UNIVERSITÉ DE MONTRÉAL

A STUDY ON SHAPE CLUSTERING AND ODOR PREDICTION

MINA MIRSHAHI
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(MATHÉMATIQUES DE L'INGÉNIEUR)
AOÛT 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

A STUDY ON SHAPE CLUSTERING AND ODOR PREDICTION

présentée par : MIRSHAHI Mina
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. LEFEBVRE Mario, Ph. D, président
M. LODI Andrea, Ph. D, membre et directeur de recherche
M. PARTOVI NIA Vahid, Ph. D, membre et codirecteur de recherche
M. ASGHARIAN Masoud, Ph. D, membre et codirecteur de recherche
Mme LABBE Aurélie, Ph. D, membre
M. YANG Yi, Ph. D, membre externe

DEDICATION

To My Parents

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my Ph.D advisers Dr. Vahid Partovi Nia and Dr. Masoud Asgharian, for their knowledge, motivation, patience, continuous support, and excellent mentorship during my Ph.D studies. I would also like to thank the examining committee for taking time out from their busy schedule to read and review my thesis and provide me with their insightful comments.

I must also acknowledge Dr. Luc Adjengue for providing me with the opportunity of a six-month internship program in Odotech, and his scientific collaboration during the course of the project. Furthermore, I would like to make special mention to Guy Laliberté for allowing me to partake in research in his company, Odotech, Éric Debeuf for his assistance and knowledge in electronic nose technology. Last but not least, I would also like to thank Mahdi Zolnouri for his teamwork and talent in programming.

I would also like to thank my parents, my sister Eliza and my brother Arash for being a continuous source of affection and love along the way. My special thanks go to Sultana, Azim, and Roumi, who have been like my second family in Montreal and have provided me with great love and support throughout my studies. I would like to thank them for all the joyful and unforgettable moments they brought to my life. I am especially grateful to Sultana, who has been a mentor beyond words, and who has been there whenever I needed her. She always has the time to hear about my successes and failures and has always been a faithful listener. Sultana and her family have always provided me a home away from home. I would like to also thank my true and trusted friend Mona Vahabpour for sharing such great moments together.

I would like to thank my friend Marek Niedbalski for helping me with building the R package related to my thesis and my wonderful friend Haafiz Alibhai for his editing assistance in general. I would also like to thank my dear friend Mouloud Belbahri for translating the abstract into French. Lastly but by no means least, I would also like to thank my friends, who have been very kind in sharing office with me during last four years.

RÉSUMÉ

La thése est divisée en deux parties principales. Dans la première partie, nous développons une nouvelle méthodologie et les outils computationnels nécessaires pour le regroupement ("clustering" en anglais) de formes. Dans la deuxième partie, nous abordons la problématique de prédiction des odeurs dans le secteur de la technologie du "nez électrique" ("e-nose" en anglais). Les chapitres 1 et 2 décrivent nos méthodologies proposées pour le regroupement de formes. Dans le chapitre 3, nous présentons une nouvelle approche pour la qualité des prédictions d'odeurs. Ensuite, nous exhibons un bref aperçu des deux problématiques, c'est-à-dire, le regroupement de formes et la prédiction d'odeurs, et nos solutions proposées.

1. **Regroupement de formes**: Les formes peuvent être interprétées comme des contours fermés dans un espace dimensionnel infini qui peut se transformer en différentes formes à travers le temps. Le principal objectif dans la modélisation de formes est de fournir un modèle mathématique qui représente chacune des formes. L'analyse statistique de formes est un outil très puissant dans l'étude des structures anatomiques des images médicales. Dans cette thése, qui est motivée principalement par les applications biologiques, nous suggérons une méthodologie pour la modélisation de surfaces des cellules. De plus, nous proposons une nouvelle technique de regroupement de formes de cellules. La méthodologie peut également être appliquée à d'autres objets géométriques.

   De nombreuses études ont été menées afin de suivre les possibles déformations des cellules à travers des descriptions qualitatives. Notre intérêt est plutôt de fournir une évaluation numérique précise des cellules. Dans le chapitre 1, des modèles statistiques utilisant différentes fonctions de base ("basis function" en anglais) sont ajustés afin de modéliser la surface des formes des cellules en 2 et 3 dimensions. Pour ce faire, la surface d'une cellule est d'abord convertie en un ensemble de données numériques. Par la suite, une courbe est ajustée à ces données. À ce stade, chaque cellule est représentée par une fonction continue. Maintenant, la question fondamentale est: comment distinguer différentes cellules en utilisant leurs formes fonctionnelles?

   Dans le chapitre 2, nous formulons un critère d'information bayésienne de regroupement ("clustering Bayesian information criterion" ou CLUSBIC en anglais) pour le regroupement hiérarchique de formes. Dans cette nouvelle approche, nous traitons les formes comme des courbes continues et nous calculons la fonction marginale postérieure associée à chaque courbe. Par conséquent, nous construisons le dendrogramme pour le regroupement hiérarchique en utilisant le CLUSBIC. Le dendrogramme est coupé lorsque la fonction marginale postérieure atteint son maximum.

Nous montrons au chapitre 2 que le CLUSBIC est une extension naturelle de la méthode de Ward, une mesure de regroupement bien connue. Comme le critère d'information bayésien (BIC) dans le cadre d'une régression, nous démontrons la cohérence du CLUSBIC dans le cadre du regroupement de données. Le CLUSBIC est une extension du BIC, qui coïncide avec le BIC si les données se regroupent dans un amas unique. L'utilité de notre méthodologie proposée dans la modélisation et le regroupement des formes est étudiée sur des données simulées ainsi que sur des données réelles.

