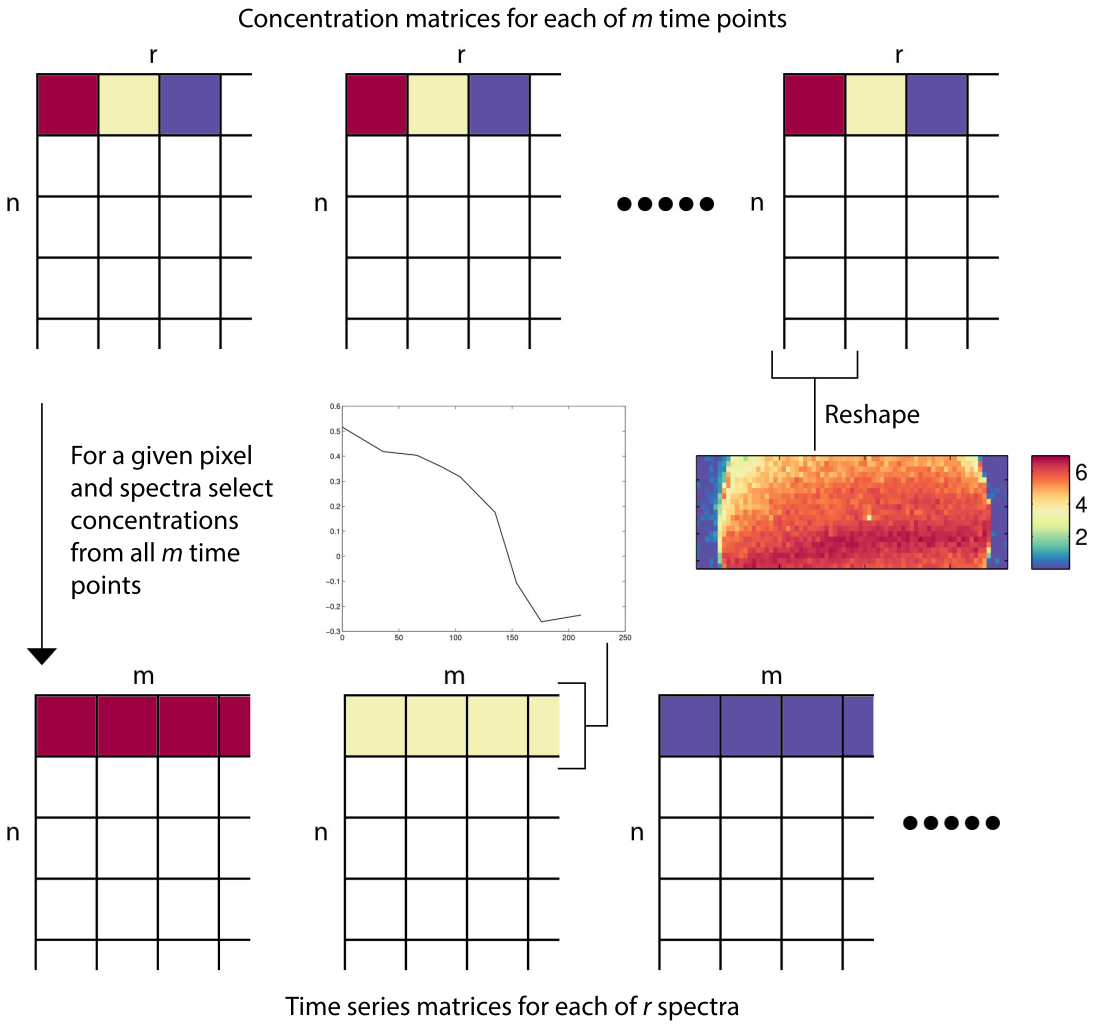


Figure 6.1: Transformation processes for a matrix of concentrations, **C**, of the form (6.2.3). Matrices used for time series analysis of the form (6.2.5). Matrices used to obtain a heatmap of the form (6.2.4).

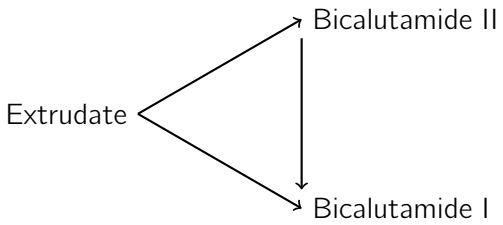


Figure 6.2: Bicalutamide dissolution process

then decreases in concentration. The scan was conducted over 2131 minutes with 14 individual scans. Throughout the scan the dissolving tablet also disperses outside of the scan area. As this may cause unusual behaviour in the concentrations of the component chemicals, for example unexpected increases or decreases as the solution disperses, we omit the final two scans resulting in a total scan time of 1597 minutes over 12 Raman images.

6.3 Initial Investigations of Sequential Scan Data

Without prior knowledge of the dissolution pattern of a dataset some initial investigations would be required. The most simple of these is simply looking at the estimated concentrations from (1.2.2) in the form of heatmaps generated with (6.2.4), a process described in §6.2. Applying this to our caffeine data we obtain the heatmaps shown in figure 6.3. These show the transition of the scanned object from caffeine to caffeine hydrate as we expect. Using this method with our bicalutamide data however gives us heatmaps, shown in figure 6.4, which are far less clear due to the more complicated dissolution process. Whilst the extrudate and bicalutamide type I concentrations shown in figure 6.4 show clear decreasing and increasing patterns respectively the bicalutamide type II concentrations are not so obvious.

Therefore for use with more complicated data we require an alternate method for identifying the dissolution process. A simple intuitive method for describing the dissolution is to use our r heatmaps of the scan subject and derive a single heatmap showing the location of each component chemical at each time point. A simple method for doing this involves taking each $\mathbf{C}^{(l)}$ from (6.2.3) and calculating [31, 50]

$$\mathbf{b}^{(l)} = \begin{bmatrix} \left\{k | \max_k c_{1,k}^{(l)}\right\} \\ \left\{k | \max_k c_{2,k}^{(l)}\right\} \\ \vdots \\ \left\{k | \max_k c_{n,k}^{(l)}\right\} \end{bmatrix}$$

an r-length vector containing the index, k, of the chemical with the highest concentration

at every pixel. We then calculate

$$\mathbf{d}^{(l)} = \begin{bmatrix} \sum_{i=1}^{n} \mathbb{I}\left\{b_i^{(l)} = 1\right\} \\ \sum_{i=1}^{n} \mathbb{I}\left\{b_i^{(l)} = 2\right\} \\ \vdots \\ \sum_{i=1}^{n} \mathbb{I}\left\{b_i^{(l)} = r\right\} \end{bmatrix}.$$

a vector containing the number of pixels at which each chemical component is the maximum concentration. Finally we create a matrix combining the $\mathbf{d}^{(l)}$ for all time points $l = 1, ..., m$,

$$\mathbf{D} = \left[\mathbf{d}^{(1)}\ \mathbf{d}^{(2)}\ ...\ \mathbf{d}^{(m)}\right]. \tag{6.3.1}$$

In figure 6.5 we show a bar chart of (6.3.1) for the bicalutamide data. This bar chart is created such that each $\mathbf{d}^{(l)}$ corresponds to a bar which is split into r sections, the areas of which are given by the $\sum_{i=1}^{n} \mathbb{I}\left\{b_i^{(l)} = k\right\}$. Whilst this plot shows the dissolution process to some extent, with the extrudate levels dropping and type I and II bicalutamide becoming prevalent, the exact process is still not clear. We also see an unusual pattern in the bicalutamide type I concentration where levels appear to increase, decrease then increase again rather than simple increasing as we expect. This may be caused by the rudimentary way in which we calculated (6.3.1) where we simply ignored the concentration levels relating to the other spectra.

With heatmaps and the simple calculation (6.3.1) based on the concentration levels providing vague indications of the dissolution patterns we turn to more detailed analysis by applying principal component analysis, §2.3, to (6.2.5) to estimate the patterns. First, using (6.2.5), we define

$$\overset{nr\times m}{\Delta} = \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{\delta}_{1,1}^T \\ \vdots \\ \boldsymbol{\delta}_{n,1}^T \\ \boldsymbol{\delta}_{1,2}^T \\ \vdots \\ \boldsymbol{\delta}_{n,2}^T \\ \vdots \\ \boldsymbol{\delta}_{1,r}^T \\ \vdots \\ \boldsymbol{\delta}_{n,r}^T \end{bmatrix} \tag{6.3.2}$$

Where matrix Δ contains the concentrations over all m time points for all pixels and chemical components. Using eigendecomposition we now calculate

$$\Delta^T\Delta = \mathbf{KLK}^T \tag{6.3.3}$$

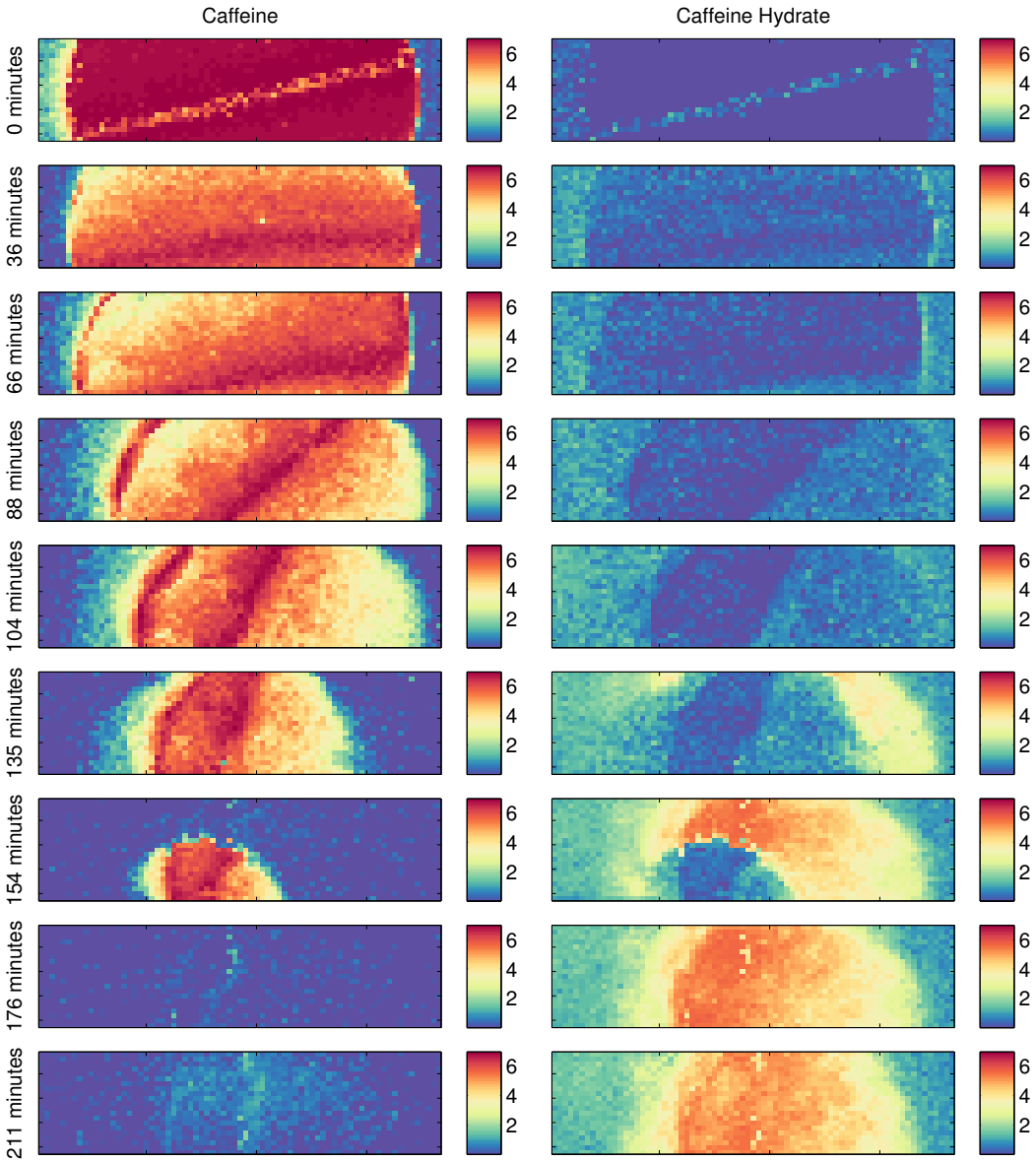


Figure 6.3: Estimated concentrations of caffeine and caffeine hydrate over the time of the scan, calculated with (1.2.2). Image generated using step 2 of the process described in §6.2

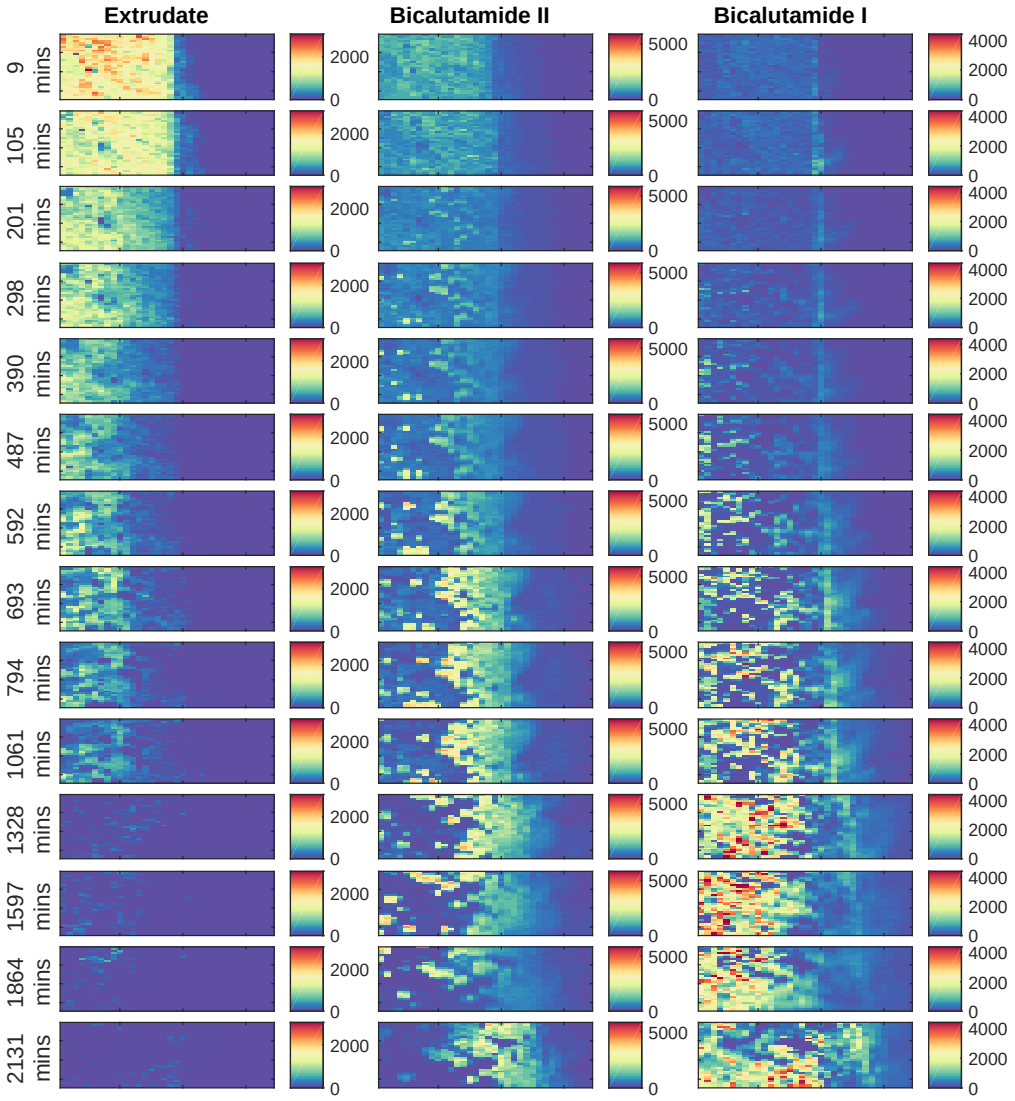


Figure 6.4: Estimated concentrations of the extrudate, bicalutamide I and bicalutamide II over the time of the scan, calculated with (1.2.2). Image generated using step 2 of the process described in §6.2

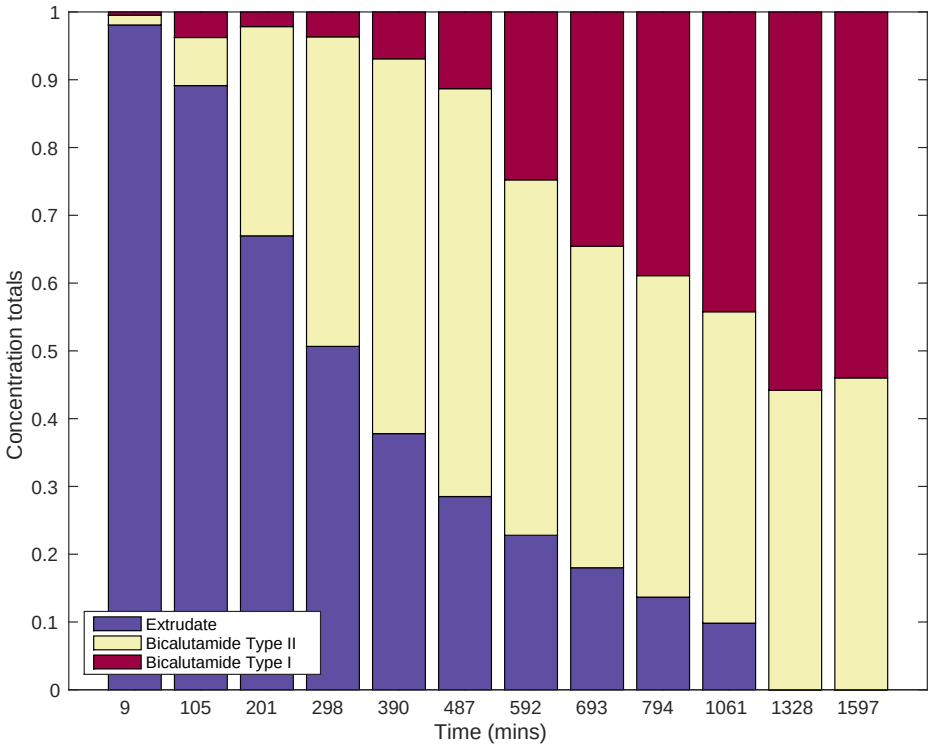
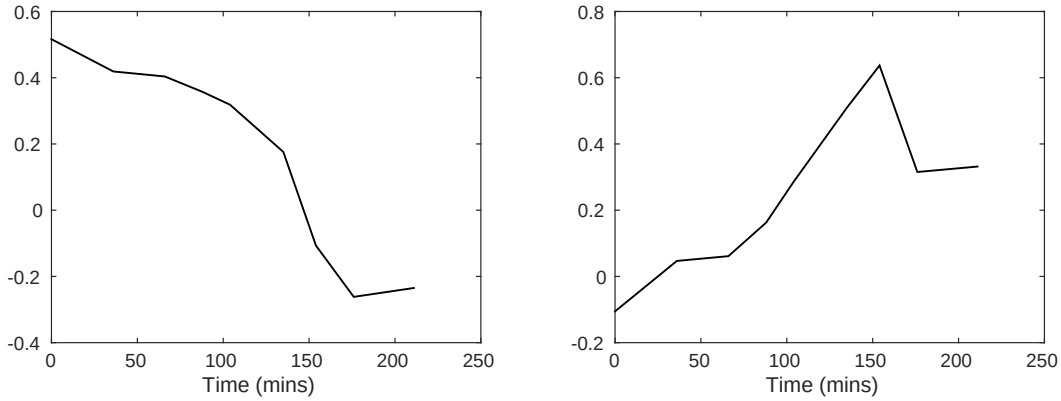


Figure 6.5: Total number of pixels containing each component at every time point. Pixel primary component chosen based on highest concentration value as in (6.3.1). The known change patterns are not clear here as bicalutamide type II does not display the increase then decrease we expect.