2. **Prédiction d'odeurs**: Un "e-nose", ou olfaction artificielle, est un dispositif qui analyse l'air afin d'identifier les odeurs en utilisant un ensemble de capteurs de gaz. Le "e-nose" produit des données multidimensionnelles pour chaque mesure qu'il saisit du milieu environnant. Un petit sous-échantillon de ces mesures est envoyé à l'olfactométrie où les activités d'odeurs sont analysées. Dans l'olfactométrie, par exemple, on attribue à chaque mesure du "e-nose" une valeur de concentration d'odeurs qui décrit l'identification des odeurs par les humains. Le processus de transfert des mesures à l'olfactométrie et l'analyse de leur concentration d'odeurs sont longs et coûteux. Ainsi, des méthodes de reconnaissance de formes ont été appliquées aux données du nez électronique pour la prévision automatique de la concentration d'odeurs.

   Il est essentiel d'évaluer la validité des mesures en raison de la sensibilité du "e-nose" aux changements environnementaux et physiques. Les mesures imprécises conduisent à des résultats de reconnaissance de formes peu fiables. Par conséquent, la vérification des échantillons de données provenant du nez électronique et la prise de mesures nécessaires en présence d'anomalies sont essentielles. Nous créons une variante améliorée du "e-nose" existant qui est capable d'évaluer automatiquement et en ligne la validité des échantillons et de prédire l'odeur en utilisant des méthodes appropriées de reconnaissance de formes.

ABSTRACT

This thesis is divided into two main parts. In the first part, we develop a new methodology and the necessary computational tools for shape clustering. In the second part, we tackle the challenging problem of odor prediction in electronic nose (e-nose) technology. Chapter 1 and Chapter 2 describe our proposed methodology for shape clustering. In Chapter 3, we present a new approach for quality odor prediction. Following is a brief overview of the two problems, i.e. shape clustering and odor prediction, and our proposed solutions.

1. **Shape Clustering**: Shapes can be interpreted as closed contours in an infinite dimensional space which can morph into different shapes over time. The main goal in shape modeling is to provide a mathematical model to represent each shape. Statistical shape analysis is a powerful tool in studying the anatomical structures in medical images. In this thesis, motivated by biological applications, we suggest a methodology for surface modeling of cells. Furthermore, we propose a novel technique for clustering cell shapes. The methodology can be applied to other geometrical objects as well.

   Many studies have been conducted to track possible deformations of cells using qualitative descriptions. Our interest is rather providing an accurate numerical assessment of cells. In Chapter 1, statistical models using different basis functions are adapted for modeling the surface of cell shapes both in 2D and 3D spaces. To this end, the surface of a cell is first converted to a set of numerical data. Afterwards, a curve is fitted to these data. At this stage, each cell is represented by a continuous function. The fundamental question, now, is how to distinguish between different cells using their functional forms.

   In Chapter 2, we formulate a clustering Bayesian information criterion (CLUSBIC) for hierarchical clustering of shapes. In this new approach, we treat shapes as continuous curves and we compute the marginal probability associated with each curve. Accordingly, we build the dendrogram for hierarchical clustering employing CLUSBIC. The dendrogram is cut when the marginal probability reaches its maximum.

   We show that CLUSBIC is a natural extension of Ward's linkage, a well-known clustering measure, in Chapter 2. Similar to Bayesian information criterion (BIC) in regression setting, we demonstrate the consistency of CLUSBIC in clustering. CLUSBIC is an extension of BIC, which coincides with BIC if data fall into a single cluster. The usefulness of our proposed methodology in modeling and clustering shapes is examined on simulated and real data.

2. **Odor Prediction**: An e-nose, or artificial olfaction, is a device that analyzes the air to identify odors using an array of gas sensors. The e-nose produces multi-dimensional data for each measurement that it takes from the surrounding environment. A small sub-sample of these measurements are sent to the olfactometry where they are analyzed for odor activities. In olfactometry, for instance, each e-nose measurement is assigned an odor concentration value which describes the odor identifiability by humans. The process of transferring the measurements to the olfactometry and analyzing their odor concentration is time consuming and costly. For this purpose, pattern recognition methods have been applied to e-nose data for automatic prediction of the odor concentration.

   It is essential to assess the validity of the measurements due to the sensitivity of the e-nose to environmental and physical changes. The imprecise measurements lead to unreliable pattern recognition outcomes. Therefore, continuous monitoring of e-nose samples and taking necessary actions in the presence of anomalies is vital. We devise an improved variant of the existing e-nose which is capable of assessing the validity of samples automatically in an online manner, and predicting odor using suitable pattern recognition methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF NOTATIONS

$\mathbb{R}^{\mathrm{K}}$	Set of real numbers
N	Sample size
K	Number of expansion terms
$f^{(k)}(x)$	The partial derivatives of order k of function f with respect to x
$\boldsymbol{\varepsilon}_{N\times 1}$	Vector of size N for error values
$\boldsymbol{\beta}$	Vector of model parameters
$\mathbf{r}_{N\times 1}$	Vector of size N of radii
$\boldsymbol{\theta}_{N\times 1}$	Vector of size N of angles
$\boldsymbol{\Theta}_{N\times K}$	Matrix of basis function values with N rows and K columns
$\mathbf{I}_K$	The identity matrix of size K
(r, θ, ϕ)	The spherical coordinates with radius $r > 0$, polar coordinate $\theta \in [0, \pi]$, and azimuthal $\phi \in [0, 2\pi]$.
$Y_l^m(\theta, \phi)$	The spherical harmonic of degree l and order m at the points θ and ϕ.
$\mathbf{H}_s$	The matrix of spherical harmonic basis functions
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathrm{G}(a, b)$	Gamma distribution with parameters a and b
$\mathrm{IG}(a, b)$	Inverse Gamma distribution with parameters a and b
$t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Student's t-distribution with ν degrees of freedom, location $\boldsymbol{\mu}$ and scale $\boldsymbol{\Sigma}$
$\mathbf{b}$	The extended vector of model parameters
$\mathbf{e}$	The extended vector of error values
$\mathbf{d}$	The vector of groupings
$\mathcal{C}(\mathbf{d})$	The number of unique elements in $\mathbf{d}$
$\mathrm{nrows}(\mathbf{A})$	Number of rows in the matrix $\mathbf{A}$
$\left\{ {n \atop k} \right\}$	Stirling number of the second kind
$\mathcal{o}$	Little-o notation
$\mathcal{O}$	Big-O notation
$\mathbf{A}^\top$	Transpose of the matrix $\mathbf{A}$
$\mathrm{C}(\mathbf{A})$	Column space of the matrix $\mathbf{A}$
$\mathrm{rank}(\mathbf{A})$	Rank of the matrix $\mathbf{A}$
$\lvert\mathbf{A}\rvert$	Determinant of the matrix $\mathbf{A}$
$\mathrm{tr}(\mathbf{A})$	Trace of the the matrix $\mathbf{A}$
$K_{\mathbf{d}}$	Number of parameters induced by the grouping vector $\mathbf{d}$

$\Delta(A, B)$	The Ward's linkage between the groups A and B
$\mathbf{H}$	$\lim_{N\to\infty} \frac{\mathbf{X}^\top \mathbf{X}}{N}$

CHAPTER 1 SHAPE MODELING

Shape modeling plays an important role in medical imaging and computer vision (Krim and Yezzi, 2006; Dryden and Mardia, 1998). Statistical analysis provides an efficient parameterisation of the variability in shapes, and gives a compact representation. A statistical shape model is built on a set of image data using a common coordinate system, like the Cartesian coordinates. The variability between shapes has been studied with different approaches: active shape models (Cootes *et al.*, 1995), computational anatomy (Grenander and Miller, 1995), planar shape analysis (Srivastava *et al.*, 2006; Klassen *et al.*, 2004), etc.

A shape must be converted into a set of numerical data. There are several methodologies for numerical representation of shapes in two-dimensions, such as distance map, binary mask, principal components, Fourier series, see Pincus and Theriot (2007) for a detailed review. One of the main applications of the shape modeling is in biology and medicine, where a cell shape is modeled.

All living organisms are composed of cells which vary widely in shape and structure. The shape of a cell is determined by its external boundary and influenced by intra-cellular activities, and by outside environmental factors. Each cell has a size and a shape suited to its job, but the cell sometimes deforms to an asymmetric shape over time as the cell ages. The cell shape evolution can be tracked using the geometrical parameters that represent the cell shape.

Many studies are devoted to cell shape analysis from different prespectives through image analysis, see for example Lehmussola *et al.* (2005); Zhao and Murphy (2007); Khairy and Howard (2010); Ducroz *et al.* (2011); Lee *et al.* (2012); Johnson *et al.* (2015). These works employ image analysis techniques, which often lack statistical error component. Statistical models serve as more precise and heterogeneous alternatives, but the developed models by other authors are more suitable for geometrically complex shapes.

In statistical shape modeling approach, a curve is fitted to the boundary of a shape, similar to Scott (1987), but perhaps using different basis functions. In other words, each shape is treated as a continuous curve.

Interpolation is a basic and fundamental concept in approximation theory (Steffensen, 1950). The notion of interpolation has been developed in many directions such as polynomials and their properties in applied analysis (Milovanovic *et al.*, 1994; Borwein and Erdélyi, 2012), spline functions (Schoenberg, 1946, 1973), basic theory (Schumaker, 2007), algorithms (De Boor *et al.*, 2013), Hilbert kernels (Atteia, 2014), computer graphics, and geometric modeling (Dierckx, 1995; Bartels *et al.*, 1995).

Suppose the function $h(x)$ is defined over the domain $X \subseteq \mathbb{R}^{\mathrm{K}}$. Assume $h(x)$ is observed

on a finite subset of X, say $\{x_1, x_2, \ldots, x_N\}$. Interpolation attempts to approximate $h(x)$ by another function, say f, such that $f(x_i) = h(x_i)$ for $i = 1, 2, \ldots, N$. Two commonly used and closely related interpolants are polynomials and piecewise cubic splines, see Steffensen (1950). Splines, Fourier, wavelets, circular harmonics are among the most commonly used basis functions for interpolations.

In curve fitting problems, the interpolant may not pass through all data points; it is only required to approach the data points closely. This demands for parameterizing the potential interpolants and having some way of measuring the error. Splines are a form of non-parametric regression model which is more adaptive to local variations (Wahba, 1990; Green and Silverman, 1994; Yuedong, 2011).

The main advantage of this approach is to acquire the probability distribution of the fitted function and to distinguish between different shapes using their probability distribution. In this thesis, we model shapes using a probability distribution and use the likelihood function to measure the similarity of shapes. Then we propose a clustering algorithm (Tryon and Bailey, 1970) using this likelihood function. The novelty in this approach is about introducing a convenient metric for clustering shapes.

The remainder of this chapter is organized as follows. We review shape modeling in 2D space in Section 1.1. Then, we discuss 3D shape modeling and its suitable set of basis functions in Section 1.2.