(a) Principal component 1 **(b)** Principal component 2

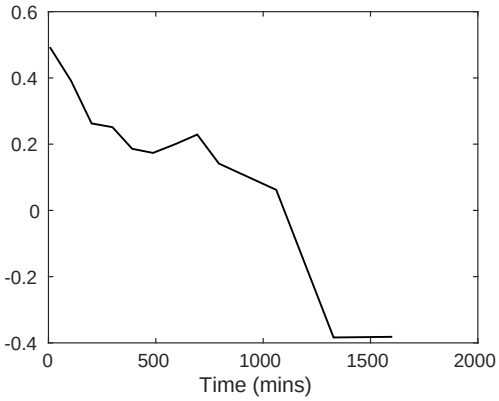
Figure 6.6: Principal component loadings for concentration changes in the caffeine data, calculated with (6.3.3), interpreted as caffeine and caffeine hydrate respectively.

where $\mathbf{K} = [\mathbf{k}_1 \ ... \ \mathbf{k}_m]$ are the orthonormal eigenvectors. Now as each $\boldsymbol{\delta}_{i,k}$, for $i = 1, ..., n$, in (6.3.2) should show, to some extent, the change in the levels of chemical k throughout the subject during dissolution we expect each Δ_k to have some common basis, $\tilde{\boldsymbol{\delta}}_k$, such that

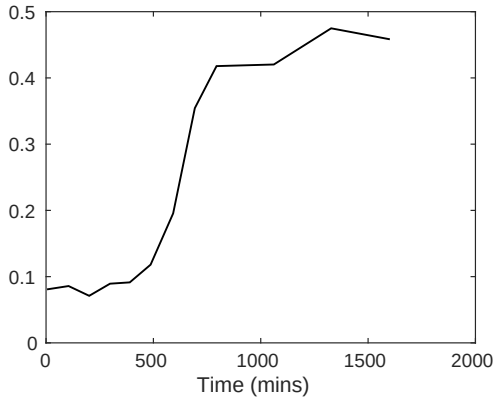
$$\boldsymbol{\delta}_{i,k} \approx g_i \tilde{\boldsymbol{\delta}}_k \quad \text{for } i = 1, ..., n$$

for some constants g_i. We should therefore see these bases, $\tilde{\boldsymbol{\delta}}_k$ for $k = 1, ..., r$, in the eigenvectors of (6.3.3), $\mathbf{k}_1, ..., \mathbf{k}_r$, corresponding to the r largest eigenvalues.

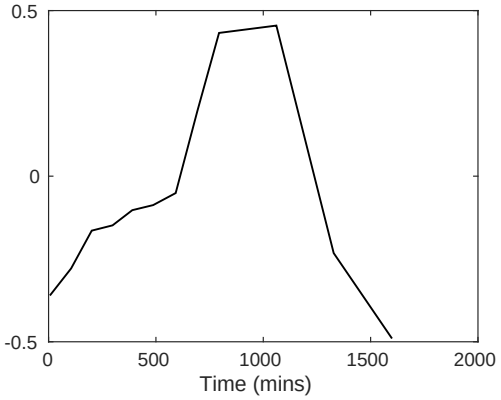
Applying (6.3.3) to our caffeine and bicalutamide datasets produces the $\mathbf{k}_1, ..., \mathbf{k}_r$ we see in figures 6.6 and 6.7 respectively where they are plotted against the times of the m scans. In both sets of figures our supposition that each Δ_k has some common basis function is shown to be sensible. In figure 6.6a and 6.6b we see the expected principal components indicating one chemical decreasing over time, caffeine, and the second increasing, caffeine hydrate. The deviations from this pattern at around 175 minutes are likely caused by the dissolving tablet dispersing across the spectrometer scan bed. The principal components derived from the concentrations of the bicalutamide data in figure 6.7 show the more complicated dissolution pattern we expect. We have principal components corresponding to an increasing concentration, a decreasing concentration and a final concentration which first increases then decreases. These will relate to the tablet extrudate, bicalutamide type I and bicalutamide type II respectively as described in figure 6.2. With both of our datasets this method proves to be simple yet effective at describing the patterns of change in the concentrations, allowing us to select the appropriate model to fit the data.



(a) Principal component 1



(b) Principal component 2



(c) Principal component 3

Figure 6.7: Principal component loadings for concentration changes in the bicalutamide data, calculated with (6.3.3), interpreted as the tablet extrudate, bicalutamide type I and bicalutamide type II respectively.

6.4 Basic Concentration Change Model

With a method of finding the concentration change patterns, using the eigenvectors of (6.3.3), we now require a technique for mapping the chemical changes of scan subject in an easily interpretable format. With the simple summation method of (6.3.1) giving less than clear results for our bicalutamide data we propose a model which will allow us to estimate the changes in chemical levels by incorporating previous and future time points.

The first model we propose is for a dataset containing one chemical with decreasing levels and another with increasing levels. Our caffeine data is an example of this type of data and, in figure 6.6, we see the change pattern for each chemical component to have a similar shape to that of a normal cumulative distribution function (CDF). We therefore require a model allowing us to fit the normal CDF to either an "S" or reverse "S" shape with a maximum greater than 1. For this purpose we define the model

$$\boldsymbol{\delta}_{i,k} = \begin{cases} \alpha_i \left(1 - \Phi\left(\mu_i, \sigma_i^2 | \mathbf{t}\right)\right) & \text{if } \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\wedge}\| < \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\vee}\| \\ \alpha_i \left(\Phi\left(\mu_i, \sigma_i^2 | \mathbf{t}\right)\right) & \text{if } \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\wedge}\| > \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\vee}\| \end{cases} \tag{6.4.1}$$

for Euclidean norm $\|\cdot\|$, $\boldsymbol{\delta}_{i,k}$ a row of the matrix defined in (6.2.5) and $\mathbf{t}$ a vector containing the times at which the object was scanned. $\Phi\left(\mu_i, \sigma_i^2 | \mathbf{t}\right)$ is the normal cumulative distribution function with mean μ_i and variance σ_i^2 evaluated at t_l for $l = 1, ..., m$. Vectors $\mathbf{k}^{\wedge}$ and $\mathbf{k}^{\vee}$ are calculated using (6.3.3) and represent the decreasing and increasing patterns respectively. The condition on the eigenvectors in (6.4.1) compare the series of concentrations, $\boldsymbol{\delta}_{i,k}$, to the eigenvectors to identify the correct orientation of the normal CDF. The vectors $\boldsymbol{\alpha}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ contain parameter estimates for each pixel, i, with the former a scaling constant set to allow the function to achieve a maximum different from one. We fit this model to each pixel by minimising the residual sum of squares

$$r_i = \begin{cases} \left\|\boldsymbol{\delta}_{i,k} - \alpha_i \left(1 - \Phi\left(\mu_i, \sigma_i^2 | \mathbf{t}\right)\right)\right\|^2 & \text{if } \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\wedge}\| < \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\vee}\| \\ \left\|\boldsymbol{\delta}_{i,k} - \alpha_i \left(\Phi\left(\mu_i, \sigma_i^2 | \mathbf{t}\right)\right)\right\|^2 & \text{if } \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\wedge}\| > \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\vee}\| \end{cases} . \tag{6.4.2}$$

In our model we can interpret the mean, μ_i, as the time of greatest concentration change and the variance, σ_i^2, as the speed of the change.

Fitting this model to our caffeine dataset showed that for pixels with a clear point of change in concentrations this model provides us a good fit, as shown in figure 6.8. However if the time at which the concentration changes is less clear, for example figure 6.9a, we see the model fit is compromised. Large differences between the eigenvectors representing the change patterns and the concentration change vectors, $\boldsymbol{\delta}_{i,k}$, have a variety of causes. For example anomalies in the data such as cosmic rays, detailed in §3.2, may result in incorrectly elevated concentrations. The dispersion throughout the solvent of the dissolving tablet

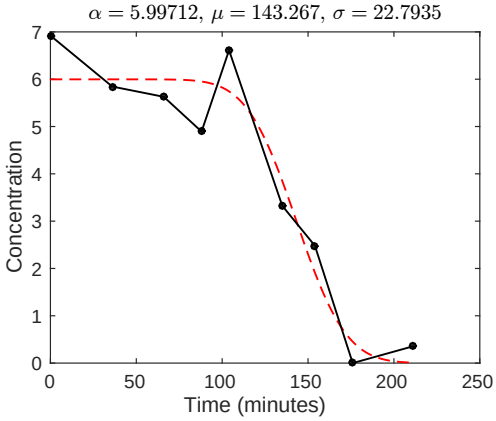
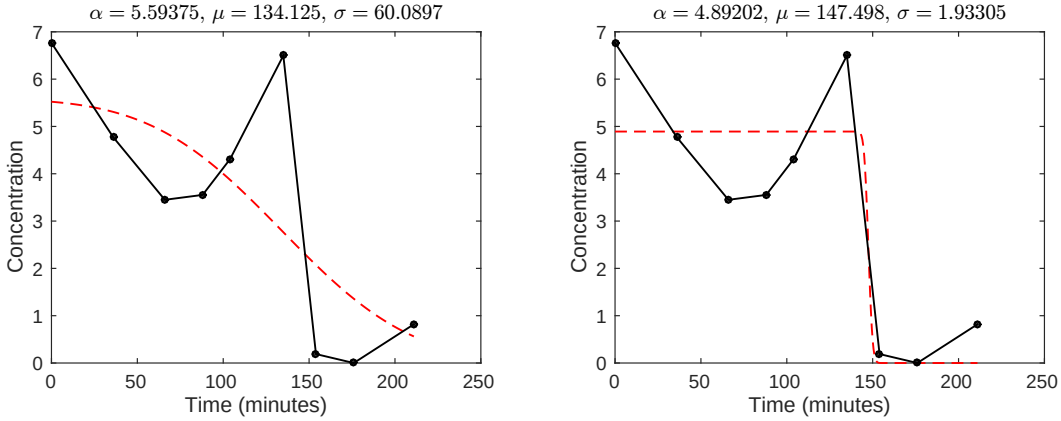


Figure 6.8: Caffeine levels at pixel 458 with model (6.4.1).



(a) Standard model, (6.4.1). **(b)** Penalty model, (6.4.3).

Figure 6.9: Pixel 1417 from the caffeine data with overlaid model fits.

across the scan bed could also cause an unusual increase or decrease in concentration levels. After fitting (6.4.1) to all n pixels we reshape our vector $\boldsymbol{\mu}$ corresponding to the change in caffeine, $k = 1$, into the Raman image dimensions using (6.2.4) to obtain the heatmap seen in figure 6.10a. This heatmap shows us the time at which each pixel undergoes the greatest change in concentration and clearly contains values of μ_i around the edges of the tablet where the objective function has been unsuccessfully fit to the $\boldsymbol{\delta}_{i,k}$.

Therefore we propose an improvement to the fit of our model by incorporating model parameters from the surrounding pixels. We include these neighbouring parameters a penalty term which will penalise deviation from the neighbouring parameter estimates. For this penalty term we use a similar method to that of the quadratic penalty method, appendix A.4. Whilst these types of penalty terms may cause a higher residual sum of squares for (6.4.1) they allow us to enforce constraints such as similar μ_i between pixels, reducing the occurrence of situations such as that seen in figure 6.9a. Using this penalty method we

(a) No penalty, (6.4.1).

(b) Penalty of 0.01 for each parameter, (6.4.3) with modified penalty term (6.4.5).

Figure 6.10: Heatmaps of $\boldsymbol{\mu}$ for the caffeine concentrations, $k = 1$, created with (6.2.4), estimated from the caffeine dataset with two levels of smoothing.

propose the objective functions

$$r_i' = \begin{cases} \left\|\boldsymbol{\delta}_{i,k} - \alpha_i \left(1 - \Phi\left(\mu_i, \sigma_i^2 | \mathbf{t}\right)\right)\right\|^2 + \\ \quad + \sum_{i' \in \mathcal{N}_i} \left[\gamma_1 \left(\frac{\alpha_i - \alpha_{i'}}{|\mathcal{N}_i|}\right)^2 + \gamma_2 \left(\frac{\mu_i - \mu_{i'}}{|\mathcal{N}_i|}\right)^2 + \gamma_3 \left(\frac{\sigma_i - \sigma_{i'}}{|\mathcal{N}_i|}\right)^2\right] \\ \qquad\qquad \text{if} \quad \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\wedge}\| < \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\vee}\| \\ \left\|\boldsymbol{\delta}_{i,k} - \alpha_i \left(\Phi\left(\mu_i, \sigma_i^2 | \mathbf{t}\right)\right)\right\|^2 + \\ \quad + \sum_{i' \in \mathcal{N}_i} \left[\gamma_1 \left(\frac{\alpha_i - \alpha_{i'}}{|\mathcal{N}_i|}\right)^2 + \gamma_2 \left(\frac{\mu_i - \mu_{i'}}{|\mathcal{N}_i|}\right)^2 + \gamma_3 \left(\frac{\sigma_i - \sigma_{i'}}{|\mathcal{N}_i|}\right)^2\right] \\ \qquad\qquad \text{if} \quad \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\wedge}\| > \|\boldsymbol{\delta}_{i,k} - \mathbf{k}^{\vee}\| \end{cases} \tag{6.4.3}$$

where the γ_g are constants to set the level of penalisation and $\mathcal{N}_i$ is the set of indices corresponding to pixels surrounding pixel i in the Raman image, defined for a Raman image of size $x \times y$ as

$$\begin{aligned} \mathcal{N}_i = [&(i - y - 1), (i - y), (i - y + 1), (i - 1), \ldots \\ &(i + 1), (i + y - 1), (i + y), (i + y + 1)] \end{aligned} \tag{6.4.4}$$

where we exclude some indices if the pixel of interest, i, lies on the edge of the Raman image. Objective function (6.4.3) clearly requires initial estimates of the parameters for all pixels so we apply this model using the following algorithm.

Algorithm 6.4.1. Iterative algorithm for modelling simple concentration changes by minimising (6.4.3)

1. Set starting parameter values, $\alpha_i = \max(\boldsymbol{\delta}_{i,k})$, σ_i, γ_g and $\mu = t_{\lfloor \frac{m}{2} \rfloor}$ where $\mathbf{t}$ is the vector of scan times. In practice we found $\sigma_i = 50$ and $\gamma_g = 0.001$ work well.

2. Minimise (6.4.2) using a method of numerical optimisation to obtain initial estimates of parameter vectors $\boldsymbol{\alpha}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. In practice we found the trust region method to work well (see appendix A.3).

3. Randomly select a pixel without replacement from $i = 1, ..., n$ and minimise (6.4.3) for α_i, μ_i and σ_i^2.

4. Replace original estimates from step 2 with the new estimates from step 3.

5. Repeat steps 3 and 4 n times until all parameters have been re-estimated with the penalty model.

We minimise (6.4.3) for pixels in a random order to reduce the risk of a series of parameter estimates being effected by poor model fits. If the algorithm were to work along columns or rows erroneous parameters could cause a sequence of errors where parameter estimates such as those in figure 6.9a could effect each successive model fit along the row or column.

Fitting (6.4.3) to our caffeine data improved the fit we saw in figure 6.9a to that shown in figure 6.9b. Our model here incorporates the relationship between neighbouring pixels to give a more spatially consistent result with a greatest concentration change time around 150 minutes.

For data displaying a decrease in one chemical level and an increase in the other chemical level it is highly likely these two will be linked as the component increasing in concentration must come from somewhere. With this in mind we extend the penalty term in (6.4.3) to include neighbouring pixel parameters estimated for the alternate spectra. The idea behind this modification is that for some $\boldsymbol{\delta}_{i,1}$ the time of greatest concentration level change should be similar to that of $\boldsymbol{\delta}_{i,2}$ as for one to increase the other must decrease. Therefore we modify the penalty term of (6.4.3) to

$$\sum_{i' \in \mathcal{N}_i} \left[\gamma_1 \left(\frac{\alpha_i - \alpha_{i'}}{|\mathcal{N}_i|} \right)^2 + \gamma_2 \left(\frac{\mu_i - \mu_{i'}}{|\mathcal{N}_i|} \right)^2 + \gamma_3 \left(\frac{\sigma_i - \sigma_{i'}}{|\mathcal{N}_i|} \right)^2 + \gamma_4 \left(\frac{\mu_i - \mu'_{i'}}{|\mathcal{N}_i|} \right)^2 \right] \quad (6.4.5)$$

where the $\mu'_{i'}$ are the neighbouring values of μ_i for the other spectra.

To test this new penalty term we applied algorithm 6.4.1 to our caffeine data with the penalty term of (6.4.3) changed to (6.4.5). After reshaping the estimated change times, $\boldsymbol{\mu}$, as before, with (6.2.4), we obtain the heatmap shown in figure 6.10b. We now have a clear representation of the caffeine tablet dissolving and beginning to spread across the scan bed.

6.5 Concentration Change Model for More Complicated Images

With our simple model proving successful at mapping the dissolution of our caffeine dataset, and providing an easily interpretable heatmap of the dissolution, we now propose a more universal model capable of fitting to more complicated dissolution patterns. As we saw in figure 6.7 the "S" shaped change patterns may feature alongside an "n" shaped pattern. Therefore the evolution of (6.4.1) we propose combines the normal CDFs to create a curve capable of fitting to data displaying decreasing, increasing and both increasing then decreasing concentrations. Our bicalutamide data is an example of this with the required change patterns as seen in figure 6.7. We propose the model

$$\boldsymbol{\delta}_{i,k} = f(\mathbf{t}|\alpha_i, \mu_{i,1}, \mu_{i,2}, \sigma_i^2) = \begin{cases} \alpha_i \left(\Phi\left(\mu_{i,1}, \sigma_i^2|t_l\right)\right) & \text{if } \ t_l \leq c \\ \alpha_i \left(1 - \Phi\left(\mu_{i,2}, \sigma_i^2|t_l\right)\right) & \text{if } \ t_l > c \end{cases} \tag{6.5.1}$$

where $\mu_1 < \mu_2$ and $c = \mu_1 + \frac{\mu_1+\mu_2}{2}$. The parameters α_i, μ_1, μ_2 and σ control the height, location and width of the curve respectively. The parameter σ is common to both normal distributions to avoid a discontinuity where the cumulative distribution functions meet. We show a simple example of this model in figure 6.11. We can fit this model to all three concentration patterns by adjusting c to translate the curve along the time axis, for example $c = t(1)$ gives us a model with the shape of decreasing concentrations.

Using the same quadratic penalty method as used for (6.4.3) we modify (6.5.1) to improve the fit with pixels displaying sporadic concentration changes. As with (6.4.3) we penalise large differences in change times, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, between pixel i and the immediate neighbours. We also induce a penalty on α_i for differing too greatly from $\max(\boldsymbol{\delta}_{i,k})$ along with a penalty on c for lying outside $\mathbf{t}$, this final penalty is to ensure at least one of $\mu_{i,1}$ or $\mu_{i,2}$ lies within $[t(1), t(m)]$. Our penalty on α_i has been changed from (6.4.5) to use $\max(\boldsymbol{\delta}_{i,k})$ in place of neighbouring pixel maxima as the presence of three chemical components with a more complicated dissolution pattern allows for a greater range of concentration levels across the image. With these modifications we define the objective function

$$\begin{aligned} r_i'' = \|\boldsymbol{\delta}_{i,k} - f(\mathbf{t}|\alpha_i, \mu_{i,1}, \mu_{i,2}, \sigma_i^2)\| + \gamma_1 \left[\alpha_i - \max(\boldsymbol{\delta}_{i,k})\right]^2 \\ + \sum_{i' \in \mathcal{N}_i} \left[\gamma_2 \left(\frac{\mu_{i,1} - \mu_{i',1}}{|\mathcal{N}_i|}\right)^2 + \gamma_3 \left(\frac{\mu_{i,2} - \mu_{i',2}}{|\mathcal{N}_i|}\right)^2\right] + \gamma_4 \mathbb{I}_{c<t_1, c>t_m} \end{aligned} \tag{6.5.2}$$

where the γ_g are constants to set the level of penalisation and $\mathcal{N}$ is the set of neighbouring pixels calculated as in (6.4.4).