1.1 Two-Dimensional Bases

A 2D shape measurement consists of points in the 2D Cartesian coordinates (x, y). We assume that curves are obtained by cutting a 3D shape at a certain height using a plane, see Figure 1.1, left panel. After cutting the 3D shape, the curve measurements are denoted by circles in Figure 1.1, middle panel. Our objective is to fit a closed curve such as the solid curve in the middle panel of Figure 1.1. We first transform the data from the Cartesian coordinates (x, y) to the polar coordinates (r, θ). This practical trick facilitates their modeling, as it transforms a smooth closed manifold to a smooth function, see the right panel of Figure 1.1. Then the fitted function in the polar coordinate should be tailored to meet the requirements of a closed curve, see the middle panel of Figure 1.1.

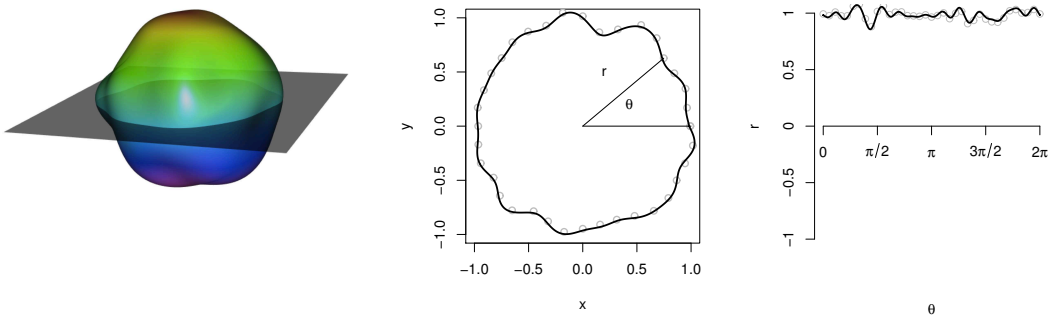

Figure 1.1 Left panel, a 3D object sliced by a plane to create the middle 2D curve. Middle panel, estimation of the closed 2D curve using wavelet transformation; the circles show the measurements of the closed curve achieved by cutting the 3D object in the left. Right panel, estimation of the closed curve in the polar coordinate.

In the polar coordinates system, one can model the data using multiple linear regression. A polynomial function relates the attributes to the response in a non-linear fashion,

$$r_i = f(\theta_i) + \varepsilon_i, \tag{1.1}$$

where

$$f(\theta_i) = \beta_0 + \beta_1\theta_i + \beta_2\theta_i^2 + \ldots + \beta_k\theta_i^k, \;\; i = 1, 2, \ldots N,$$

k is the degree of the polynomial, and ε_i represents the measurement error. It is often helpful to transform the attributes into a representation given by an alternate basis. Let $h_k(\theta) : \mathbb{R} \longrightarrow \mathbb{R}$ be the kth basis of θ, $k = 0, \ldots, K-1$. Then

$$f(\theta) = \sum_{k=0}^{K-1} \beta_k h_k(\theta), \quad \text{with} \quad h_0(\theta) = 1. \tag{1.2}$$

The expansion is nonlinear as a function of the original attribute θ and linear in terms of the β_k's. Having fixed the number of expansion terms K, different choices of $h_k(\theta)$ gives different fits. The parameters β_k's are estimated using least squares.

B-splines, wavelets, Fourier and circular harmonics can be chosen as the basis function $h_k(\theta)$ in (1.2). The modification of equation (1.2) towards closed curve fitting is described in Sections 1.1.1, 1.1.2, 1.1.3, 1.1.4, and 1.1.5.

1.1.1 Cubic Spline

A *cubic spline* with knot ξ is a polynomial of degree 3, which has continuous derivatives up to order 2. The continuity in derivatives up to order 2 gives some constraints on the curve. These constraints can be fulfilled by starting with a basis for cubic polynomial and then add one truncated power functions per knot as a basis in equation (1.2). A truncated power basis is defined as $h(\theta, \xi) = (\theta - \xi)_+^3 = (\theta - \xi)^3 \mathrm{I}_{\{\theta > \xi\}}(x)$, where I is the indicator function

$$\mathrm{I}_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise.} \end{cases} \tag{1.3}$$

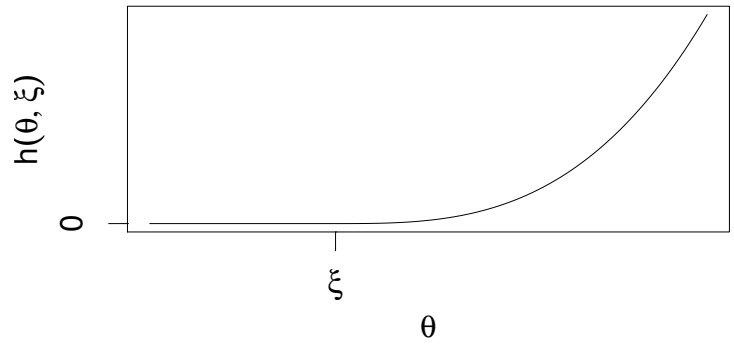

Figure 1.2 The truncated power basis function $h(\theta, \xi) = (\theta - \xi)_+^3$.

The form $f(\theta_i) = \beta_0 + \beta_1\theta_i + \beta_2\theta_i^2 + \beta_3\theta_i^3 + \beta_4(\theta_i - \xi_1)_+^3$ represents a cubic spline with a knot at ξ_1. The cubic spline is a *periodic* function of period $(b - a)$ if the condition $f^{(k)}(a^+) = f^{(k)}(b^-)$ is satisfied, where $f^{(k)}$ is the partial derivative of order k with respect to θ. The method can be extended to $K \geq 1$ knots, $\{\xi_1, \xi_2, \ldots, \xi_K\}$. Multiple knots provide more flexibility and allow for more control on the curvature of the fit. In the case of multiple knots, the cubic spline function has the following form

$$f(\theta_i) = \beta_0 + \beta_1\theta_i + \beta_2\theta_i^2 + \beta_3\theta_i^3 + \sum_{k=1}^{K-4} \beta_{k+3}(\theta_i - \xi_k)_+^3, \quad i = 1, 2, \ldots, N. \tag{1.4}$$

In order to make sure that the closed manifold remains continuous, some continuity constraints are added for the estimated function in the polar coordinates. Let θ_i for $i = 1, \ldots, N$ be the corresponding angle for each observation in polar coordinates and

$$z_1 = \xi_0 < \xi_1 < \ldots < \xi_{K-4} < \xi_{K-3} = z_2$$

where $z_1 = \min(\boldsymbol{\theta})$, $z_2 = \max(\boldsymbol{\theta})$ and $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_N)$. Function continuity, first derivative continuity, and second derivative continuity at the extreme points, z_1 and z_2, in equation (1.4), implies the following conditions on the estimation of β_k's,

$$\beta_1(z_1 - z_2) + \beta_2(z_1^2 - z_2^2) + \beta_3(z_1^3 - z_2^3) - \sum_{k=1}^{K-4} \beta_{k+3}(z_2 - \xi_k)^3 = 0, \tag{1.5}$$

$$2\beta_2(z_1 - z_2) + 3\beta_3(z_1^2 - z_2^2) - 3\sum_{k=1}^{K-4} \beta_{k+3}(z_2 - \xi_k)^2 = 0, \tag{1.6}$$

$$\beta_3(z_1 - z_2) - \sum_{k=1}^{K-4} \beta_{k+3}(z_2 - \xi_k) = 0. \tag{1.7}$$

The equations (1.5), (1.6), and (1.7) can be rewritten as $\mathbf{T}\boldsymbol{\beta} = \mathbf{0}$,

$$\mathbf{T} = \begin{bmatrix} 0 & (z_1 - z_2) & (z_1^2 - z_2^2) & (z_1^3 - z_2^3) & (z_2 - \xi_1)^3 & \cdots & (z_2 - \xi_K)^3 \\ 0 & 0 & 2(z_1 - z_2) & 3(z_1^2 - z_2^2) & -3(z_2 - \xi_1)^2 & \cdots & -3(z_2 - \xi_K)^2 \\ 0 & 0 & 0 & (z_1 - z_2) & (z_2 - \xi_1) & \cdots & (z_2 - \xi_K) \end{bmatrix},$$

$$\boldsymbol{\beta}^\top = \begin{bmatrix} \beta_0 & \beta_1 & \ldots & \beta_K \end{bmatrix},$$

where $\boldsymbol{\beta}^\top$ is the transpose of $\boldsymbol{\beta}$. The least square estimate of $\boldsymbol{\beta}$ subject to the constrains in equations (1.5), (1.6), and (1.7) is

$$\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}} - (\boldsymbol{\Theta}^\top \boldsymbol{\Theta})^{-1}\mathbf{T}^\top [\mathbf{T}(\boldsymbol{\Theta}^\top \boldsymbol{\Theta})^{-1}\mathbf{T}^\top]^{-1}\mathbf{T}\hat{\boldsymbol{\beta}}, \tag{1.8}$$

where $\boldsymbol{\Theta}$ is the matrix of attributes. Here, $\boldsymbol{\Theta}$ contains the expansion terms $h_k(\theta)$. For more details on constrained least squares, see Theil (1963). The effect of the conditions (1.5), (1.6), and (1.7) is demonstrated in Figure 1.3.