To fit this model to all pixels of a Raman image we modify algorithm (6.4.1) to fit (6.5.1) to the concentration change patterns shown in figure 6.7.

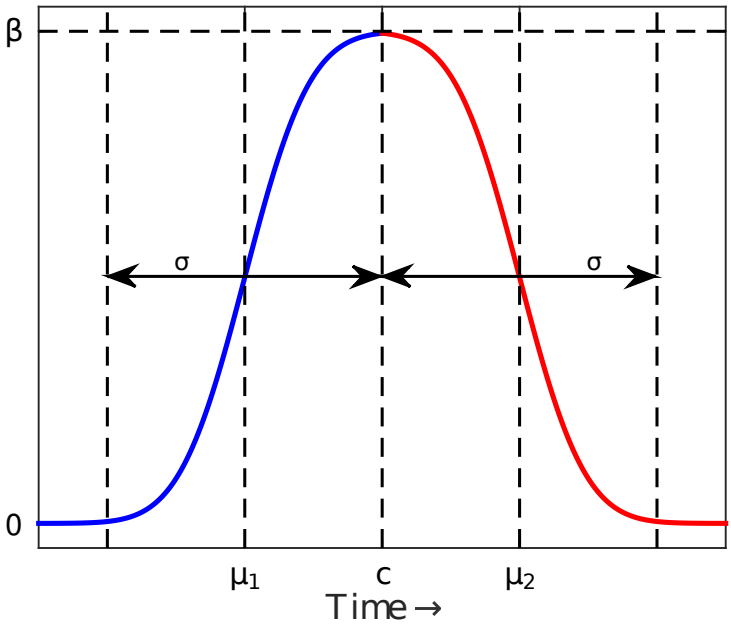


Figure 6.11: Proposed double normal CDF model to fit all three concentration change patterns displayed by the bicalutamide data, seen in figure 6.7. Specific shape generated by $\mu_1 = -4$, $\mu_2 = 4$ and $\sigma = 1.5$.

Algorithm 6.5.1. Iterative algorithm for minimising (6.5.2)

1. Estimate concentration change pattern for each set of concentrations by calculating $\|\boldsymbol{\delta}_{i,k} - \mathbf{k}_k\|$ for $k \in [1, 3]$, where the $\mathbf{k}_k$ are principal components derived with (6.3.3) and ordered such that they represent decreasing, increasing and increasing then decreasing concentrations respectively.

2. Set starting parameter values, $\alpha_i = \max(\boldsymbol{\delta}_{i,k})$, $\sigma_i^2 = \frac{t_m - t_1}{10}$ and γ_g. For $\mu_{i,1}$ and $\mu_{i,2}$ we use starting values such that the point(s) of greatest change lie close to where we expect for that change pattern, for example an increasing concentration will require $\mu_{i,1}$ close to the median time point and $\mu_{i,1}$ beyond the maximum time to ensure the correct part of the curve is used.

$$\{\mu_{i,1}, \mu_{i,2}\} = \begin{cases} \{t_1 - t_{\lfloor \frac{m}{2} \rfloor}, t_{\lfloor \frac{m}{2} \rfloor}\}, & \text{if the pattern is decreasing} \\ \{t_{\lfloor \frac{m}{2} \rfloor}, t_m + t_{\lfloor \frac{m}{2} \rfloor}\}, & \text{if the pattern is increasing} \\ \{t_{\lfloor \frac{m}{4} \rfloor}, t_{\lfloor \frac{3m}{4} \rfloor}\}, & \text{if the pattern is increasing then decreasing} \end{cases}$$

   In the increasing and decreasing cases we respectively set $\mu_{i,1}$ and $\mu_{i,2}$ outside the range of $\mathbf{t}$ to force only half of the model to be fit to our concentrations, $\boldsymbol{\delta}_{i,k}$. We also found $\gamma_g = 0.0001$ to perform well in practice.

3. Minimise (6.5.2) without the penalty term to obtain initial estimates of parameter

vectors $\boldsymbol{\alpha}$, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\sigma}$. As before we found the trust region method of numerical optimisation to perform well.

4. Randomly select a pixel without replacement from $i = 1, ..., n$ and minimise (6.5.2) for α_i, $\mu_{i,1}$, $\mu_{i,2}$ and σ_i^2.

5. Replace original estimates from step 3 with those from step 4.

6. Repeat steps 4 and 5 n times until all parameters have been re-estimated with the penalty model.

Applying this algorithm to our bicalutamide dataset we obtained vectors $\boldsymbol{\mu}_1^{(k)}$ and $\boldsymbol{\mu}_2^{(k)}$, for $k \in [1, 3]$, corresponding to each component chemical. To obtain the heatmaps shown in 6.12 we performed the following steps,

- For $k = 1$, decreasing concentrations, ignore $\boldsymbol{\mu}_1^{(k)}$ and set $\mu_{i,2}^{(k)} \notin [t(1), t(m)]$, for $i = 1, ..., n$, equal to zero.
- For $k = 2$, increasing concentrations, ignore $\boldsymbol{\mu}_2^{(k)}$ and set $\mu_{i,1}^{(k)} \notin [t(1), t(m)]$, for $i = 1, ..., n$, equal to zero.
- For $k = 3$, increasing then decreasing concentrations, set $\mu_{i,1}^{(k)} \notin [t(1), t(m)]$ and $\mu_{i,2}^{(k)} \notin [t(1), t(m)]$, for $i = 1, ..., n$, equal to zero.

We then reshape these modified $\boldsymbol{\mu}_1^{(k)}$ and $\boldsymbol{\mu}_2^{(k)}$ vectors as in (6.2.4). From these heatmaps we see the expected dissolution pattern

- Extrudate decreasing early on.
- Bicalutamide type I increasing throughout the scan.
- Bicalutamide type II increasing early and then decreasing towards the end.

We now also recreate the bar chart seen in figure 6.5 however using the results of our model in place of the chemical component with the maximum concentration value at each pixel. To recreate vectors, $\mathbf{d}^{(l)}$, used in (6.3.1) we calculate

$$\tilde{\mathbf{d}}^{(l)} = \begin{bmatrix} \sum_{i=1}^{n} \mathbb{I}\left\{\mu_{i,2}^{(1)} \geq t(l), \mu_{i,2}^{(1)} \neq 0\right\} \\ \sum_{i=1}^{n} \mathbb{I}\left\{\mu_{i,1}^{(2)} \leq t(l), \mu_{i,1}^{(2)} \neq 0\right\} \\ \sum_{i=1}^{n} \mathbb{I}\left\{\mu_{i,1}^{(3)} \leq t(l), \mu_{i,2}^{(3)} \geq t(l), \mu_{i,1}^{(3)} \neq 0, \mu_{i,2}^{(1)} \neq 0\right\} \end{bmatrix}. \tag{6.5.3}$$

The entries in this vector can be interpreted as

- The number of pixels yet to see a large decrease in extrudate concentrations by time $t(l)$.
- The number of pixels to have seen a large increase in bicalutamide type I concentrations by time $t(l)$.
- The number of pixels to have seen a large increase in bicalutamide type II concentrations but not yet seen a large decrease by time $t(l)$.

Now to create the matrix from which we plot our bar chart we define

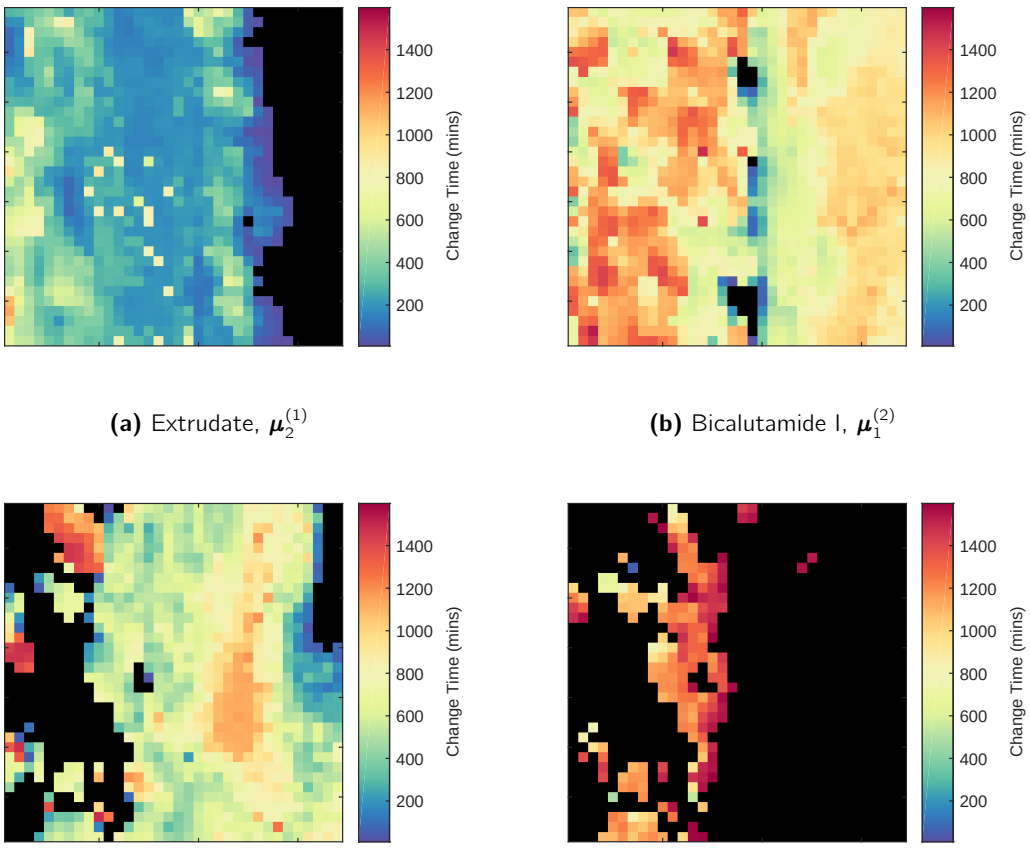
$$\tilde{\mathbf{D}} = \left[\tilde{\mathbf{d}}^{(1)}\ \tilde{\mathbf{d}}^{(2)}\ \ldots\ \tilde{\mathbf{d}}^{(m)}\right]. \tag{6.5.4}$$

As before in §6.3 we create the bar chart by taking each $\tilde{\mathbf{d}}^{(l)}$ as a bar which is split into 3 sections with the areas defined by (6.5.3). The bar chart we obtain from this is shown in figure 6.13. We now very clearly see the pattern of concentration change as bicalutamide type II increases then decreases with bicalutamide type I becoming the dominant spectra by the end of the scan. This is a great improvement over figure 6.5 where we could discern very little about the behaviour of bicalutamide type II.

6.6 Discussion

Our goal in this chapter has been to develop a model capable of accurately representing the dissolution of a tablet, a vital element in the effectiveness of a drug. We first demonstrated methods of preliminary investigation using principal component analysis and a simple visual inspection and count of the estimated concentrations, (6.3.1). Whilst the latter of these was only useful for simple datasets the principal component analysis based method proved effective at representing the patterns of change undergone by the chemical component concentrations over the course of the scan. For example with the bicalutamide data we saw a clear indication of decreasing, increasing and increasing then decreasing concentrations, figure 6.7.

We first defined a simple model to fit datasets with two chemical components where one increases whilst the other decreases. This led us to require an "S" shaped function such as the normal cumulative distribution function or the logistic function. Our chosen function to use was the normal CDF where the two standard parameters, μ and σ, were used to control the time and speed of the chemical change with a third, α, used to allow a maximum different from one. This model fitted well to simple chemical concentrations over time however our parameter estimates proved susceptible to outliers resulting in poor



(a) Extrudate, $\boldsymbol{\mu}_2^{(1)}$

(b) Bicalutamide I, $\boldsymbol{\mu}_1^{(2)}$

(c) Bicalutamide II Increase, $\boldsymbol{\mu}_1^{(3)}$

(d) Bicalutamide II Decrease, $\boldsymbol{\mu}_2^{(3)}$

Figure 6.12: Heatmaps of change times, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, for each chemical in the bicalutamide dataset. Results obtained using algorithm 6.5.1 and heatmaps created with (6.2.4). These show the dissolution patterns more clearly than figure 6.4 as we now have obvious times of increase then decrease for bicalutamide II.

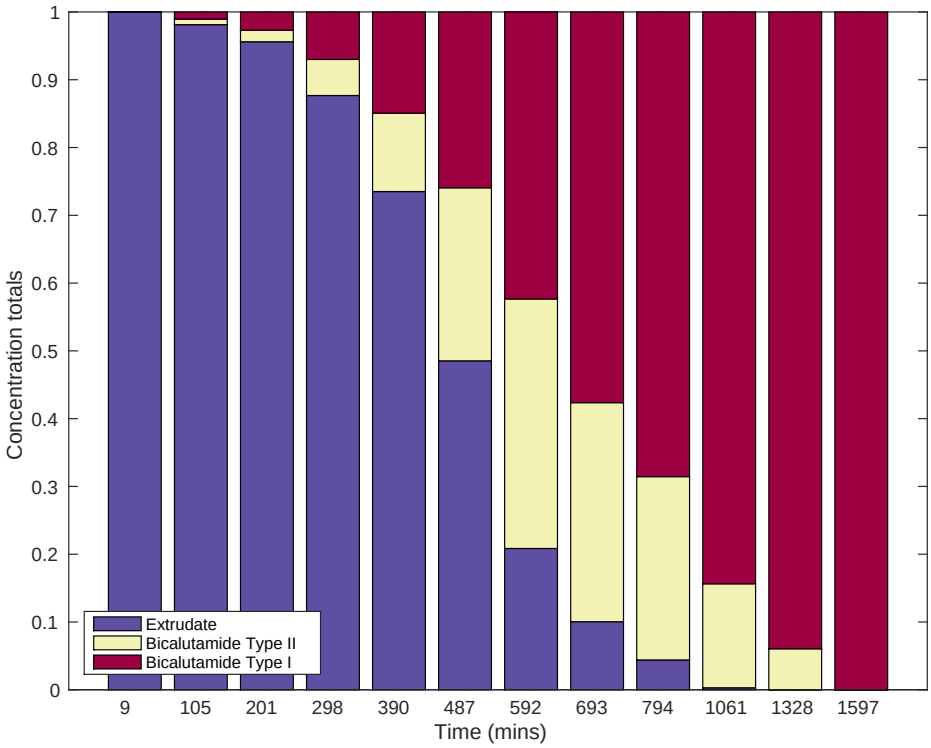


Figure 6.13: Total number of pixels containing each chemical component at every time point. Pixel primary component chosen based on the results from algorithm 6.5.1 as defined by (6.5.4). The known change patterns are now very clear as bicalutamide type II displays the expected increase then decrease.

model fit. To make our model more robust to anomalies we used a penalty term to smooth the model fit on pixels displaying uneven concentrations, figure 6.9. Due to the simplicity of a quadratic penalty method we modified our model to penalise large differences from the parameter estimates of neighbouring pixels in the Raman image. We modified the model further to exploit the pattern that in a dissolving subject a decrease in one chemical component would likely lead to an increase in the other. This model proved very effective when fit to our caffeine dataset and gave us a clear representation of the times at which areas of the Raman image changed between the discrete scan times.

With a model fit to the more simple data style we turned our focus to a more complicated model capable of mapping increasing, decreasing and both increasing and decreasing chemical concentrations. We were able to obtain a rough indication of the way a tablet of this structure dissolved by summing how many pixels at each time point were maximised by each chemical. However this method suffered as the concentrations of only one component were taken into consideration. When applied to our bicalutamide dataset this meant identifying that one chemical component increased then decreased was very difficult. To define our model for images displaying these change patterns we combined the two functions used previously to form a bell shaped curve capable of fitting all three concentrations. To incorporate robustness into this new model we used the same quadratic penalty method to constrain the parameters based on the estimates at neighbouring pixels. We also had to constrain the location of the peak of this curve to ensure temporally correct results as during fitting the function is translated along the axis representing time, therefore negative time estimates had to be avoided. These constraints gave us a reliably fitting model with the same resistance to inconsistent concentrations as we saw with the previous model. When applied to our bicalutamide dataset we recreated heatmaps and bar charts which now showed the pattern of dissolution accurately enough that no prior knowledge was required to interpret the results. The function used for this method could potentially be extended to other change patterns, for example a fluctuating chemical concentration requiring a sine curve shape which could easily be obtained by including a third normal CDF.

Our model demonstrates advantages over the current methods for analysis of spectroscopic images over time which involve mapping the edge of the subject [56], visual examination of heatmaps [100] and calculating the total concentrations of each component [101] estimated using the bilinear model (1.2.2). The model we propose incorporates all pixels in the images, not just the edge pixels of the dissolving subject. It also gives a single heatmap of the dissolution rather than requiring inspection of a heatmap per time point and, as we show in figure 6.13, provides clearer results than simply summing the concentrations.

Chapter 7

Analysis of Spectroscopy Data using Cluster Analysis and Mixture Models

7.1 Objectives of Analysis using Clusters

When capturing a Raman spectroscopic image it is common to scan a region larger than the subject to ensure we miss nothing important. This background however may have a detrimental effect on the results. This is because increasing the number of pixels increases the potential for anomalies, detailed in §3.2, yet in the case of background pixels, containing no component spectra, add nothing to estimated concentrations and spectra from

$$\mathbf{Y} = \mathbf{C}\mathbf{S}^T + \mathbf{E} \tag{1.2.2}$$

my minimising

$$\underset{\mathrm{S,C}}{\operatorname{argmin}} \left\| \mathbf{Y} - \mathbf{C}\mathbf{S}^T \right\|.$$

Therefore we wish to devise a method for solving (1.2.2) on the background and the subject separately, the motivvation being that this will in theory give us clearer estimated spectra and concentrations. For performing this analysis we have a choice of methods. The first involves using cluster analysis such as k-means, §2.10, and EMGM, §2.12.1, to isolate the background from the subject then using a bipartite model to estimate separate spectra and concentrations for each group. The second method combines the bipartite model and Gaussian mixture model of EMGM with the expectation maximisation algorithm to both cluster and decompose the data. This combined technique may prove more accurate as each refinement step of the spectra and concentrations should allow for more accurate clustering of the Raman image pixels.

7.2 Fitting a Bipartite Bilinear Model

Before we can fit a bipartite model to our data we must first distinguish between the scan subject and the background which is visually very clear in our results from previous chapters such as figure 5.1. The most simple option is to sum our data,

$$\overset{n\times p}{\mathbf{Y}} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_n^T \end{bmatrix}$$

over all columns to obtain

$$\tilde{\mathbf{y}} = \begin{bmatrix} \sum_{j=1}^{p} y_{1,j} \\ \vdots \\ \sum_{j=1}^{p} y_{n,j} \end{bmatrix} \tag{7.2.1}$$

and threshold the resulting vector by selecting all $\tilde{y}_i > \alpha$ for some constant α, with the idea that pixels representing the subject will have larger $\tilde{y}_i$ than pixels in the background. This difference is likely as a large $\tilde{y}_i$ will be the result of peaks in the pixel spectrum. However background pixels will display little in the way of peaks due to the materials used on the spectrometer base, such as etched silver or gold [82], which are textured in such a way as to reduce reflectivity.

Principal component analysis of $\frac{\mathbf{Y}^T\mathbf{Y}}{n}$ may provide us with similar results in the scores, defined in §2.3 and used in Chapter 4, however will take considerably longer as p eigenvectors must be computed. In the situation where either the background displays some component spectra or our scan subject is in fact two very different compositions PCA will also provide us with more detail which may help us distinguish these regions. For example PCA will return us an eigenvector representing each component spectra, as used in §4.2, and so the corresponding scores may allow us to locate these two groups that summing the data may miss.

We used a similar process in chapter 5 when we created binary data in §5.4.2. In this chapter we saw a benefit in using a cluster analysis method as it required no user input in grouping the data.

7.2.1 Determining Subject Boundaries Using Cluster Analysis

To cluster our data we consider two methods, k-means and expectation maximisation of a Gaussian mixture model (EMGM), detailed in §2.10 and §2.12.1 respectively. Of these two k-means is the more simple technique, grouping the data in relation to cluster means.

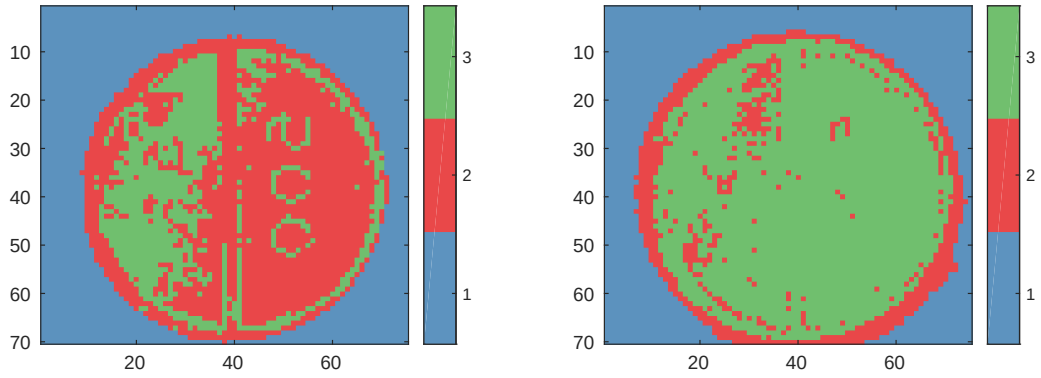
EMGM is more complex, and thus computationally more expensive, as it assumes each observation in our data has come from one of m Gaussian distributions as defined in §2.12.1. Whilst both of these methods can be applied to our entire $n \times p$ data matrix this will greatly increase the computation time, especially in the case of EMGM. We therefore sum our data matrix as in (7.2.1) to give a vector $\tilde{\mathbf{y}}$ of dimension $n \times 1$ which we can very quickly cluster.

We investigate the performance of the clustering methods by applying both to the paracetamol and human tooth data, see §1.2.2. For both of these datasets we expect to find a cluster of background pixels and a cluster of subject pixels. The paracetamol data contains a single component chemical, therefore a single cluster, and the human tooth contains two very similar components which we expect to be clustered together. However we must also consider the border pixels where the laser may have impacted both the subject and the background potentially giving us a less pronounced version of the subject spectra. We therefore search for three clusters using our algorithms, allowing us to choose in which group we place the border pixels. This will be useful as the border pixels may increase noise in our final results so removing them from the subject group will allow us to obtain the clearest results possible.

Our results from applying both k-means and EMGM to the datasets are shown in figures 7.1, 7.2 and 7.3. To generate these heatmaps of clusters we simply took our vector of clusters and reshaped it to the original Raman image dimensions as in §1.2. Figure 7.2 shows little difference in the performance of the two clustering methods. However figure 7.1 shows a benefit of EMGM over k-means in the detection of the border pixels. In figure 7.1a we see k-means has detected differences in the $\tilde{y}_i$ based on the surface pattern of the tablet, with the stamp of 500, from the dosage of 500mg, visible. Whilst in some forms of analysis this may be a useful tool, for our purposes EMGM appears more suitable. This increased ability of EMGM to detect border pixels and avoid assigning them to the background group due to higher levels of noise is explained in §2.12.2 where we detail the assigmnemt steps of the two methods.

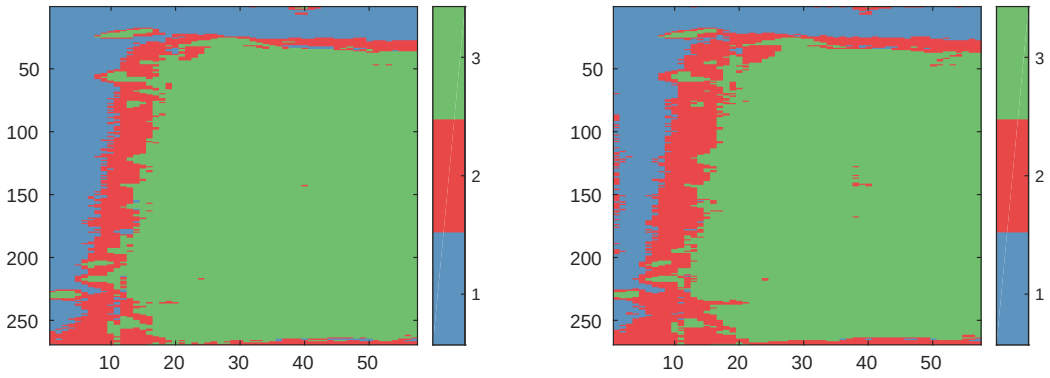
Therefore given three EMGM derived groups of $\tilde{y}_i$, say H_1, H_2 and H_3, containing subject, border and background pixels respectively we form our final groups by setting,

- $G_1 = H_1 \cup H_2$, $G_2 = H_3$, combine border and subject pixels, the method we use in this chapter.
- $G_1 = H_1$, $G_2 = H_3 \cup H_2$, exclude border pixels from the sample set for potentially clearer results.



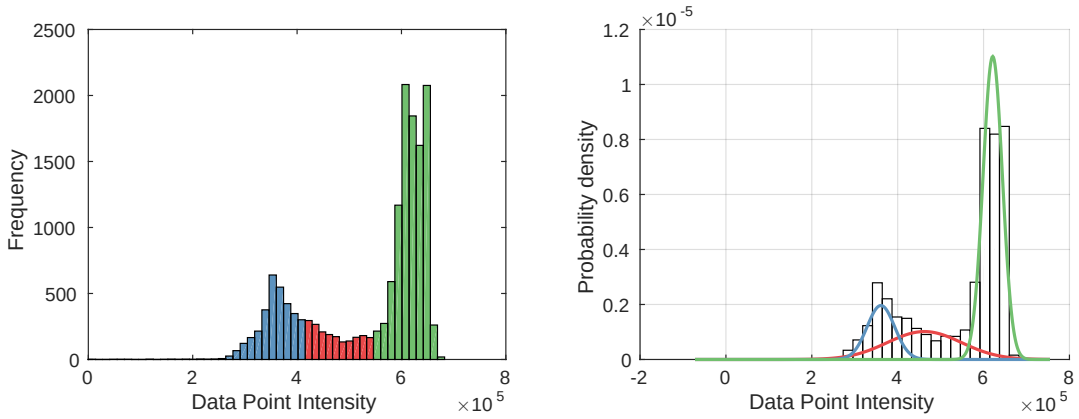
(a) *k*-means **(b)** EMGM

Figure 7.1: Heatmap of clusters generated by applying (6.2.4) to a vector of group numbers for each $\tilde{y}_i$, from (7.2.1), in the paracetamol dataset.



(a) *k*-means **(b)** EMGM

Figure 7.2: Heatmap of clusters generated by applying (6.2.4) to a vector of group numbers for each $\tilde{y}_i$, from (7.2.1), in the human tooth dataset.



(a) *k*-means **(b)** EMGM

Figure 7.3: Histograms of the $\tilde{y}_i$, from (7.2.1), for the human tooth data. Histograms are coloured using *k*-means derived groups or have the EMGM Gaussian models overlaid.

7.2.2 Bipartite Bilinear Model

With a method of separating our Raman image into two groups, background pixels and the scan subject, we now define our bipartite bilinear model

$$\mathbf{y}_i = \begin{cases} \mathbf{S}'\mathbf{c}'_i + \mathbf{e}_i & i \in \mathcal{S} \\ \mathbf{S}''\mathbf{c}''_i + \mathbf{e}_i & i \notin \mathcal{S} \end{cases} \tag{7.2.2}$$

for spectra

$$\mathbf{S}' = \left[\mathbf{s}'_1 \ \ldots \ \mathbf{s}'_{r'}\right] \quad \text{and} \quad \mathbf{S}'' = \left[\mathbf{s}''_1 \ \ldots \ \mathbf{s}''_{r''}\right] \tag{7.2.3}$$

and concentrations $\mathbf{c}'_i$ and $\mathbf{c}''_i$. The number of component spectra in the two groups, r' and r'' need not be equal. For EMGM derived clusters G_1 and G_2, detailed in §7.2.1, the set $\mathcal{S} = \{\tilde{y}_i | i \in G_1\}$ contains the observations, or pixels, assigned to the subject cluster. In the case of a dataset with no background but a scan subject comprised of regions of different chemical compositions this model can clearly still be fit by redefining groups G_1 and G_2. We can also easily extend the model to accommodate more than two groups of pixels.

We fit this model using the following algorithm and any method for solving (1.2.3).

Algorithm 7.2.1. Bipartite Model Algorithm

1. Use EMGM to cluster the $\tilde{y}_i$, from (7.2.1), and obtain G_1 and G_2 containing sample and background pixels respectively.

2. Partition $\mathbf{Y}$ into

$$\mathbf{Y}' = \{\mathbf{y}_i | i \in \mathcal{S}\}$$
$$\mathbf{Y}'' = \{\mathbf{y}_i | i \notin \mathcal{S}\}$$

for $\mathcal{S} = \{\tilde{y}_i | i \in G_1\}$

3. Set initial values for

$$\mathbf{C}' = \begin{bmatrix} \mathbf{c}_1'^T \\ \vdots \\ \mathbf{c}_{n_1}'^T \end{bmatrix} \quad \text{and} \quad \mathbf{C}'' = \begin{bmatrix} \mathbf{c}_1''^T \\ \vdots \\ \mathbf{c}_{n_2}''^T \end{bmatrix},$$

where $n_1 = |\mathcal{S}|$ and $n_2 = n - |\mathcal{S}|$, as randomly filled matrices, in practice $\mathcal{U}(0,1)$ data is a simple choice for the elements.

4. Minimise

$$\underset{S',C'}{\operatorname{argmin}} \|\mathbf{Y}' - \mathbf{C}'\mathbf{S}'^T\| \quad \text{and} \quad \underset{S'',C''}{\operatorname{argmin}} \|\mathbf{Y}'' - \mathbf{C}''\mathbf{S}''^T\| \tag{7.2.4}$$

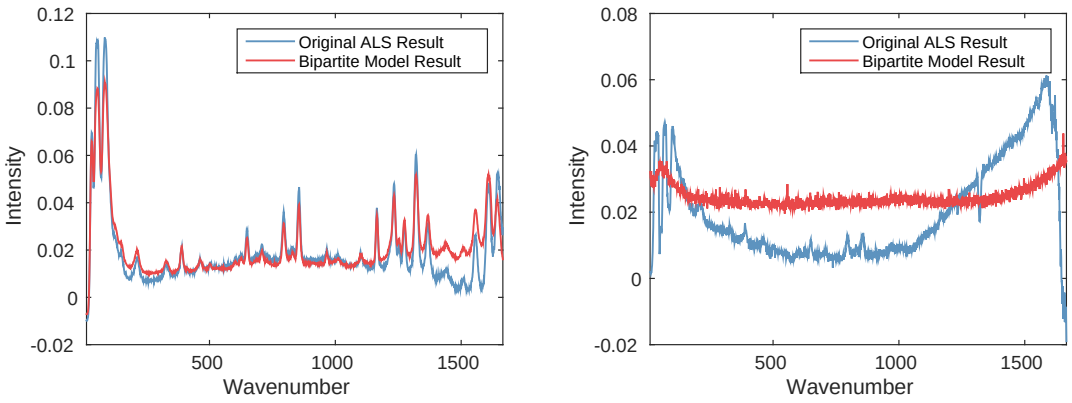
where $\|\cdot\|$ is an extension of the Euclidean norm to matrices,

$$\|\mathbf{X}\|^2 = \operatorname{tr}\left(\mathbf{X}^T\mathbf{X}\right).$$

7.2.3 Applying the Bipartite Model Algorithm to Raman Spectroscopy Data

To illustrate algorithm 7.2.1 we apply it to both the paracetamol and human tooth data used to investigate the clustering methods in §7.2.1. In implementing algorithm 7.2.1 we employed the ALS method to solve (7.2.4), detailed in §2.7.1. For the paracetamol data we set $r' = 2$ in (7.2.2) to allow for two component spectra in the scan subject group and $r'' = 1$ for a single spectra in the background group. Whilst there is only one component spectra in this dataset we include a second spectra so remaining background signal can be detected separately and any structure present will not affect our component spectra. For the human tooth data we set $r' = 3$as we expect to find the two component chemicals, enamel and dentin, and a third containing remaining background signal.

Our estimated spectra from algorithm 7.2.1 are shown in figures 7.4 and 7.5 for the paracetamol and human tooth datasets respectively. Our results for the paracetamol dataset show a marginal improvement. We see in figure 7.4b the remaining background signal present in the scan subject pixels. This now has a far less structured spectrum, with little in the way of peaks, as we expect from the unreflective scan bed. This reduction

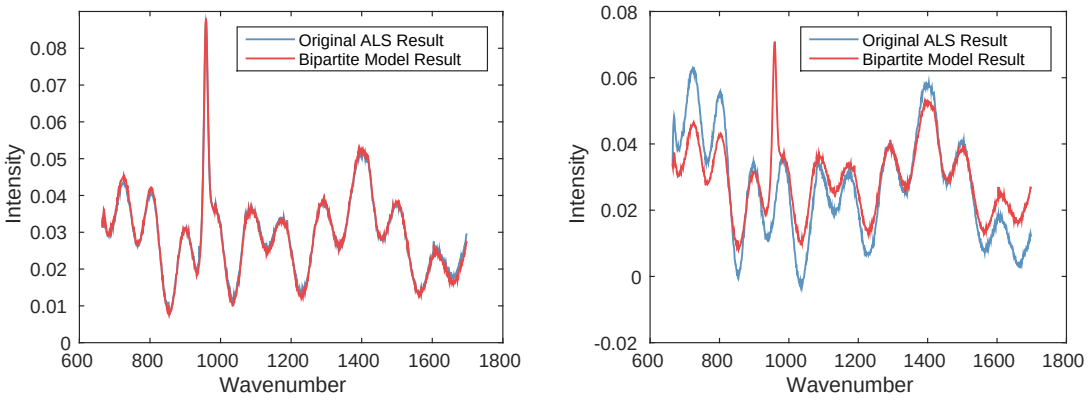


(a) Paracetamol, $\mathbf{s}_1'$ **(b)** Background Signal, $\mathbf{s}_1''$

Figure 7.4: Estimated spectra from algorithm 7.2.1 and conventional ALS, §2.7.1, for the paracetamol dataset. Use of the bipartite model (7.2.2) removes the parabolic structure estimated in the spectra.

in background signal structure has in turn provided us with a clearer paracetamol Raman spectra.

Our results for the human tooth data show almost no difference in the estimated enamel spectrum from solving (1.2.2) and (7.2.4), shown in figure 7.5a. However we see a large improvement in figure 7.5b in the estimated spectrum for dentin. Solving (1.2.2) did not find the dentin spectrum. This is likely due to it being extremely similar to that of enamel [38, 105] differing only in the intensity of the single peak [25]. Therefore the dentin spectrum may explain less variance than the background signal which replaced it as a result. However in using our bipartite model the majority of the background pixels were removed from the analysis of the scan subject, reducing the amount of variance in the data the background signal spectrum could explain. This results in the dentin spectrum being returned as a solution as it now explains a greater proportion of variance. In the case of this data where two of the component spectra are very similar it is common in the chemometrics community to separate the data into 'training data' and use a method such as linear discriminant analysis (LDA) to isolate the two spectra. However without the knowledge required to separate the Raman image this may not be possible and so an alternative to conventional rational resolution methods which is capable of distinguishing between extremely similar spectra will be beneficial.



(a) Enamel, $\mathbf{s}'_1$ **(b)** Dentin, $\mathbf{s}'_2$

Figure 7.5: Estimated spectra from algorithm 7.2.1 and conventional ALS, §2.7.1, for the human tooth dataset. Use of the bipartite model (7.2.2) allowed the algorithm to detect the dentin spectrum which differs from that of enamel only in the intensity of the single peak [25].

7.3 Fitting a Bilinear Gaussian Mixture Model

Whilst our algorithm for separating a Raman image into groups then solving (1.2.2) separately for each group is successful in providing us clearer spectra it fixes the groups to which pixels are assigned where these may be more accurately assigned using estimated spectra from subsequent stages of the algorithm. We therefore propose a method to both cluster the Raman image and solve (1.2.2). The goal is to improve the accuracy of the estimated spectra when compared to algorithm 7.2.1 by allowing the clusters to be adjusted based on the estimated spectra rather than fixed by the sum of each observation, (7.2.1).

We begin with the definition of a Gaussian mixture model, as defined in §2.12.1, for two Gaussian distributions,

$$f(\mathbf{y}_i) = \alpha f_1(\mathbf{y}_i|\boldsymbol{\mu}_{1i}, \sigma_1^2\mathbf{I}_p) + (1-\alpha) f_2(\mathbf{y}_i|\boldsymbol{\mu}_{2i}, \sigma_2^2\mathbf{I}_p)$$

where α is the mixture weight and f_1 and f_2 are probability density functions of p-dimensional multivariate Gaussian distributions. We incorporate our bilinear models by defining the means of the Gaussian distributions as $\boldsymbol{\mu}_{1i} = \mathbf{S}'\mathbf{c}'_i$ and $\boldsymbol{\mu}_{2i} = \mathbf{S}''\mathbf{c}''_i$ respectively. We define the variance of these distributions as $\sigma^2\mathbf{I}_p = \sigma_1^2\mathbf{I}_p = \sigma_2^2\mathbf{I}_p$ to continue our assumption of independent and identically distributed errors from §1.2. This gives us two parameters, $\mathbf{c}$ and $\mathbf{S}$, in place of $\boldsymbol{\mu}$ and our bilinear Gaussian mixture model

$$f(\mathbf{y}_i) = \alpha f_1(\mathbf{y}_i|\mathbf{S}'\mathbf{c}'_i, \sigma^2\mathbf{I}_p) + (1-\alpha) f_2(\mathbf{y}_i|\mathbf{S}''\mathbf{c}''_i, \sigma^2\mathbf{I}_p) \tag{7.3.1}$$

We immediately see the benefit of this separation of $\boldsymbol{\mu}$ when we look at the number

of parameters in the standard model and in the separated model. Before decomposing $\boldsymbol{\mu}_{1i} = \mathbf{S}'\mathbf{c}'_i$ and $\boldsymbol{\mu}_{2i} = \mathbf{S}''\mathbf{c}''_i$ our model had np observations and $2np + 2$ parameters, the means and variances. However after decomposition the number of parameters in the decomposed $\boldsymbol{\mu}_{1i}$ and $\boldsymbol{\mu}_{2i}$ equals $n(r' + r'') + p(r' + r'') - (r' + r'')$, where the subtraction of $(r' + r'')$ comes from the unit constraints on the $\mathbf{S}$. Our total number of parameters is now $n(r' + r'') + p(r' + r'') - (r' + r'') + 2$. Therefore if we compare the parameter numbers for some of our common datasets, where $n = 5000$, $p = 1000$ and $r = 3$ (leading to $r' = 1$ and $r'' = 2$), our parameter numbers fall from a little over 10^7 to only around 18000.

We now use a vector, $\mathbf{z}$, of indicator variables such that $\alpha = \mathrm{P}(z_i = 1)$ to augment (7.3.1) [94] allowing us to define the joint density as a product, instead of the sum (7.3.1),

$$f(\mathbf{y}_i, z_i) = \{\alpha f_1(\mathbf{y}_i)\}^{z_i} \{(1 - \alpha) f_2(\mathbf{y}_i)\}^{1 - z_i}$$

because in the case that $z_i = 1$, the joint likelihood equals $\alpha f_1(\mathbf{y}_i)$ and when $z_i = 0$ we get $(1 - \alpha) f_2(\mathbf{y}_i)$. We therefore define the likelihood,

$$L(\boldsymbol{\theta}|\mathbf{Y}, \mathbf{z}) = \prod_{i=1}^{n} \{\alpha f_1(\mathbf{y}_i)\}^{z_i} \{(1 - \alpha) f_2(\mathbf{y}_i)\}^{1 - z_i}$$

and thus the log likelihood

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{Y}, \mathbf{z}) = \sum_{i=1}^{n} \left\{ z_i \log \left[\alpha f_1(\mathbf{y}_i|\mathbf{S}'\mathbf{c}'_i, \sigma^2\mathbf{I}_p)\right] + (1 - z_i) \log \left[(1 - \alpha) f_2(\mathbf{y}_i|\mathbf{S}''\mathbf{c}''_i, \sigma^2\mathbf{I}_p)\right] \right\} \tag{7.3.2}$$

for parameter vector $\boldsymbol{\theta} = \left\{\alpha, \mathbf{C}', \mathbf{C}'', \mathbf{S}', \mathbf{S}'', \sigma^2\right\}$. Augmenting our data using $\mathbf{z}$ gives a log likelihood that is easily differentiable, and therefore easier to maximise, as we can use the identity $\log(AB) = \log(A) + \log(B)$. If we were to calculate the log-likelihood using (7.3.1) no such logarithmic identity would be possible.