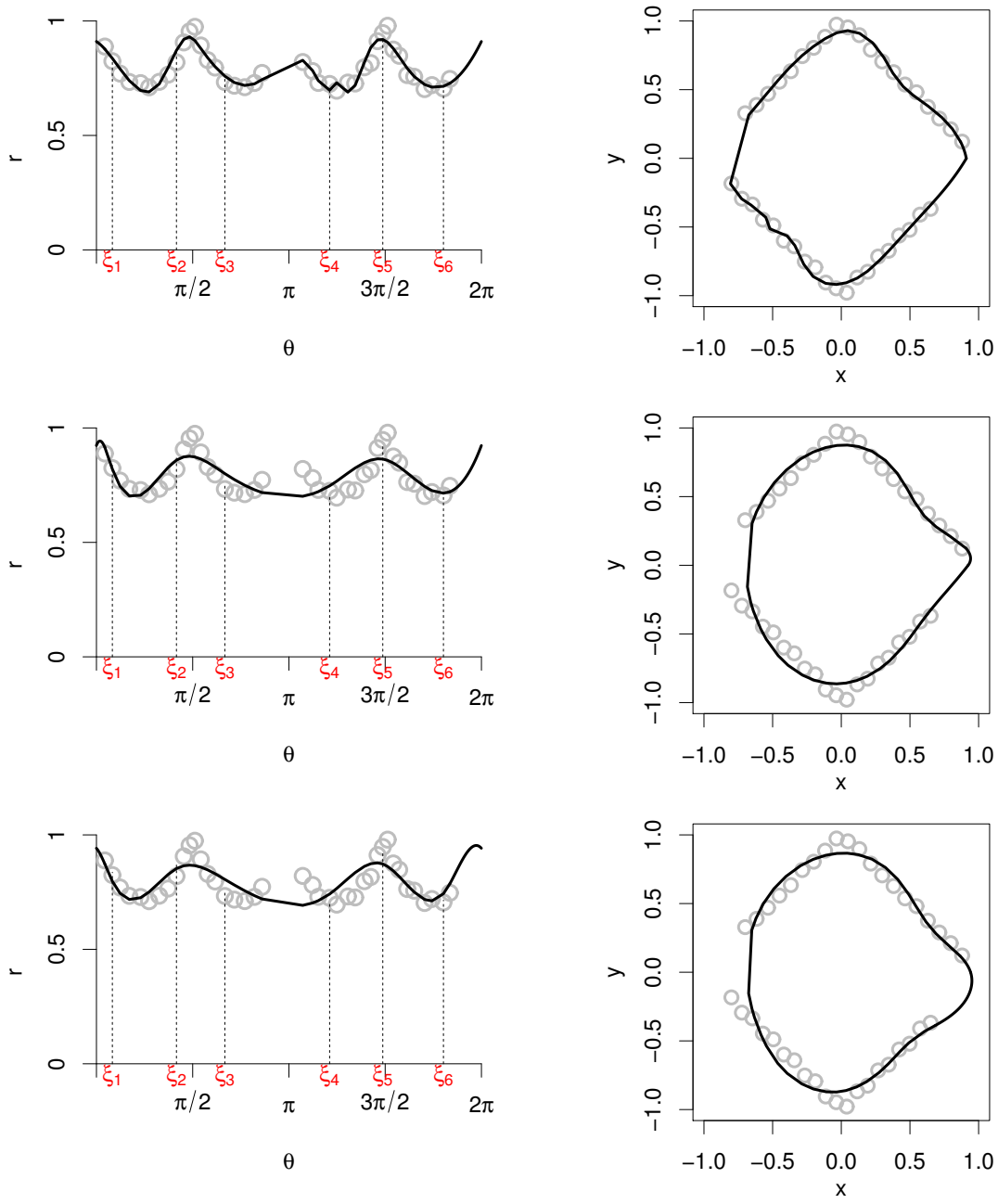

Figure 1.3 The effect of the condition matrix $\mathbf{T}$ on estimating the coefficient $\boldsymbol{\beta}$ of the model. The gray circles represent the data points generated from the function $|x|+|y|=1$ with some Gaussian noise. The black line shows the cubic spline approximation to the data points with six knots $\{\xi_1, \xi_2, \ldots, \xi_6\}$. The top panel only imposes continuity of the curve. The middle panel imposes continuity of the curve and continuity of the first derivative. The bottom panel impose continuity of the curve, continuity of the first derivative and continuity of the second derivative of the curve.

1.1.2 Fourier

Fourier series expansion is one way of representing a periodic signal as an infinite sum of sine wave components. A periodic function $f(\theta)$ with period t expanded using Fourier series is

$$f(\theta) = \frac{a_0}{2} + \sum_{k=1}^{K-1} \{a_k \cos(k\omega_0\theta) + b_k \sin(k\omega_0\theta)\},$$

where $\omega_0 = \frac{2\pi}{t}$. Note that the set of $\{1, \cos(x), \sin(x), \cos(2x), \sin(2x), \ldots\}$ is a complete orthogonal system on $[-\pi, \pi]$ (Tolstov, 2012).

1.1.3 Circular Harmonics

Circular harmonics are another type of basis functions composed of the sine and cosine functions. The bases are similar to Fourier series, but produces orthornormal bases over $[-\pi, \pi]$,

$$f(\theta) = \frac{a_0}{\sqrt{2\pi}} + \sum_{k=0}^{K-1} \left\{ a_k \frac{\cos(k\theta)}{\sqrt{\pi}} + b_k \frac{\sin(k\theta)}{\sqrt{\pi}} \right\}.$$

1.1.4 Wavelets

Wavelets is another basis function useful in equation (1.2). Wavelets were developed mostly over the past two and a half decades. Wavelets refer to a set of orthonormal basis functions generated by dilation and translation of a compactly supported scaling function (or father wavelet), ϕ, and a mother wavelet, ψ, associated with an r-regular multiresolution analysis of $L^2(\mathbb{R})$. A variety of different wavelet families now exist that combine compact support with various degrees of smoothness and numbers of vanishing moments. These wavelets are now the most intensively used wavelet families in practical applications in statistics.

The first wavelet basis was suggested by Haar (1910). He showed that any continuous function $f(x)$ on $[0, 1]$ can be approximated by

$$f_n(x) = \langle \phi_0, f \rangle \phi_0 + \langle \phi_1, f \rangle \phi_1 + \ldots + \langle \phi_n, f \rangle \phi_n,$$

when $n \to \infty$, f_n converges to f uniformly. The wavelet coefficients are defined as $\langle \phi_i, f \rangle = \int \phi_i f(x) dx$, and the Haar basis is,

- $\phi_0 = \mathrm{I}(0 \leq x \leq 1)$,
- $\phi_1 = \mathrm{I}(0 \leq x \leq 1/2) - \mathrm{I}(1/2 \leq x \leq 1)$,

- $\vdots$
- $\phi_n = 2^{j/2}\mathrm{I}(k2^{-j} \leq x \leq (k+1/2)2^{-j}) - \mathrm{I}((k+1/2)2^{-j} \leq x \leq (k+1)2^{-j})$,

where $n = 2^j + k$, $j \geq 0$ and $0 \leq k \leq 2^j - 1$ and $\mathrm{I}_A(x)$ is the indicator function of set A defined in (1.3).