7.3.1 Expectation Maximisation of the Bilinear Gaussian Mixture Model

With our augmented log likelihood defined we now calculate the update formulae for the E-step and M-step of the EM algorithm, detailed in §2.12.1. To calculate the E-step of the expectation maximisation algorithm, from which new estimates of $\boldsymbol{\theta}$ can be obtained

using current estimates $\boldsymbol{\theta}^{(t)}$, it can be shown [11]

$$\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}, \mathbf{Y}) &= \sum_{z\in\{0,1\}^n} p(\mathbf{z}|\mathbf{Y}, \boldsymbol{\theta}^{(t)})\mathcal{L}(\boldsymbol{\theta}^{(t)}|\mathbf{Y}, \mathbf{z}) \\
&= \mathrm{E}_{\mathrm{z}|\mathrm{Y},\boldsymbol{\theta}^{(t)}}\left[\mathcal{L}(\boldsymbol{\theta}|\mathbf{Y}, \mathbf{z})\right] \\
&= \sum_{i=1}^{n} \mathrm{E}\left[\mathcal{L}(\boldsymbol{\theta}|\mathbf{y}_i, \mathbf{z}_i)\right] \\
&= \sum_{i=1}^{n} \left\{ \mathrm{P}(z_i = 1|\mathbf{y}_i, \boldsymbol{\theta}^{(t)}) \log\left[\alpha f_1(\mathbf{y}_i|\boldsymbol{\theta}^{(t)})\right] + \mathrm{P}(z_i = 0|\mathbf{y}_i, \boldsymbol{\theta}^{(t)}) \log\left[(1-\alpha) f_2(\mathbf{y}_i|\boldsymbol{\theta}^{(t)})\right] \right\}.
\end{aligned}$$

Now using Bayes' theorem, $\mathrm{P(A|B)} = \frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}}$, we can define posterior class probabilities $w_i^{(t)} = \mathrm{P}(z_i = 1|\mathbf{y}_i, \boldsymbol{\theta}^{(t)})$ with,

$$w_i^{(t)} = \frac{\alpha^{(t-1)} f_1(\mathbf{y}_i|\boldsymbol{\theta}^{(t-1)})}{\alpha^{(t-1)} f_1(\mathbf{y}_i|\boldsymbol{\theta}^{(t-1)}) + (1-\alpha^{(t-1)}) f_2(\mathbf{y}_i|\boldsymbol{\theta}^{(t-1)})} \tag{7.3.3}$$

as identifying $P(\mathrm{B})$ with $\mathrm{P}(\mathbf{y}_i, \boldsymbol{\theta}^{(t)})$ is the same formula as our mixture model (7.3.1), $\mathrm{P(B|A)} = \mathrm{P}(\mathbf{y}_i, \boldsymbol{\theta}^{(t)}|z_i = 1) = f_1(\mathbf{y}_i|\boldsymbol{\theta}^{(t)})$ from (7.3.1) and $\mathrm{P(A)} = \alpha$. In this formula we take the $\alpha^{(t-1)}$ from either the previous iteration of the algorithm or the initial membership weights, the estimation of which is discussed in algorithm 7.3.1.

Therefore we define our E-step formula as

$$\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}, \mathbf{Y}) &= \sum_{i=1}^{n} \left\{ w_i \log\left[\alpha f_1(\mathbf{y}_i|\boldsymbol{\theta}^{(t)})\right] + (1-w_i) \log\left[(1-\alpha) f_2(\mathbf{y}_i|\boldsymbol{\theta}^{(t)})\right] \right\} \\
&= \sum_{i=1}^{n} \Big\{ w_i \log(\alpha) - \frac{w_i p}{2}\log(2\pi) - \frac{w_i p}{2}\log(\sigma^2) \ldots \\
&\ldots - \frac{w_i}{2\sigma^2}\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}_i'\right)^T\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}_i'\right) + (1-w_i)\log(1-\alpha) \\
&\ldots - \frac{(1-w_i)p}{2}\log(2\pi) - \frac{(1-w_i)p}{2}\log(\sigma^2) \ldots \\
&\ldots - \frac{(1-w_i)}{2\sigma^2}\left(\mathbf{y}_i - \mathbf{S}''\mathbf{c}_i''\right)^T\left(\mathbf{y}_i - \mathbf{S}''\mathbf{c}_i''\right) \Big\}.
\end{aligned} \tag{7.3.4}$$

We now differentiate (7.3.4) with respect to the parameters in $\boldsymbol{\theta}$ to perform the M-step. By differentiating with respect to α we get

$$\begin{aligned}
\frac{\partial \mathrm{Q}}{\partial \alpha} &= \sum_{i=1}^{n} \left\{ \frac{w_i}{\alpha} - \frac{1-w_i}{1-\alpha} \right\} \\
&= \sum_{i=1}^{n} \left\{ \frac{w_i - \alpha}{\alpha(1-\alpha)} \right\} = 0 \\
\implies \hat{\alpha} &= \frac{1}{n}\sum_{i=1}^{n} w_i
\end{aligned}$$

which we write as update formula

$$\alpha^{(t)} = \frac{1}{n}\sum_{i=1}^{n} w_i^{(t)}. \tag{7.3.5}$$

We show this is a maximum by evaluating the second derivative and substituting in $\hat{\alpha}$ to give

$$-\sum_{i=1}^{n}\left\{\frac{\hat{\alpha}^2 + w_i - 2\hat{\alpha}w_i}{\hat{\alpha}^2(1-\hat{\alpha})^2}\right\}.$$

As the denominator is clearly positive showing this is negative is equivalent to showing

$$n\sum_{i=1}^{n} w_i > \left(\sum_{i=1}^{n} w_i\right)^2.$$

Now in the case that all $w_i = 1$, i.e. there is only one Gaussian distribution in our mixture, both sides equal n^2 and we have a saddle point. However assuming there exists two distributions in our mixture at least one $w_i < 1$. Therefore $\sum_{i=1}^{n} w_i < n$, our inequality holds and $\hat{\alpha}$ is our maximum.

Similarly we find the maximum likelihood estimator for σ^2 with

$$\begin{aligned}\frac{\partial Q}{\partial \sigma^2} &= \frac{-np}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}\Big\{w_i\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}_i'\right)^T\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}_i'\right) \ldots \\ &\quad \ldots + (1-w_i)\left(\mathbf{y}_i - \mathbf{S}''\mathbf{c}_i''\right)^T\left(\mathbf{y}_i - \mathbf{S}''\mathbf{c}_i''\right)\Big\} = 0 \\ \implies \hat{\sigma}^2 &= \frac{1}{np}\sum_{i=1}^{n}\Big\{w_i\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}_i'\right)^T\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}_i'\right) + (1-w_i)\left(\mathbf{y}_i - \mathbf{S}''\mathbf{c}_i''\right)^T\left(\mathbf{y}_i - \mathbf{S}''\mathbf{c}_i''\right)\Big\}\end{aligned}$$

which becomes update formula

$$\begin{aligned}\sigma^{2(t)} = \frac{1}{np}\sum_{i=1}^{n}\Big\{&w_i^{(t)}\left(\mathbf{y}_i - \mathbf{S}'^{(t-1)}\mathbf{c}_i'^{(t-1)}\right)^T\left(\mathbf{y}_i - \mathbf{S}'^{(t-1)}\mathbf{c}_i'^{(t-1)}\right) + \\ &(1-w_i^{(t)})\left(\mathbf{y}_i - \mathbf{S}''^{(t-1)}\mathbf{c}_i''^{(t-1)}\right)^T\left(\mathbf{y}_i - \mathbf{S}''^{(t-1)}\mathbf{c}_i''^{(t-1)}\right)\Big\}.\end{aligned} \tag{7.3.6}$$

Again we calculate the second derivative and substitute in $\hat{\sigma}^2$ to obtain

$$\frac{np\hat{\sigma}^2 - 2\sum_{i=1}^{n}\left\{w_i\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}_i'\right)^T\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}_i'\right) + (1-w_i)\left(\mathbf{y}_i - \mathbf{S}''\mathbf{c}_i''\right)^T\left(\mathbf{y}_i - \mathbf{S}''\mathbf{c}_i''\right)\right\}}{2\hat{\sigma}^6}.$$

As before the denominator is clearly positive and so we require a negative numerator for a maximum. This is clear when we write the numerator as $np\hat{\sigma}^2 - 2np\hat{\sigma}^2 = -np\hat{\sigma}^2$. With n, p and $\hat{\sigma}^2$ all positive we have a maximum.

Deriving the partial derivatives for $\mathbf{c}'$ and $\mathbf{c}''$ is a little more difficult as we must differentiate with respect to vectors. However this is a similar process to deriving the ordinary least

squares estimator giving us

$$\frac{\partial Q}{\partial \mathbf{c}'_i} = \frac{-w_i}{2\sigma^2}\left(-2\mathbf{S}'^T\mathbf{y}_i + 2\mathbf{S}'^T\mathbf{S}'\mathbf{c}'_i\right) = \mathbf{0}$$
$$\implies \hat{\mathbf{c}}'_i = \left(\mathbf{S}'^T\mathbf{S}'\right)^{-1}\mathbf{S}'^T\mathbf{y}_i$$

and, by similar calculations,

$$\hat{\mathbf{c}}''_i = \left(\mathbf{S}''^T\mathbf{S}''\right)^{-1}\mathbf{S}''^T\mathbf{y}_i$$

which lead us to update formulae

$$\mathbf{c}'^{(t)}_i = \left(\mathbf{S}'^{T(t)}\mathbf{S}'^{(t)}\right)^{-1}\mathbf{S}'^{T(t)}\mathbf{y}_i \tag{7.3.7}$$
$$\mathbf{c}''^{(t)}_i = \left(\mathbf{S}''^{T(t)}\mathbf{S}''^{(t)}\right)^{-1}\mathbf{S}''^{T(t)}\mathbf{y}_i \tag{7.3.8}$$

These estimators are in the same form as the least squares estimator of the $\mathbf{c}_i$ in $\mathbf{y}_i = \mathbf{S}\mathbf{c}_i$ and we found in testing the algorithm that it was possible to substitute in the non-negative least squares algorithm detailed in §2.6.2.

As analytical differentiation with respect to a matrix is only possible in matrix-by-scaler situations we must decompose $\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}'_i\right)^T\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}'_i\right)$ before we are able to differentiate (7.3.4) with respect to $\mathbf{S}'$ or $\mathbf{S}''$. We first define $\mathbf{s}_j$ as a column representation of the j^{th} row of $\mathbf{S}$ and then derive

$$\begin{aligned}
&\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}'_i\right)^T\left(\mathbf{y}_i - \mathbf{S}'\mathbf{c}'_i\right) \\
=&\sum_{j=1}^{p}\left(y_{i,j} - \mathbf{c}'^T_i\mathbf{s}'_j\right)^T\left(y_{i,j} - \mathbf{c}'^T_i\mathbf{s}'_j\right) \\
=&\sum_{j=1}^{p}\left(y^2_{i,j} + \mathbf{s}'^T_j\mathbf{c}'_i\mathbf{c}'^T_i\mathbf{s}'_j - 2y_{i,j}\mathbf{c}'^T_i\mathbf{s}'_j\right)
\end{aligned} \tag{7.3.9}$$

allowing us to replace differentiation with respect to a matrix, $\mathbf{S}'$ or $\mathbf{S}''$, by the sum of derivatives with respect to a vector. By substituting (7.3.9) into (7.3.4) we can now find the remaining partial derivatives

$$\frac{\partial Q}{\partial \mathbf{s}'_j} = \frac{1}{\sigma^2}\sum_{i=1}^{n} -w_i\left(2\mathbf{c}'_i\mathbf{c}'^T_i\mathbf{s}'_j - 2y_{i,j}\mathbf{c}'_i\right) = \mathbf{0}$$
$$\implies \left(\sum_{i=1}^{n} w_i\mathbf{c}'_i\mathbf{c}'^T_i\right)\mathbf{s}'_j = \sum_{i=1}^{n} w_i y_{i,j}\mathbf{c}'_i$$
$$\implies \hat{\mathbf{s}}'_j = \left(\sum_{i=1}^{n} w_i\mathbf{c}'_i\mathbf{c}'^T_i\right)^{-1}\sum_{i=1}^{n} w_i y_{i,j}\mathbf{c}'_i.$$

Using similar substitutions we derive

$$\hat{\mathbf{s}}''_j = \left(\sum_{i=1}^{n}(1-w_i)\mathbf{c}''_i\mathbf{c}''^T_i\right)^{-1}\sum_{i=1}^{n}(1-w_i)y_{i,j}\mathbf{c}''_i$$

which lead us to update formulae

$$\mathbf{s}_j'^{(t)} = \left(\sum_{i=1}^{n} w_i^{(t)} \mathbf{c}_i' \mathbf{c}_i'^{T(t)} \right)^{-1} \sum_{i=1}^{n} w_i^{(t)} y_{i,j} \mathbf{c}_i'^{(t)}. \tag{7.3.10}$$

$$\mathbf{s}_j''^{(t)} = \left(\sum_{i=1}^{n} (1 - w_i^{(t)}) \mathbf{c}_i''^{(t)} \mathbf{c}_i''^{T(t)} \right)^{-1} \sum_{i=1}^{n} (1 - w_i^{(t)}) y_{i,j} \mathbf{c}_i''^{(t)}. \tag{7.3.11}$$

With our update functions for (7.3.1) derived we must address the issue surrounding $\mathbf{S}$ and $\mathbf{C}$ before we can implement the EM algorithm for our bilinear Gaussian mixture model. The issue with these two parameters is that they must simultaneously satisfy (7.3.7), (7.3.8), (7.3.10) and (7.3.11). However we note that this is the same situation in which we find ourself when estimating solutions to $\mathbf{Y} = \mathbf{C}\mathbf{S}^T$ using alternating least squares, §2.6.1, which also has simultaneous MLEs. It has been shown that alternately updating the MLEs will provide convergence [64] and so we implement the same process in our EM algorithm.

Algorithm 7.3.1. Expectation Maximisation of a Bilinear Gaussian Mixture Model

1. For iteration $t = 0$ initialise $w_i^{(0)}$ using one of the following

   - Threshold of $\tilde{y}_i \geq \phi$, for $\tilde{y}_i$ from (7.2.1), to give initial membership probabilities

   $$w_i = \begin{cases} 1 & \text{if } \tilde{y}_i \geq \phi \\ 0 & \text{if } \tilde{y}_i < \phi \end{cases}$$

   - Cluster analysis as in §7.2.1 to give mixture weights

   $$w_i = \begin{cases} 1 & \text{if } i \in G_1 \\ 0 & \text{if } i \in G_2 \end{cases}$$

   - $\mathcal{U}(0,1)$ random numbers to give random mixture weights for all n pixels.

   where the former two methods are very effective with Raman images due to very clear patterns emerging when the observations are summed. Initialise $\hat{\mathbf{c}}_i'$ and $\hat{\mathbf{c}}_i''$ with matrices of random numbers from a non-negative distribution such as the exponential distribution or absolute values from a distribution such as the normal distribution.

2. For $t \geq 1$ calculate $w_i^{(t)}$ in (7.3.3), for $i = 1, ..., n$.

3. For $t \geq 1$ calculate $\hat{\alpha}^{(t)}$ in (7.3.5).

4. Calculate $\hat{\sigma}^{2(t)}$ in (7.3.6).

5. Calculate $\boldsymbol{\mu}_{1i}^{(t)} = \mathbf{c}_i' \mathbf{S}'^T$ and $\boldsymbol{\mu}_{2i}^{(t)} = \mathbf{c}_i'' \mathbf{S}''^T$ by separating $\mathbf{Y}$ into groups $\mathbf{Y}'$ and $\mathbf{Y}''$ and finding solutions to $\mathbf{Y}' = \mathbf{C}'^{(t)} \mathbf{S}'^{T(t)}$ and $\mathbf{Y}''^{(t)} = \mathbf{C}'' \mathbf{S}''^{T(t)}$ with the steps

(a) Calculate $\hat{\mathbf{S}}'^{(t)}$ and $\hat{\mathbf{S}}''^{(t)}$ with (7.3.10) and (7.3.11) respectively. Normalise such that $\|\hat{\mathbf{s}}_k'^{(t)}\| = 1$ and $\|\hat{\mathbf{s}}_k''^{(t)}\| = 1$ for $k = 1, ..., r$. Use $\mathbf{C}'^{(t-1)}$ and $\mathbf{C}'^{(t-1)}$ in the calculations

(b) Calculate $\hat{\mathbf{c}}_i'^{(t)}$ and $\hat{\mathbf{c}}_i''^{(t)}$ for $i = 1, ..., n$ with (7.3.7) and (7.3.8) respectively. We found estimating $\hat{\mathbf{c}}_i'^{(t)}$ and $\hat{\mathbf{c}}_i''^{(t)}$ was possible using NNLS, §2.6.2, in spite of it solving a different model. However as this is not the maximiser of our bilinear Gaussian model convergence is not guaranteed.

(c) Return to step (a) and repeat as in algorithm 2.7.1 until convergence of the estimates.

6. Calculate the log-likelihood, (7.3.2), for $\boldsymbol{\theta}^{(t)} = \left\{\hat{\alpha}, \hat{\mathbf{C}}', \hat{\mathbf{C}}'', \hat{\mathbf{S}}', \hat{\mathbf{S}}'', \hat{\sigma}^2\right\}$ and check for convergence by comparing to iteration $t-1$ using

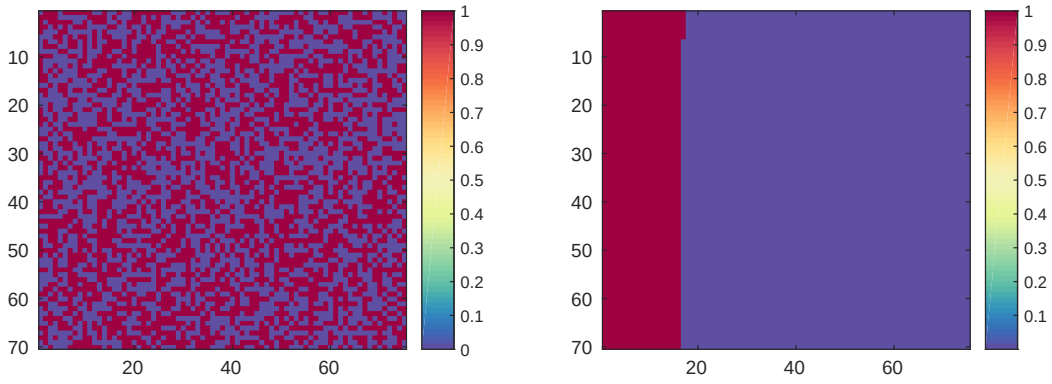
$$Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t-1)}, \boldsymbol{Y}) - Q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^{(t-1)}, \boldsymbol{Y}) > \Delta$$

for some threshold Δ. If the difference between update function values is greater than Δ progress to $t+1$ and return to step 2 else progress to step 7. This criterion also ensures the algorithm increases at each iteration, a requirement worth checking due to the potential sub-optimal estimation of $\mathbf{S}$ and $\mathbf{C}$ using an alternating, or even NNLS, method. In practice we found $\Delta = 1 \times 10^{-8}$ to be sufficient for stable solutions without excessive iterations.