The Haar mother, ψ, and father, ϕ, wavelets are

$$\psi(x) = \begin{cases} 1 & \text{if } x \in [0, \frac{1}{2}), \\ -1 & \text{if } x \in [\frac{1}{2}, 1), \\ 0 & \text{otherwise,} \end{cases} \qquad \phi(x) = \begin{cases} 1 & \text{if } x \in [0,1], \\ 0 & \text{otherwise.} \end{cases}$$

With complete knowledge of a function, $f(x)$, one can initiate a Haar wavelet transform at a fixed resolution. Suppose a function $f(x)$ defined on the interval $[0,1]$. An approximation of the function at fixed level J can be defined as

$$f_J(x) = \sum_{k=0}^{2^J-1} c_{J,k}\phi_{J,k}(x),$$

where the wavelet coefficients $c_{J,k} = \int_0^1 f(x)\phi_{J,k}(x)dx$, and

$$\phi_{J,k}(x) = \begin{cases} 2^{\frac{J}{2}} & \text{if } x \in [2^{-J}k, 2^{-J}(k+1)], \\ 0 & \text{otherwise.} \end{cases}$$

The Haar basis is not an appropriate basis for all applications for several reasons. The building blocks in Haar's decomposition are discontinuous functions that are not effective in approximating smooth functions.

Here, for $\theta \in [0, 2\pi]$, we propose using the Mexican hat wavelet which is the second derivative of the Gaussian density function,

$$f_J(\theta) = \sum_{k=1}^{2^J-1} c_{J,k}\phi_{J,k}(\theta), \quad \phi_{J,k}(\theta) = \pi^{-\frac{1}{4}}2^{\frac{J}{2}}\frac{2}{\sqrt{3}}\{1-(2^J\theta-k)^2\}\exp\left\{-\frac{(2^J\theta-k)^2}{2}\right\}, \quad (1.9)$$

where the wavelet coefficients

$$c_{J,k} = \int_0^1 f(\theta)\phi_{J,k}(\theta)d\theta.$$

The Mexican hat wavelet belongs to the family of continuous wavelet transforms which are non-orthogonal. Applying this basis function to a closed curve requires adjustments using

three constraints in estimating the coefficients of the fit, $\boldsymbol{\beta}$, as in cubic splines (1.5), (1.6), and (1.7). Figure 1.4 exhibits how the method accomplishes in modeling the closed curves. As K increases in the equation (1.2), the model becomes more flexible in capturing the curvature of the shape.

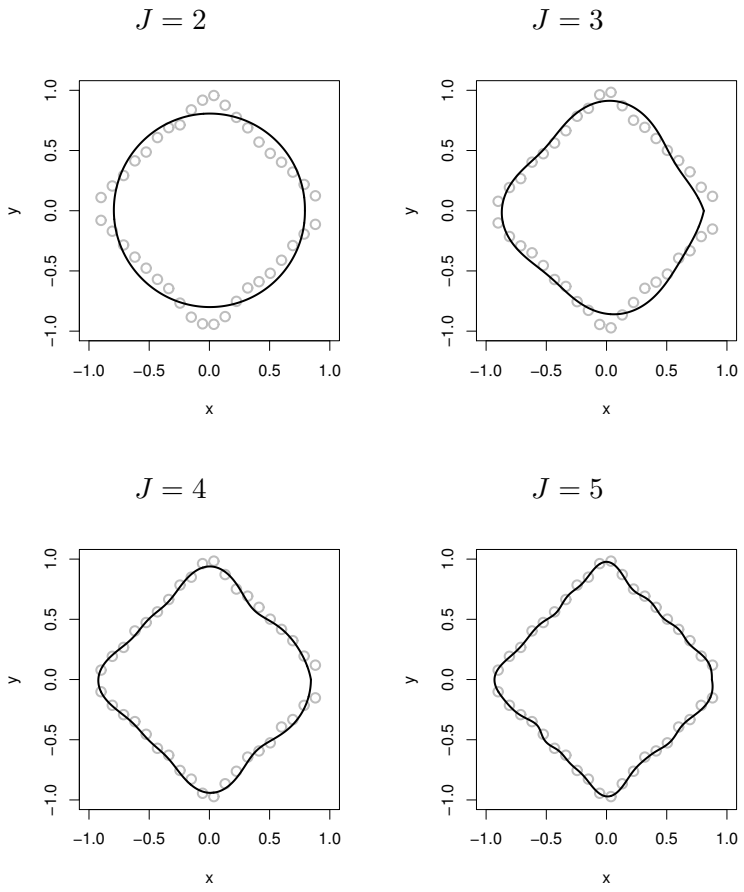

Figure 1.4 The gray circles represent the data points generated from the function $|x|+|y| = 1$ with some Gaussian noise. The black curve shows the Mexican hat wavelet approximation to the data points with different resolutions, J. From top left panel to the bottom right panel J varies from 2 to 5, see also Figure 1.3.

1.1.5 Smoothing Splines

Using more knots in splines modeling provides more flexibility. However, the excessive amount of knots leads to over-fitting and increase the complexity of the fit. There is another spline basis that circumvent the knot selection using a maximal set of knots. In this approach the complexity of the fit is controlled by the regularization parameter λ. A smoothing spline

is the function which minimizes the following equation

$$\text{RSS}(f,\lambda)=\sum_{i=1}^{N}\{r_i-f(\theta_i)\}^2+\lambda\int\{f^{(2)}(t)\}^2dt. \tag{1.10}$$