7. Calculate the final group memberships using

$$\mathbf{z}_i = \begin{cases} 1 & \text{if } w_i > \frac{1}{2} \\ 0 & \text{if } w_i \leq \frac{1}{2} \end{cases}$$

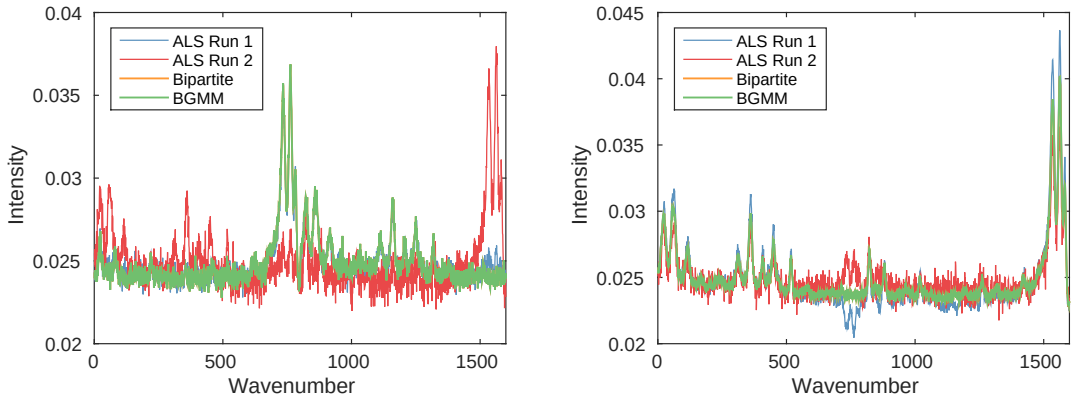
Assuming NNLS is not used in step 5 convergence of the EM algorithm follows from the work of Wu [104] if $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}, \mathbf{Y})$ is a continuous function and if at each step the parameters are updated using the maximum of the transfer function $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}, \mathbf{Y})$. In this situation the algorithm will converge to parameters that are a maximum of the likelihood function of our data. Although this convergence may be a local maximum so Wu [104] recommended running the algorithm several times with different starting values for $\boldsymbol{\theta}$. The convergence of the EM algorithm with NNLS substituted is not guaranteed as we are not using the maximiser of our model but that of a different constrained model. Proof of convergence of the EM algorithm when applied to this model is an area open for further research. In practice we found that using the NNLS algorithm still allowed the EM algorithm to converge.



(a) Initial

(b) Final estimates from algorithm 7.3.1 and true clusters

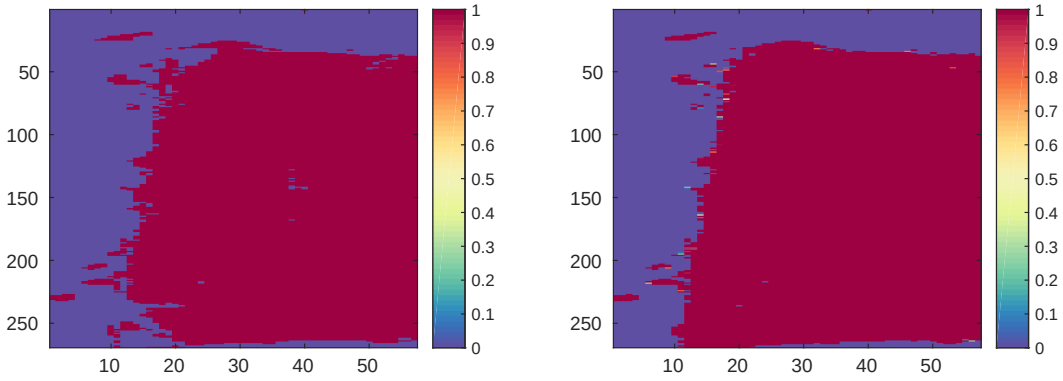
Figure 7.6: Heatmap of clusters generated by applying (6.2.4) to a vector of group numbers for each pixel in the simulated data.



(a) Cluster 1

(b) Cluster 2

Figure 7.7: Component spectra detected in the simulated data using ALS, §2.7.1, the bipartite model algorithm, 7.2.1, and algorithm, 7.3.1.



(a) Initial **(b)** Final estimates from algorithm 7.3.1

Figure 7.8: Heatmap of clusters generated by applying (6.2.4) to a vector of group numbers for each pixel in the human tooth data.

7.3.2 Fitting the Bilinear Gaussian Mixture Model to Simulated and Real Data

To test algorithm 7.3.1 we first simulated a simple dataset with $n = 4900$, $r' = r'' = 1$ and

$$\mathbf{y}_i = c_{i,1}\mathbf{s}_1 + 0\mathbf{s}_2 + \varepsilon_i \quad \text{for } i = 1, \ldots, 1225$$
$$\mathbf{y}_i = 0\mathbf{s}_1 + c_{i,2}\mathbf{s}_2 + \varepsilon_i \quad \text{for } i = 1226, \ldots, 4900$$

with

$$\varepsilon_i \sim \mathcal{N}(0, 1)$$
$$c_{i,1} \sim \mathcal{N}(10, 1)$$
$$c_{i,2} \sim \mathcal{N}(5, 1)$$

where $\mathbf{s}_1$ is a spectrum with a single peak in the centre and $\mathbf{s}_2$ is the paracetamol spectrum. This gives us a dataset where 25% of the pixels are generated by one spectrum and the remaining 75% are generated by a different spectrum. We see our initial, entirely random, cluster estimates in figure 7.6 alongside the final clusters identified by algorithm 7.3.1 which are equal to the original simulated mixture weights. Figure 7.7 also shows an improvement in the detected spectra over conventional methods of solving (1.2.2) such a ALS, §2.7.1. The results obtained from our first run with ALS correctly detected $\mathbf{s}_1$ however returned an incorrect estimate of $\mathbf{s}_2$, with an anomalous reversed peak. The second ALS run detected two noisy copies of $\mathbf{s}_2$, having selected two different solutions from the solution space defined in chapter 4. The results from algorithm 7.3.1 however detected both spectra correctly and clearly with no interference between the two. Our two stage algorithm, 7.2.1, also correctly identified the clusters, using EMGM, and the component spectra. The advantage

of algorithm 7.3.1 however is that the spectra and clusters are alternately estimated and so the clusters can be refined based on the updated spectra estimates.

We then applied algorithm 7.3.1 to the human tooth data used in §7.2.3. For this data we initialised $\mathbf{w}$ using the threshold method with $\tilde{y}_i < 150000$ giving the scanned object cluster estimate we see in figure 7.8a. Due to the clusters being initialised from the data rather than random $\mathcal{U}(0,1)$ values the difference between the initial and final group estimates is not as distinct as with the simulated data. However figure 7.8b shows a moderate improvement in isolating the tooth from the background. The estimated spectra were also very similar to those found with the bipartite model shown in figure 7.5, with almost no perceivable change. Throughout our application of this algorithm we noticed that the choice of initialisation parameters made little difference to the results of the algorithm, with any small differences likely due to the random selection of a solution from a space of solutions as with the standard ALS algorithm 2.7.1.

7.4 Discussion

In this chapter our objective was to investigate methods of separating our data into clusters to provide clearer estimated component spectra and concentrations. We first investigated k-means and EMGM clustering methods and came to the conclusion that EMGM was more successful in separating pixels on the border between the scan subject and the background. We then defined and implemented a bipartite bilinear model which allowed us to reduce the dimension of parts of the model where there were fewer component spectra. Results from both the paracetamol and human tooth datasets were promising with estimated component spectra for paracetamol appearing clearer and with less interference from background signal. It was with the human tooth data however that we saw the greatest improvement with the bipartite model detecting both component spectra in spite of their high similarity. This similarity caused (1.2.2) to detect only a single component chemical when solutions were estimated with ALS, §2.7.1.

With the EMGM clustering algorithm and our bipartite model proving effective at estimating results with greater clarity we investigated a combination of the two. We modified the Gaussian mixture model of EMGM to include our bilinear model estimates of the concentrations and spectra as the distribution means, allowing us to reduce the model dimensions in parts of the image whilst allowing the clusters to change throughout the algorithm. With the maximum likelihood estimators of this mixture model derived we calculated the steps of the EM algorithm to iteratively fit this new model. Our algorithm performed well on both simulated data and our human tooth dataset, accurately isolating clusters and very similar

component spectra. Whilst the results obtained were very similar to those generated by the previous algorithm we noted the key improvement that the iterative refinement of both the parameter estimates and the mixture weights may allow the algorithm to predict more accurate clusters based on repeatedly refined parameters.

Our proposed method bears some resemblance to spectral decomposition methods which use isolated regions of the Raman image to provide clearer estimated spectra, some of which are detailed in §2.5 and [59, 71, 72]. In comparison to these methods our proposed algorithm requires no user interaction to isolate regions of the image or specific features in the dataset where component chemicals are known to be fewer in number.

Future work in this area may investigate the convergence of the estimates of $\mathbf{S}$ and $\mathbf{C}$ in algorithm 7.3.1. With the MLEs for these two sets of parameters being simultaneous we rely on the proof of convergence from the alternating least squares method, §2.6.1. This proof however may need confirming due to the changing clusters. The same convergence investigation may also be required if NNLS, §2.6.2, is used in step 5 of algorithm 7.3.1 to estimate non-negative concentrations. Whilst this algorithm has been proven to converge the presence of changing clusters could, once again, cause problems.

Chapter 8

Discussion and Further Work

In this thesis we investigated a variety of multivariate techniques for analysing Raman images. For half of the thesis our focus was on improving the spectral decomposition of Raman images, as defined in §1.2, using current methods, Chapters 3 and 4, and a new spectral decomposition method, Chapter 7. For the second half of the thesis we investigate methods of analysing the results of spectral decomposition, Chapters 5 and 6.

8.1 Improving the Spectral Decomposition

In analysing a Raman spectroscopic image the resulting spectra, **S**, and concentrations, **C**, can often include anomalies from the original data. These can manifest in many ways for example extra peaks in spectra, linear trends in spectra and pixels with a considerably higher concentration than others in the image. To improve our results from spectral decomposition in Chapter 3 we investigated techniques for pre-processing the data. For anomaly removal we proposed a process similar to a Windsorised mean where the values lying within the data not including the 5% tails are used to replace the anomalously high or low values. We used this technique to remove cosmic ray anomalies, manifesting as high elements in the data matrix, and burnt pixels, appearing as high data matrix rows. We selected potential anomalies by ordering the data and taking the highest 0.001% and 0.0001% of values for cosmic rays and burnt pixels respectively. Even using these small values we were able to almost completely remove visible occurrences of both types of anomalies in our Raman image of a paracetamol tablet. Removal was judged by comparing the estimated spectra to a reference spectra [75] and by inspecting the estimated concentrations for unusually high pixels where a tablet surface would be expected to be smooth. Our methods were also more simple than currently proposed techniques and required less computation time and cost. For example the specialised laser receivers required to limit the occurrence of

cosmic ray anomalies will increase the cost of the already expensive Raman spectrometers [76] and taking repeated scans at each pixel, another popular method, can greatly increase scan time and cost [50]. The methods sharing more similarities with our proposed method use clusters of pixels or the standard deviation of pixels to locate anomalies [83, 98]. It is here that our method improves on computation time at the expense of requiring user input for the threshold above which observation intensities are judged to be anomalously high. Future work on this area could involve automating the identification of the cut threshold. For example if the distribution of the values in the Raman image were identified removing cosmic rays, anomalously high observations, could be a matter of removing or smoothing the values in the upper 95% of the distribution.

To more directly improve the results of spectral decomposition we investigated the effects of variance scaling the data. With standard variance scaling a common technique in analysing Raman images [32], used to improve the clarity of estimated spectra, we took the idea of using Poisson scaling from Keenan and Kotula [57] who applied it to mass spectra. Whilst conventional scaling performed as expected and was used throughout the thesis when we required clear spectra our implementation of Poisson scaling was less successful with Raman images than mass spectroscopy.

We also improved the clarity of our estimated spectra by correcting the baseline of the Raman image using a modified linear model to enforce positive residuals from the model. This proved very effective at making spectra appear more physically correct for the purposes of identifying component chemicals. Currently published methods for achieving this are primarily focused on chromatography NMR spectroscopy and involve separating the spectra into parts containing no peaks and calculating how far from zero these are before correcting [19, 20]. A popular method for Raman spectroscopy data was proposed by Zhang et al. [106] and uses weighted least squares to correct the baseline whilst preserving peaks. Whilst our method may not be as effective as these when compared with the results in [19, 20, 106] it is more simple, as the spectra are corrected as a whole rather than in parts, and can be applied with a wide range of optimisation techniques.

In Chapter 4 we investigated the non-identifiability of our model (1.2.3) and the resulting solution space, with the aim of accounting for the infinite range of solutions into any testing we perform. We began with a more in depth investigation into Lawton's original method, extending the two spectra solution space to allow for three or more component spectra. For three or more component spectra we were unable to calculate analytic solution space bounds and so defined a random walk about this space. The purpose of this random walk was to allow us to sample a range of possible solutions to our bilinear model under the knowledge that our assumptions, (1.2.5), would be met. Our random walk began with

a known solution, estimated using any method of solving (1.2.3), and randomly moved this solution about the solution space testing for non-negative spectra. With a region generating non-negative spectra identified we then tested for non-negative concentrations to obtain an estimate of the solution space. On our sample data our random walk proved very effective at identifying the range of possible solutions our model may obtain. We then combined these results with cluster analysis to group the possible solutions for each spectra. In comparison to published work on the non-identifiability of (1.2.3) by Tauler [95] and Garrido et al. [35] our method has no specific requirements on features in the data. (For example Tauler [95] restricts the transformation matrix Ξ to enforce unimodal spectra and Garrido et al. [35] restricts Ξ using regions of known concentration values where a component spectra has zero concentration [35].) However, the random walk we propose is slower than current methods due to the process with which all transformations of the solutions must be tested. Whereas the two published methods previously mentioned use numerical optimisation to find transformation matrices giving component spectra which differ from each other as much as possible, with difference calculated using the Euclidean distance between the spectra.

We therefore have scope for further work in our random walk of the solution space to speed up the process and reduce computation costs. Currently selection of the transformation matrix Ξ, used to move about the solution space, is entirely random and therefore results in rejected steps. The only obvious limit we have with regards to the generation of $\boldsymbol{\xi}$ is on the need for $\xi_1 > 0$. Compliance of this can be obtained by limiting the solution space to the positive ξ_1 hemisphere. Further limitations on the rotation matrix however is difficult due to the complicated shape of the hypersphere region comprising the solution space. We may also be able to improve our acceptance rate of proposed points by incorporating prior information on the Ξ. This prior information may include expected chemical components, if they are known, or more information regarding the possible spectra and concentrations generated from the eigenvectors.

Our final area of research into spectral decomposition was to propose an improved spectral decomposition algorithm in Chapter 7. For this algorithm we began with methods of cluster analysis for isolating the subject of the scan from the background. In testing k-means and expectation-maximisation for a Gaussian mixture model we found EMGM to be the most accurate for our purposes as it was able to detect border pixels where the laser may have impacted both the subject and the background. The first method we proposed was a two stage process involving clustering the data then fitting a bipartite model to each cluster separately. With this method performing well at returning clearer spectrum estimates we combined the two steps. By modifying the Gaussian mixture model, to incorporate the

bilinear model into the means, we calculated the expectation-maximisation steps to form our new algorithm. In practice the spectra returned were similar to those from the two stage process, with less noise in the estimated spectra when compared to the standard algorithm. The combined algorithm had the benefits of a shorter computation time and requiring less user interaction as the algorithm could be initialised with no visual identification of clusters representing sample and background.

The algorithm we propose has similarities with algorithms known as unique resolution methods, §2.5, where regions of the Raman image are exploited to provide clearer estimated solutions. The regions these unique resolution methods use contain zero concentrations for certain component chemicals [71, 72] or are generated by only a single component chemical [59]. In the former case zero concentrations of specific chemicals allow r to be reduced and a separate analysis to be performed on that region with the idea that estimating fewer component spectra will reduce the range of solutions possible from the solution space. The latter case is a more extreme version of the former in that $r = 1$ for the region generated by a single component. They therefore require specific features in the Raman image and a user experienced in analysing spectroscopy data [53]. In comparison our method requires no user interaction or advanced knowledge to obtain clearer estimated spectra than standard least squares based methods. Results may be further improved if the algorithm were modified to autonomously detect zero concentration regions and separately analyse these as in the aforementioned methods. We must also consider the non-identifiability still present in our model which we explore in Chapter 4. Whilst this will still need to be accounted for in any testing we perform on results due to only the scan subject being analysed we aim to have smaller errors in our model leading to less variable solutions and therefore a smaller solution space.

One area for future work around our EM based method would be a proof of convergence for the standard estimates of $\mathbf{S}$ and $\mathbf{C}$ and in the case where the NNLS algorithm was used in step 5 of algorithm 7.3.1. With the standard estimates being iteratively estimated to achieve convergence and the NNLS algorithm solving $\mathbf{y}_i = \mathbf{c}_i\mathbf{S}^T$ however constrained such that $c_{i,r} > 0$ there is no guarantee the EM algorithm will converge as we are not simultaneously maximising the bilinear Gaussian model parameters. Further to this the initialisation of the mixture weights may require investigation in the case where the summation of observations provides little information on the clusters, for example if applied to data other than Raman images. There is also potential for future research into a clustering method using prior knowledge on the scan subject, in the case of our data this prior knowledge may incorporate the known circular shape of the tablet and the central position in the image.

8.2 Analysing the Results of Spectral Decomposition

Our testing of spectral decomposition results in Chapter 5 involved both Hotelling's T^2 test statistic for sample means and new test statistics we proposed for quantifying how randomly mixed pixels are in an image. These tests were also combined with the random walk defined in Chapter 4 allowing us to incorporate some of the infinite range of solutions available to us.

We began by using the standard T^2 test for sample means on concentrations from a single solution to (1.2.3) allowing us to compare regions of a single image or compare two separate images. Whilst this conventional test returned sensible results, in applying it to multiple solutions to our bilinear model we saw a range of p-values sometimes crossing 0.05, the 95% rejection region of the test statistic, giving us conflicting conclusions to our hypothesis test. We also had to take care that the concentrations being compared were either independent samples drawn from multivariate normal distributions with equal covariance matrices or were sufficiently large to asymptotically follow this distribution. This would ensure the assumptions of the T^2 test held and so to check this assumption we compared the test results from using tabled distribution values to those obtained from a bootstrap test. Whilst the results from both the conventional test and bootstrap tests concurred in their conclusions we could not guarantee that for all datasets the assumptions of the Hotelling's T^2 test would be met. We therefore proposed an algorithm combining our bootstrap test and random walk to both remove concerns over the assumptions of the test statistic and to account for non-identifiability of the model.

Our second area of testing was to investigate the mixing in the component chemicals and identify instances where the components were poorly mixed. Current methods for testing mixing quality are either performed using substitute component chemicals of different colours to test the mixing apparatus or use spectroscopy however focus on the estimated spectra. The methods of Koc et al. [58] and Coënt et al. [21] are of the former type where the apparatus is set to mix two dyed compounds and photographs of the process are taken. These are then converted to greyscale with the number of pixels of each level counted to quantify the mixing. The method of Lee and Lin [65] is more similar to ours in that it uses spectra repeatedly estimated during the mixing process. The intensities of these spectra are then compared across several regions of the Raman image to assess the quality of mixing. However these spectra will only indicate the presence of component spectra in the image region, not how thoroughly mixed they are, as the estimated spectra are shared across all pixels of an image and so provide no spatial information.

To devise a method taking into account both the magnitudes of the estimated concen-

trations and their locations we took the Minkowski functionals, grey-level co-occurrence matrices and correlations between neighbouring columns of an image to form three test statistics for quantifying the mixing. To narrow down our test statistics to the most reliable we simulated two sets of data to represent poorly mixed images and perfectly mixed images to which we would apply our proposed statistics. The first investigation of our test statistics involved estimating Type I and II error rates which indicated that the best performing were the neighbour-correlation-based test statistics, specifically F_{diff}, (5.4.17). With this test statistic applied in a bootstrap test combined with the random walk from Chapter 4 we compared how thoroughly mixed the component chemicals were in tablets from two production methods. For our datasets we concluded that the older crystalline-mixing production method provided a more thoroughly mixed product, however with only three tablets from each a larger sample size would be beneficial as the three amorphous tablets may have been from a development run and so not representative of amorphously produced tablets.

In comparison to the most similar current method of Lee and Lin [65] our test statistic more thoroughly tests the mixing of components as we test the concentrations in place of the estimated spectra which contain no spatial data. However both methods are susceptible to the non-identifiability of estimated solutions and must therefore be combined with our random walk to provide stable solutions. The importance of this is demonstrated in §5.3.1 and figure 5.2b where the same null hypothesis was both accepted and rejected for multiple solutions from the solution space. However incorporating this random walk greatly increases computation time and so future work in this area is tied in with that of Chapter 4 in improving the speed of the solution space exploration.

Future testing using the random walk from Chapter 4 could involve the point rankings we are able to calculate during step 6 of algorithm 4.6.1. With these points we would be able to weight our selection of new spectra from the solution space to use those which more often meet the non-negative concentration constraint more often in step 6. Using these points will be preferable to a random selection of points as, whilst a least squares optimisation method may find a local minima containing some negative concentrations, we will be able to avoid this. There is also scope for future research into the area of testing the estimated component spectra. This may be possible using the distributions of the eigenvectors from which we form our estimated spectra as a linear combination. These asymptotic distributions were derived by Anderson [5, 6] with the main distribution results from Anderson [6] summarised in Mardia et al. [73, Theorem 8.3.3]. With the asymptotic distributions of estimated spectra we may be able to avoid the need for the solution space as we can sample new spectra which will greatly reduce computation time. We will then be able to perform hypothesis tests for the presence of component chemicals by comparing

estimated spectra to reference spectra of chemicals we expect to find.

Our final area of testing is specific to Raman images gathered over time as the subject changes or dissolves. The current pharmaceutical industry standard for testing dissolution is to dissolve a produced tablet [8] in human body conditions. The flow rate of the dissolving tablet is then monitored to ensure the tablet dissolves at a steady rate. However in using spectroscopy to monitor the dissolution not only can the rate of dissolution be monitored, using the increase and decrease in chemical concentrations, but the spread of the individual component chemicals can also be investigated. With this scope for improved methods of dissolution testing there have been some published methods involving spectroscopy. In Chapter 6 we propose our own method for mapping and predicting the times at which chemical levels change as a subject dissolves. Our most basic method of analysis is identifying the maximum chemical concentration at each pixel and time point and and counting how many pixels are assigned to each spectra. Whilst simple to calculate, and represent in the form of a histogram, this method will suffer as the concentrations of only one component are taken into consideration. To improve upon this we defined a simple two part model based on the Gaussian cumulative distribution function. This model is applicable to datasets where one chemical decreases and another increases. We also had to add penalty terms to correct anomalous chemical changes, for example caused by the flow of the subject about the scan region. With this model we successfully mapped the chemical levels in our caffeine dataset as the tablet dissolved allowing us to estimate the process between the discrete scan times.

We then proposed a more complicated model capable of fitting chemical concentrations which increased then decreased. To achieve this we combined the separate parts of our previous model to form a continuous curve capable of fitting to more of dissolution pattern than a simple increase or decrease. Applying this model to our bicalutamide dataset proved effective at predicting the times at which chemical levels changed and allowed us to estimate the level change between the scan times. We also recreated the histogram from the basic analysis detailed previously and obtained a far clearer representation of the pattern with which the chemical components changed over time.

In comparing our model to current methods for analysing dissolving subjects with spectroscopy we see improvements in most areas. Where the method of Kazarian and van der Weerd [56] detects the edge of the dissolving subject to map the dissolution pattern our model identifies the time at which each individual pixel changes, giving a more complete picture of the process. Although the visual inspection of heatmaps used by Windbergs et al. [100] was possible for our Caffeine dataset the more complicated bicalutamide data was more difficult to interpret in such a manner, therefore our single heatmap showing the

change times is a more accessible format. Finally we saw the method of calculating the number of pixels dominated by each component chemical [31] give ambiguous results with the bicalutamide data in figure 6.5. In using the results from our model the clarity of these results improved to the point where the dissolution process was easily distinguishable. Our method however is not perfect and there is potential for future work in making the model applicable to more patterns of concentration change in place of the "S" shaped processes on which we focused.

Appendix A

Numerical Optimisation

Numerical optimisation is the minimisation or maximisation of a function, often subject to constraints on the variables. Numerical methods are particularly useful when an analytic solution is difficult, or impossible, to find and can provide a close estimate to the true solution [77]. We have a wide range of numerical optimisation methods available, each suited to a particular type of problem or model. We may optimise functions subject to constraints using either strict or soft penalties. The trust region, active set and quadratic penalty methods will feature in this thesis so we will describe their processes in the following sections.

A.1 Notation and Definitions

For unconstrained optimisation the problem is usually written in the form

$$\min_{x \in \mathbb{R}^n} f(\mathbf{x}) \tag{A.1.1}$$

and for constrained optimisation the convention is

$$\min_{x \in \mathbb{R}^n} f(\mathbf{x}) \qquad \text{subject to} \qquad \begin{array}{ll} c_i(\mathbf{x}) = 0, & i \in \mathcal{E} \\ c_i(\mathbf{x}) \geq 0, & i \in \mathcal{I} \end{array} . \tag{A.1.2}$$

where

- f is the objective function.
- $\mathbf{x} \in \mathbb{R}^n$ is a vector of parameters.
- c_i are constraint functions that define equations and inequalities to be satisfied by $\mathbf{x}$.
- $\mathcal{E}$ and $\mathcal{I}$ are the indices for equality and inequality constraints respectively.

There are also more specific functions like quadratic functions

$$\min_x q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x} + \mathbf{x}^T\mathbf{c} \quad \text{subject to} \begin{cases} \mathbf{a}_i^T\mathbf{x} = b_i, & i \in \mathcal{E}, \\ \mathbf{a}_i^T\mathbf{x} \geq b_i, & i \in \mathcal{I}, \end{cases} \tag{A.1.3}$$

where $\mathbf{c}, \mathbf{a}_i \in \mathbb{R}^n$, $\mathbf{G}$ is an $n \times n$ matrix for a quadratic function.

Most optimisation methods are iterative where the value of $\mathbf{x}$ at the k^{th} iteration is denoted $\mathbf{x}_k$. For simplicity we often drop the iteration subscript k giving

$$f_k = f(\mathbf{x}_k)$$

that is f_k is the value of f evaluated at $\mathbf{x}_k$ [77].

A.2 Active Set Methods

Active set methods use sets containing the active and inactive constraints and are applied to constrained quadratic problems of the form

$$\min_x q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x} + \mathbf{x}^T\mathbf{c} \quad \text{subject to} \begin{cases} \mathbf{a}_i^T\mathbf{x} = b_i, & i \in \mathcal{E}, \\ \mathbf{a}_i^T\mathbf{x} \geq b_i, & i \in \mathcal{I}, \end{cases} \tag{A.2.1}$$

where $\mathcal{E}$ and $\mathcal{I}$ are sets of indices for which the constraints are equalities and inequalities respectively. If the active set of constraints, $\mathcal{A}(\mathbf{x}) = \left\{i \in \mathcal{E} \cup \mathcal{I} | a_i^T\mathbf{x} = b_i\right\}$, were known then this would simply be a matter of solving the equality constrained quadratic problem

$$\min_x q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x} + \mathbf{x}^T\mathbf{c} \quad \text{subject to } \mathbf{a}_i^T\mathbf{x} = b_i,\ i \in \mathcal{A}(\mathbf{x})$$

for the solution $\mathbf{x}^*$. However finding this optimal active set of constraints is often difficult. The active set method starts with an estimate of $\mathcal{A}$ known as the working set, $\mathcal{W}_k$. This set consists of the indices of the equality constraints and those inequality constraints to be imposed as equalities. Now we define a subproblem to be solved at each iteration as

$$\min_p \frac{1}{2}\mathbf{p}^T\mathbf{G}\mathbf{p} + \mathbf{g}_k^T\mathbf{p} \quad \text{subject to } \mathbf{a}_i^T\mathbf{p} = 0,\ i \in \mathcal{W}_k$$

where $\mathbf{p} = \mathbf{x} - \mathbf{x}_k$ and $\mathbf{g}_k = \mathbf{G}\mathbf{x}_k + \mathbf{c}$. We now have two cases, if $\mathbf{p} = 0$ we compute a system of Lagrange multipliers that satisfy

$$\sum_{i \in \mathcal{W}_k} \mathbf{a}_i\hat{\lambda}_i = \mathbf{g}_k = \mathbf{G}\mathbf{x}_k + \mathbf{c}.$$

If $\hat{\lambda}_i \geq 0$ for all $i \in \mathcal{W}_k \cap \mathcal{I}$ we halt the algorithm with the solution $\mathbf{x}^* = \mathbf{x}_k$. If the algorithm is not halted we set $\mathbf{x}_{k+1} = \mathbf{x}_k$ and modify $\mathcal{W}$ by removing the constraint that minimises $\hat{\lambda}_j$ for $j \in \mathcal{W}_k \cap \mathcal{I}$.

If $\mathbf{p} \neq 0$ then $\mathbf{p}_k$ can be thought of as the step direction towards $\mathbf{x}_{k+1}$ where we calculate the step length α_k using

$$\alpha_k = \min\left(1, \min_{i \notin \mathcal{W}_k, \mathbf{a}_i^T \mathbf{p}_k < 0} \frac{b_i - \mathbf{a}_i^T \mathbf{x}_k}{\mathbf{a}_i^T \mathbf{p}_k}\right). \tag{A.2.2}$$

With (A.2.2) we update $\mathbf{x}$ with $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$. If $\alpha_k = 1$ then the step is uninhibited and there are no inactive constraints blocking the minimisation. However if $\alpha_k < 1$ then the step along $\mathbf{p}_k$ was blocked by some constraint(s) not in $\mathcal{W}_k$. We identify the blocking inactive constraint by the i for which the minimum was found in (A.2.2). In this case we calculate $\mathbf{x}_{k+1}$ as before however now the blocking constraint(s) are added to $\mathcal{W}_{k+1}$. We repeat this process until an optimum is found with the proof of convergence shown by Nocedal and Wright [77].

A.3 Trust-Region

Trust region optimisation was first proposed by Winfield [103] as a method of solving (A.1.1). The theory behind a trust-region is to use information about f to construct a model function, m, which behaves similarly to f near the current solution iterate $\mathbf{x}_k$. We then restrict the search region around $\mathbf{x}_k$ using a candidate step, p, as m may not behave like f far from $\mathbf{x}_k$.

The model m_k is usually defined to be a quadratic function of the form

$$m_k(\mathbf{p}) = f_k + \mathbf{p}^T \nabla f_k + \frac{1}{2}\mathbf{p}^T \mathbf{B}_k \mathbf{p} \tag{A.3.1}$$

where $\mathbf{B}_k$ is taken to be either the Hessian, $\nabla^2 f_k$, or some approximation to it. These two choices lead to trust-region Newton and quasi-Newton methods respectively [77].

The basic outline of a trust-region approach to optimisation is to first define our trust region about the model function m. With our trust region selected we calculate the step direction $\mathbf{p}$ by solving

$$\min_p m_k(\mathbf{p})$$

where $\mathbf{x}_k + \mathbf{p}$ lies inside the trust-region.

A.3.1 Trust Region Size

We select the size of the trust region, Δ_k, for the next iteration using the ratio

$$\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{p}_k)}{m_k(0) - m_k(\mathbf{p}_k)} \tag{A.3.2}$$

This ratio will be close to 1 if there is a good agreement between the model m_k and the function f over the current step. If this optimal ratio occurs it is safe to expand the region size, Δ_k, for the next iteration. However if the ratio is close to 0 or negative the region size is reduced at the next iteration. A common algorithm using this ratio sets the region size at the next iteration as

$$\Delta_{k+1} = \begin{cases} \frac{1}{4}\Delta_k & \text{if } \rho_k < \frac{1}{4} \\ \min(2\Delta_k, \hat{\Delta}) & \text{if } \rho_k > \frac{3}{4} \text{ and } \|\mathbf{p}_k\| = \Delta_k \\ \Delta_k & \text{otherwise} \end{cases}$$

for region bound $\hat{\Delta} > 0$, initial region size $\Delta_0 \in (0, \hat{\Delta}]$ and Euclidean norm $\|\cdot\|$. We also use ratio (A.3.2) to decide whether to progress from $\mathbf{x}_k$ using the conditions

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{p}_k & \text{if } \rho_k > \eta \\ \mathbf{x}_k & \text{otherwise} \end{cases}$$

for $\eta \in \left[0, \frac{1}{4}\right)$. This algorithm is studied in greater detail by Powell [81], with the conclusion that with $\eta = 0$ the trust-region always takes a step if f decreases.

A.3.2 Step Direction

With our region size, Δ_k, calculated we choose the step direction within that region using one of a number of techniques. These techniques are all based on solving the subproblem

$$\min_{\mathbf{p}\in\mathbb{R}^n} m_k(\mathbf{p}) = f_k + \mathbf{g}_k^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T\mathbf{B}_k\mathbf{p}$$

which often has the iteration subscript k removed for convenience to give

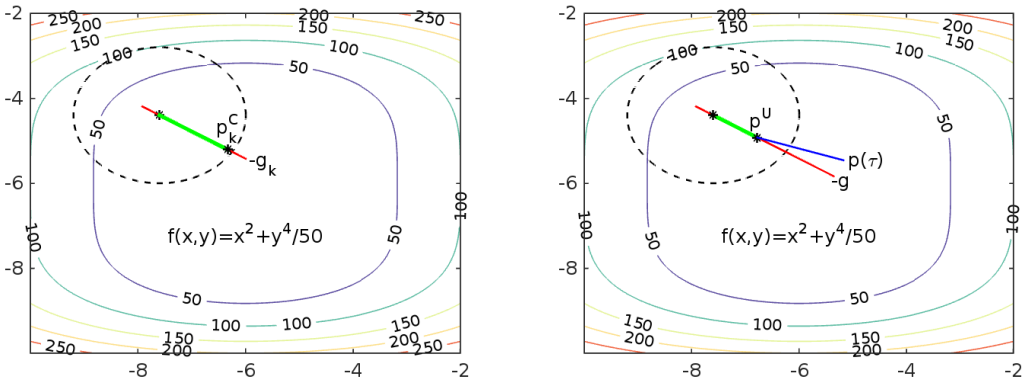
$$\min_{\mathbf{p}\in\mathbb{R}^n} m(\mathbf{p}) = f + \mathbf{g}^T\mathbf{p} + \frac{1}{2}\mathbf{p}^T\mathbf{B}\mathbf{p}$$

where $\mathbf{g}_k = \nabla f(\mathbf{x}_k) = \mathbf{g}$ and $\mathbf{B}$ is the Hessian or an estimate.

A.3.3 Step Direction: Cauchy Point

The Cauchy point algorithm uses the steepest descent direction of m_k, denoted $-\mathbf{g}_k$, which we write as the following two minimisation problems

$$\begin{aligned} \mathbf{p}_k^s &= \arg\min_{\mathbf{p}\in\mathbb{R}^n} f_k + \mathbf{g}_k^T\mathbf{p} && \text{for } \|\mathbf{p}\| \le \Delta_k \\ \tau_k^s &= \arg\min_{\tau\ge 0} m_k(\tau\mathbf{p}_k^s) && \text{for } \|\tau\mathbf{p}_k^s\| \le \Delta_k. \end{aligned}$$



(a) The Cauchy point, appendix A.3.3.

(b) The dogleg approximation, appendix A.3.4.

Figure A.1: Contour plots of $f(x, y) = x^2 + \frac{y^4}{50}$ with region size $\Delta_k = 8$ and the trust region marked as a dashed ellipse.

We now define $\mathbf{p}_k^C = \tau_k \mathbf{p}_k^s$ as the Cauchy point or search direction. Closed-form solutions to these are

$$\mathbf{p}_k^c = -\tau_k \frac{\Delta_k}{\|\mathbf{g}_k\|} \mathbf{g}_k \tag{A.3.3}$$

$$\tau_k = \begin{cases} 1 & \text{if } \mathbf{g}_k^T \mathbf{B}_k \mathbf{g}_k \leq 0 \\ \min\left(\frac{\|g_k\|^3}{(\Delta_k g_k^T B_k g_k)}, 1\right) & \text{otherwise.} \end{cases} \tag{A.3.4}$$

The step calculated by this method is inexpensive to calculate, as no matrix factorisations are required, and always lies within the trust region. However this method often performs poorly by achieving the worst case bound of the solution [3]. Therefore modified methods based on the Cauchy point were developed. We see an example of the Cauchy point in figure A.1a.

A.3.4 Step Direction: Dogleg Method

The dogleg method uses two line segments to select the step to take inside the trust region. The first of these segments runs from the origin along the steepest descent direction, $-\mathbf{g}$, the same direction used by the Cauchy point, and is defined as

$$\mathbf{p}^U = -\frac{\mathbf{g}^T \mathbf{g}}{\mathbf{g}^T \mathbf{B} \mathbf{g}} \mathbf{g}.$$

The second segment known as the full step follows the trajectory defined by

$$\tilde{\mathbf{p}}(\tau) = \begin{cases} \tau \mathbf{p}^U & 0 \leq \tau \leq 1, \\ \mathbf{p}^U + (\tau - 1)(\mathbf{p}^B - \mathbf{p}^U) & 1 \leq \tau \leq 2. \end{cases}$$

where $\mathbf{p}^B$ is the unconstrained minimiser of m given by $-\mathbf{B}^{-1}\mathbf{g}$.

In the case that the unconstrained minimiser lies within the trust region, $\|\mathbf{p}^B\| \leq \Delta$, this is the optimal value of $\mathbf{p}$. However this is unlikely for all but the step attaining the minimum, therefore we take the appropriate value of $\mathbf{p}$ to be the point of intersection between the dogleg and the trust-region boundary. In figure A.1b this is the intersection between the line $\mathbf{p}(\tau)$ and the dashed ellipse. We fine the appropriate value of τ by solving the quadratic

$$\|\mathbf{p}^U + (\tau - 1)(\mathbf{p}^B - \mathbf{p}^U)\|^2 = \Delta^2. \tag{A.3.5}$$

More detailed discussions on both the Cauchy point and Dogleg methods can be found in the works of Nocedal and Wright [77] and Dennis and Schnabel [27].

A.4 Quadratic Penalty Method

Penalty methods are a form of constrained optimisation where we modify the original problem with a penalty function. This function includes both the original minimisation problem plus a term for each constraint which is positive when the constraint is violated. These methods work by increasing the objective function when the constraints are not met so that not only must the original function be minimised but the constraints are also satisfied. Unlike the active set method the quadratic penalty method does not strictly impose the constraints. Depending on the penalty parameter used the minimum possible value of the function may only partially satisfy them.

Given an equality-constrained problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to } c_i(\mathbf{x}) = 0,\ i \in \mathcal{E}, \tag{A.4.1}$$

we define the quadratic penalty function, first proposed by Courant [22], as

$$Q(\mathbf{x}; \mu) = f(\mathbf{x}) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i^2(\mathbf{x}),$$

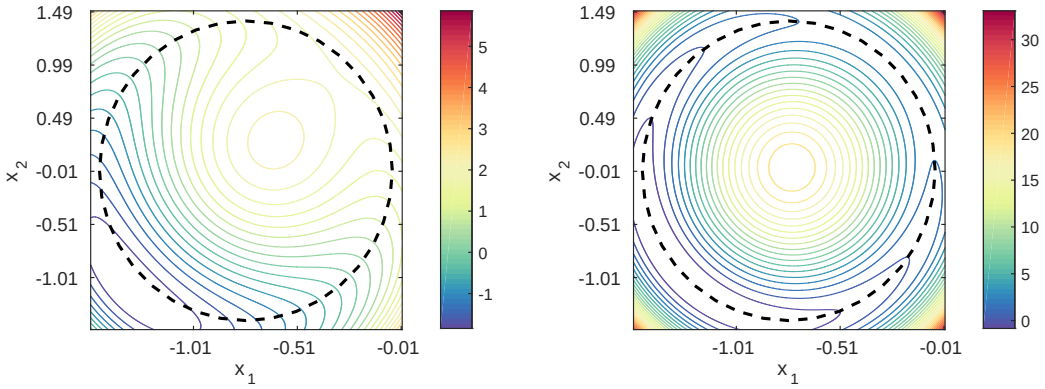
where $\mu > 0$ is the penalty parameter controlling the severity of constraint violations. We can also define this for a general constrained optimisation problem of the form

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to } \begin{cases} c_i(\mathbf{x}) = 0, & i \in \mathcal{E}, \\ c_i(\mathbf{x}) \geq 0, & i \in \mathcal{I}, \end{cases}$$

where $\mathcal{E}$ and $\mathcal{I}$ are sets of indices for which the constraints are equalities and inequalities respectively. We now define Q as

$$Q(\mathbf{x}; \mu) = f(\mathbf{x}) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i^2(\mathbf{x}) + \frac{1}{2\mu} \sum_{i \in \mathcal{I}} \left([c_i(\mathbf{x})]^-\right)^2,$$

where $[c_i(\mathbf{x})]^- = \max(-c_i(\mathbf{x}), 0)$. The algorithm for using these formulae is



(a) Contour plot of $Q(\mathbf{x};\mu)$ with $\mu = 1$, minimum at (-1,-1).

(b) Contour plot of $Q(\mathbf{x};\mu)$ with $\mu = 10$.

Figure A.2: Contour plots of $Q(\mathbf{x};\mu)$, for problem (A.4.3), with 30 contour levels and the constraint plot as a dashed ellipse.

Algorithm A.4.1. Quadratic Penalty Method

1. Choose starting values $\mu_0 > 0$, tolerance $\tau_0 > 0$ and starting point $\mathbf{x}_0$.
2. Find an approximate minimiser, $\mathbf{x}_k$, of $Q(\mathbf{x};\mu_k)$.
3. Terminate if $|\nabla Q(\mathbf{x}_k;\mu_k)| \leq \tau_k$ otherwise set $\mu_{k+1} \in (0,\mu_k)$, choose a new $\mathbf{x}_{k+1}$ then return to step 2.

The new point $\mathbf{x}_{k+1}$ can be calculated using any unconstrained minimisation technique such as the trust region method described in appendix A.3. At each iteration we adapt μ_{k+1} based on how expensive the minimisation of Q was. For example if the minimisation was expensive we use $\mu_{k+1} = 1.5\mu_k$, however if the minimisation was cheap to compute a larger reduction may prove possible such as $\mu_{k+1} = 10\mu_k$.

The drawbacks of this method include the difficulty of minimising Q when μ_k becomes small and the ill conditioning of $\nabla Q(\mathbf{x};\mu_k)$ close to the optimum where a small change in $\mathbf{x}$ may produce a large change in the function. An algorithm to avoid problems caused by ill conditioning was proposed by Gould [40] however because of the drawbacks augmented Lagrangian methods [77] are commonly used instead.

A simple example of a quadratic penalty method is the function

$$\min\left(x_1 + x_2\right) \qquad \text{subject to } x_1^2 + x_2^2 - 2 = 0 \tag{A.4.2}$$

which gives

$$Q(\mathbf{x};\mu) = x_1 + x_2 + \frac{1}{2\mu}\left(x_1^2 + x_2^2 - 2\right)^2. \tag{A.4.3}$$

We see how different values for μ effect the optimisation in figure A.2. In figure A.2a we see the contours of (A.4.3) are only slightly effected by the circular constraint and the minimisation of $(x_1 + x_2)$ is still taking priority. However for $\mu = 10$ we now see the contours of (A.4.3) are largely dominated by the constraint of $x_1^2 + x_2^2 - 2 = 0$ in favour of minimising $(x_1 + x_2)$. Therefore the true solution of (-1,-1) may not be found in favour of another point on the ellipse.

References

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2:433–459, 2010.

[2] H. J. Adèr and G. J. Mellenbergh. *Advising on research methods: A consultant's companion*. Johannes van Kessel Publishing, 2008.

[3] H. Akaike. On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Annals of the Institute of Statistical Mathematics*, 11:1–16, 1959.

[4] F. Albregtsen. Statistical texture measures computed from gray level co-occurrence matrices. Technical report, Department of Informatics, University of Oslo, Norway, 1995.

[5] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, 1958.

[6] T. W. Anderson. Asymptotic theory for principal component analysis. *Annals of Mathematical Statistics*, 34:122–148, 1963.

[7] F. Aurenhammer. Voronoi diagrams - a survey of fundamental geometric data structure. *ACM Computing Surveys*, 23:345–405, 1991.

[8] G. Bai and Y. Wang. Velocity profiles and shear strain rate variability in the USP dissolution testing apparatus 2 at different impeller agitation speeds. *Journal of Pharmaceutics*, 403:1–14, 2011.

[9] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52:155–173, 2007.

[10] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[11] J. A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report ICSI-TR-97-021, International Computer Science Institute, 1998.

[12] O. S. Borgen and B. R. Kowalski. An extension of the multivariate component-resolution method to three components. *Analytica Chimica Acta*, 174: 1–26, 1985.

[13] B. T. Bowie, D. B. Chase, I. R. Lewis, and P. R. Griffiths. *Handbook of Vibrational Spectroscopy*, chapter Anomalies and Artifacts in Raman Spectroscopy. John Wiley & Sons, Ltd, 2006.

[14] R. Bro and S. De Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11:393–401, 1997.

[15] A. D. Burnett, H. G. M. Edwards, M. D. Hargreaves, T. Munshi, and K. Page. A forensic case study: the detection of contraband drugs in carrier solutions by Raman spectroscopy. *Drug Testing and Analysis*, 3:539–543, 2010.

[16] J. Cadima and I. Jolliffe. On relationships between uncentred and column-centred principal component analysis. *Pakistan Journal of Statistics*, 25:473–503, 2009.

[17] N. A. Campbell. Robust procedures in multivariate analysis I: Robust covariance estimation. *Applied Statistics*, 29:231–237, 1980.

[18] K. L. A. Chan and S. G. Kazarian. FT-IR spectroscopic imaging of reactions in multiphase flow in microfluidic channels. *Analytical Chemistry*, 84:4052–4056, 2012.

[19] D. Chang, C. D. Bannack, and S. L. Shah. Robust baseline correction algorithm for signal dense NMR spectra. *Journal of Magnetic Resonance*, 187:288–292, 2007.

[20] J. C. Cobas, M. A. Bernstein, M. Martín-Pastor, and P. G. Tahoces. A new general-purpose fully automatic baseline-correction procedure for 1d and 2d NMR data. *Journal of Magnetic Resonance*, 183:145–151, 2006.

[21] A. Le Coënt, A. Rivoire, S. Briançon, and J. Lieto. An original image-processing technique for obtaining the mixing time: The box-counting with erosions method. *Powder Technology*, 152:62–71, 152.

[22] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. Bulletin of the American Mathematical Society, 49, 1943.

[23] M. W. Crofton. On the theory of local probability, applied to straight lines drawn at random in a plane; the methods used being also extended to the proof of certain new theorems in the integral calculus. *Philosophical Transactions of the Royal Society of London*, 158:181–199, 1868.

[24] A.C. Davison and D.V. Hinkley. *Bootstrap Methods and their Application*. Cambridge University Press, 1997.

[25] F. B. de Carvalho, A. F. S. Barbosa, F. A. A., A. B. Júnior, L. S. Júnior, and A. L. B. Pinheiro. Use of laser fluorescence in dental caries diagnosis: a fluorescence x biomolecular vibrational spectroscopic comparative study. *Brazilian Dental Journal*, 24:59–63, 2013.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1–38, 1977.

[27] J. E. Jr. Dennis and Robert B. Schnabel. *Handbooks in Operations Research and Management Science, Vol. 1, Optimisation*, chapter A View of Unconstrained Optimisation, pages 1–72. Elsevier Science Publishers, 1989.

[28] W. J. Dixon. Simplified estimation from censored normal samples. *The Annals of Mathematical Statistics*, 31:385–391, 1960.

[29] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7:1–26, 1979.

[30] L. Euler. Elementa doctrinae solidorum. *Novi Commentarii Academiae Scientarum Petropolitanae*, pages 109–140, 1758.

[31] F. Tres, University of Nottingham, *School of Pharmacy*. Unpublished paper.

[32] J. R. Ferraro, K. Nakamoto, and C. W. Brown. *Introductory Raman Spectroscopy*. Academic Press, 2002.

[33] R. A. Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large popularion. *Biometrika*, 10:507–521, 1915.

[34] R. A. Fisher. On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron*, 1:3–32, 1921.

[35] M. Garrido, M. S. Larrechi, F. X. Rius, and R. Tauler. Calculation of band boundaries of feasible solutions obtained by Multivariate Curve

Resolution-Alternating Least Squares of multiple runs of a reaction monitored by NIR spectroscopy. *Chemometrics and Intelligent Laboratory Systems*, 76:111–120, 2005.

[36] P. Geladi. Chemometrics in spectroscopy. Part 1. Classical chemometrics. *Spectrochimica Acta Part B: Atomic Spectroscopy*, 58:767–782, 2003.

[37] D. L. Gerrard and H. J. Bowley. *Instrumentation for Raman Spectroscopy*. Springer, 1989.

[38] F. Gilchrist, A. Santini, K. Harley, and C. Deery. The use of micro-Raman spectroscopy to differentiate between sound and eroded primary enamel. *International Journal of Paediatric Dentistry*, 17:274–280, 2007.

[39] E. F. Gonzales and Y. Zhang. Accelerating the Lee-Seung algorithm for non-negative matrix factorization. Technical report, Dept. Comput. & Appl. Math., Rice Univ., Houston, TX, Tech. Rep. TR-05-02, 2005.

[40] N. I. M. Gould. On the accurate determination of search directions for simple differentiable penalty functions. *IMA Journal of Numerical Analysis*, 6:357–372, 1986.

[41] R. Guerra, A. M. Polansky, and W. R. Schucany. Bootstrap with kernel smoothing for discrete data. Dallas: Department of Statistical Science, Southern Methodist University.

[42] R. Guerra, A. M. Polansky, and W. R. Schucany. Smoothed bootstrap confidence intervals with discrete data. *Computational Statistics & Data Analysis*, 26:163–176, 1997.

[43] P. Güntert and K. Wüthrich. FLATT- A new procedure for high-quality baseline correction of multidimensional NMR spectra. *Journal of Magnetic Resonance*, 96: 403–407, 1992.

[44] P. Hall and S. R. Wilson. Two guidelines for bootstrap hypothesis testing. *Biometrics*, 47:757–762, 1991.

[45] G. Hamerly and C. Elkan. Alternatives to the *k*-means algorithm that find better clusterings. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 600–607. ACM, 2002.

[46] R. M. Haralick, S. R. Sternerg, and X. Zhuang. Image analysis using mathematical morphology. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 9: 532–550, 1987.

[47] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[48] H. Hotelling. The generalization of Student's ratio. *Annals of Mathematical Statistics*, 2:360–378, 1931.

[49] H. Hung, P. Wu, I. Tu, and S. Huang. On multilinear principal component analysis of order-two tensors. *Biometrika*, 99:569–583, 2012.

[50] J. Burley, University of Nottingham, *School of Pharmacy*. Personal communication.

[51] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, 1999. ISSN 0360-0300.

[52] J.-H. Jiang, Y. Z. Liang, and Y. Ozaki. On simplex-based method for self-modeling curve resolution of two-way data. *Chemometrics and Intelligent Laboratory Systems*, 65:51–65, 2003.

[53] J.-H. Jiang, Y. Liang, and Y. Ozaki. Principles and methodologies in self-modeling curve resolution. *Chemometrics and Intelligent Laboratory Systems*, 71:1–12, 2004.

[54] M. Kac. *Probability and Related Topics in Physical Sciences*. Interscience Publishers, 1959.

[55] E. J. Karjalainen. The spectrum reconstruction problem. *Chemometrics and Intelligent Laboratory Systems*, 7:31–38, 1989.

[56] S.G. Kazarian and J. van der Weerd. Simultaneous FT-IR spectroscopic imaging and visible photography to monitor tablet dissolution and drug release. *Pharmaceutical Research*, 25:853–860, 2008.

[57] M. R. Keenan and P. G. Kotula. Accounting for Poisson noise in the multivariate analysis of ToF-SIMS spectrum images. *Surface and Interface Analysis*, 36: 203–212, 2004.

[58] A. B. Koc, H. Silleli, C. Koc, and M. A. Dayioglu. Monitoring of dry powder mixing with real-time image processing. *Journal of Applied Sciences*, 7:1218–1223, 2007.

[59] O. M. Kvalheim and Y. Z. Liang. Heuristic evolving latent projections: resolving two-way multicomponent data. 1. selectivity, latent-projective graph, datascope, local rank, and unique resolution. *Analytical Chemistry*, 64:936–946, 1992.

[60] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *Society for Industrial and Applied Mathematics*, 9:112–147, 1998.

[61] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.

[62] W. H. Lawton and E. A. Sylvestre. Self modeling curve resolution. *Technometrics*, 13:617–633, 1971.

[63] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[64] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13:556–562, 2001.

[65] T. Lee and S. Lin. Microspectroscopic FT-IR mapping system as a tool to assess blend homogeneity of drug-excipient mixtures. *European Journal of Pharmaceutical Sciences*, 23:117–122, 2004.

[66] D. Legland, K. Kiêu, and M.-F. Devaux. Computation of Minkowski measures on 2D and 3D binary images. *Image Analysis & Stereology*, 26:83–92, 2007.

[67] C.-J. Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *Neural Networks, IEEE Transactions on*, 18:1589–1596, 2007.

[68] C. J. Lloyd. A numerical investigation of the accuracy of parametric bootstrap for discrete data. *Computational Statistics & Data Analysis*, 61:1–6, 2013. ISSN 0167-9473.

[69] S. P. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28:129–137, 1982.

[70] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[71] M. Maeder. Evolving factor analysis for the resolution of overlapping chromatographic peaks. *Analytical Chemistry*, 59:527–530, 1987.

[72] E. R. Malinowski. Window factor analysis: Theoretical derivation and application to flow injection analysis data. *Journal of Chemometrics*, 6:29–40, 1992.

[73] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 2003.

[74] H. L. C. Meuzelaar, editor. *Computer-Enhanced Analytical Spectroscopy*, volume 2. Plenum Press, 1989.

[75] A. C. Moffat, M. D. Osselton, B. Widdop, and J. Watts, editors. *Clarke's Analysis of Drugs and Poisons*. Pharmaceutical Press, fourth edition, 2011.

[76] C. A. Murray and S. B. Dierker. Use of an unintensified charge-coupled device detector for low-light-level raman spectroscopy. *Journal of the Optical Society of America A*, 3:2151–2159, 1986.

[77] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, second edition, 2006.

[78] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, 2nd Edition*. Wiley, 2000.

[79] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[80] J. M. Peña, J. A. Lozano, and P. Larrañaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20: 1027–1040, 1999.

[81] M. J. D. Powell. *Nonlinear Programming 2*, chapter Convergence Properties of a Class of Minimization Algorithms, pages 1–27. Acadamic Press, 1975.

[82] Princeton Instruments. Raman spectroscopy basics. URL `http://content.piacton.com/Uploads/Princeton/Documents/Library/UpdatedLibrary/Raman_Spectroscopy_Basics.pdf`.

[83] W. Pych. A fast algorithm for cosmic-ray removal from single images. *Publications of the Astronomical Society of the Pacific*, 116:148–153, 2004.

[84] R. H. Randles. A simpler, affine-invariant, multivariate, distribution-free sign test. *Journal of the American Statistical Association*, 95:1263–1268, 2000.

[85] F. H. Roger, P. Grönroos, R. Tervo-Pellikka, and R. O'Moore, editors. *Medical Informatics Europe 85, Helsinki, Finland*, Berlin, 1985. Springer Verlag.

[86] F. C. Sánchez, J. Toft, B. van den Bogaerta, and D. L. Massart. Orthogonal projection approach applied to peak purity assessment. *Analytical Chemistry*, 68: 79–85, 1996.

[87] F. C. Sánchez, B. van den Bogaerta, S.C. Rutan, and D.L. Massart. Multivariate peak purity approaches. *Chemometrics and Intelligent Laboratory Systems*, 34: 139–171, 1996.

[88] K. Sandeman, P. Coombe, B. Barber, C. Best, and D. Brook. Raman scattering. URL `http://www.doitpoms.ac.uk/tlplib/raman/raman_scattering.php`.

[89] H. Shinzawa, K. Awa, W. Kanematsu, and Y. Ozaki. Multivariate data analysis for Raman spectroscopic imaging. *Journal of Raman Spectroscopy*, 40:1720–1725, 2009.

[90] D. Skillicorn. *Understanding Complex Datasets: Data Mining with Matrix Decompositions*. Chapman and Hall/CRC, 2007.

[91] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *The University of Kansas Science Bulletin*, 38:1409–1438, 1958.

[92] H. Spath. *Cluster Dissection and Analysis*. Halsted Press, 1985.

[93] C. J. Strachan, M. Windbergs, and H. L. Offerhaus. Pharmaceutical applications of non-linear imaging. *International Journal of Pharmaceutics*, 417:163–172, 2011.

[94] M. A. Tanner. *Tools for Statistical Inference*. Springer, 1996.

[95] R. Tauler. Calculation of maximum and minimum band boundaries of feasible solutions for species profiles obtained by multivariate curve resolution. *Journal of Chemometrics*, 15:627–646, 2001.

[96] R. Tauler and E. Casassas. Spectroscopic resolution of macromolecular complexes using factor analysis: Cu(II)-polyethyleneimine system. *Chemometrics and Intelligent Laboratory Systems*, 14:305–317, 1992.

[97] J.W. Tukey. The future of data analysis. *The Annals of Mathematical Statistics*, 33:17–18, 1962.

[98] P. G. van Dokkum. Cosmic-ray rejection by laplacian edge detection. *Publications of the Astronomical Society of the Pacific*, 113:1420–1427, 2001.

[99] G. Wernimont. Evaluating laboratory performance of spectrophotometers. *Analytical Chemistry*, 39:554–562, 1967.

[100] M. Windbergs, M. Jurna, H. L. Offerhaus, J. L. Herek, P. Kleinebudde, and C. J. Strachan. Chemical imaging of oral solid dosage forms and changes upon dissolution using coherent anti-Stokes Raman scattering microscopy. *Analytical Chemistry*, 81:2085–2091, 2009.

[101] M. Windbergs, M. Haaser, C. M. McGoverin, K. C. Gordon, P. Kleinebudde, and C. J. Strachan. Investigating the relationship between drug distribution in solid lipid matrices and dissolution behaviour using Raman spectroscopy and mapping. *Journal of Pharmaceutical Sciences*, 99:1464–1475, 2010.

[102] W. Windig and J. Guilment. Interactive self-modeling mixture analysis. *Analytical Chemistry*, 63:1425–1432, 1991.

[103] D. H. Winfield. *Function and Functional Optimization by Interpolation in Data Tables*. Harvard University, 1970.

[104] C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11:95–103, 1983.

[105] V. Zavala-Alonso, J. P. Loyola-Rodríguez, H. Terrones, N. Patiño-Marín, G. A. Martínez-Castañon, and K. Anusavice. Analysis of the molecular structure of human enamel with fluorosis using micro-Raman spectroscopy. *Journal of Oral Science*, 54:93–98, 2012.

[106] Z. M. Zhang, S. Chen, and Y. Z. Liang. Baseline correction using adaptive iteratively reweighted penalized least squares. *The Analyst*, 135:1138–1146, 2010.