The equation (1.10) has an explicit unique minimizer which is a natural cubic spline with knots at the unique values of θ_i's. More details on this topic can be found in Hastie *et al.* (2001, Chapter 5). It is computationally more convenient to work with cubic spline basis functions. Here, smoothing splines are defined as $f(\boldsymbol{\theta})=\boldsymbol{\Theta\gamma}$ which is the solution to the following quadratic optimization problem

$$\text{RSS}(\boldsymbol{\gamma},\lambda)=(\boldsymbol{r}-\boldsymbol{\Theta\gamma})^\top(\boldsymbol{r}-\boldsymbol{\Theta\gamma})+\lambda\boldsymbol{\gamma}^\top\Omega_{\boldsymbol{\Theta}}\boldsymbol{\gamma}, \tag{1.11}$$

where $\boldsymbol{\Theta}$ is the matrix of cubic splines and $\Omega_{\boldsymbol{\Theta}}$ the associated penalty matrix. Knots are located equally spaced from each other starting from the minimum to the maximum of the data. The matrix $\Omega_{\boldsymbol{\Theta}}$ contains

$$\omega_{ij}=\int\boldsymbol{\Theta}^{(2)}_{i.}(t)\boldsymbol{\Theta}^{(2)}_{.j}(t)dt,\quad i,j=1,2,\ldots,N,$$

where $\boldsymbol{\Theta}^{(2)}_{i.}(t)$ and $\boldsymbol{\Theta}^{(2)}_{.j}(t)$ refer to the second order derivative of the ith row and jth column of the matrix $\boldsymbol{\Theta}$ with respect to t. The regularization parameter λ controls the roughness of the function $f(\boldsymbol{\theta})=\boldsymbol{\Theta\gamma}$ by imposing its effect on the matrix $\Omega_{\boldsymbol{\Theta}}$.

Fitting splines to closed curves inquires three additional constraints at the extreme points of the curves. Thus, the goal here is to find an estimate for the coefficients of smoothing splines under those additional constraints.

Theorem 1.1. A least squares estimate of $\boldsymbol{\gamma}$ in the model $\boldsymbol{r}=\boldsymbol{\Theta\gamma}+\boldsymbol{\varepsilon}$ subject to a set of equality constraints on $\boldsymbol{\gamma}$, say $\mathbf{T\gamma}=\mathbf{c}_1$ and penalization on curvature, $\boldsymbol{\gamma}^\top\Omega_{\boldsymbol{\Theta}}\boldsymbol{\gamma}\le c_2$, is

$$\tilde{\boldsymbol{\gamma}}=\hat{\boldsymbol{\gamma}}-(\boldsymbol{\Theta}^\top\boldsymbol{\Theta}+\lambda\Omega_{\boldsymbol{\Theta}})^{-1}\mathbf{T}^\top(\mathbf{T}(\boldsymbol{\Theta}^\top\boldsymbol{\Theta}+\lambda\Omega_{\boldsymbol{\Theta}})^{-1}\mathbf{T}^\top)^{-1}[\mathbf{T}\hat{\boldsymbol{\gamma}}-\mathbf{c}_1], \tag{1.12}$$

where $\mathbf{T}$ is $3\times N$ matrix of constraints, $\Omega_{\boldsymbol{\Theta}}$ is $N\times N$ penalty matrix and λ is the fixed smoothing parameter.

Proof. Assume that $\mathbf{T}$ is of size $r\times p$ and $\boldsymbol{\gamma}$ is of size $p\times 1$. Our interest is to minimize

$$S(\boldsymbol{\gamma})=(\mathbf{r}-\boldsymbol{\Theta\gamma})^\top(\mathbf{r}-\boldsymbol{\Theta\gamma}),$$

$$\text{subject to}\quad \mathbf{T\gamma}=\mathbf{c}_1\quad\text{and}\quad\boldsymbol{\gamma}^\top\Omega_{\boldsymbol{\Theta}}\boldsymbol{\gamma}\le c_2,$$

which is another form of constrained least squares (Theil, 1963). By the method of Lagrange multiplier, we may equivalently minimize

$$g(\boldsymbol{\gamma}) = (\mathbf{r} - \boldsymbol{\Theta}\boldsymbol{\gamma})^\top(\mathbf{r} - \boldsymbol{\Theta}\boldsymbol{\gamma}) + \boldsymbol{\delta}^\top(\mathbf{T}\boldsymbol{\gamma} - \mathbf{c}) + \lambda(\boldsymbol{\gamma}^\top\Omega_{\boldsymbol{\Theta}}\boldsymbol{\gamma} - c_2),$$

where $\boldsymbol{\delta}$ is a vector of size $r \times 1$ and λ is a fixed smoothing parameter.

$$\frac{\partial g(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} = -2\boldsymbol{\Theta}^\top\mathbf{r} + 2(\boldsymbol{\Theta}^\top\boldsymbol{\Theta})\boldsymbol{\gamma} + \mathbf{T}^\top\boldsymbol{\delta} + 2\lambda\Omega_{\boldsymbol{\Theta}}\boldsymbol{\gamma} = \mathbf{0}, \tag{1.13}$$

$$\frac{\partial g(\boldsymbol{\beta})}{\partial \lambda} = \mathbf{T}\boldsymbol{\gamma} - \mathbf{c}_1 = \mathbf{0}, \tag{1.14}$$

$$\frac{\partial g(\boldsymbol{\beta})}{\partial \boldsymbol{\delta}} = \boldsymbol{\gamma}^\top\Omega_{\boldsymbol{\Theta}}\boldsymbol{\gamma} - c_2 = 0. \tag{1.15}$$

Solving the equation (1.13) gives

$$\tilde{\boldsymbol{\gamma}} = (\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \lambda\Omega_{\boldsymbol{\Theta}})^{-1}(\boldsymbol{\Theta}^\top\mathbf{r} - \frac{1}{2}\mathbf{T}^\top\boldsymbol{\delta}). \tag{1.16}$$

Substituting the equation (1.16) in equation (1.14),

$$\begin{aligned} \mathbf{T}\tilde{\boldsymbol{\gamma}} &= \mathbf{c}_1, \\ \mathbf{T}(\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \lambda\Omega_{\boldsymbol{\Theta}})^{-1}(\boldsymbol{\Theta}^\top\mathbf{r} - \frac{1}{2}\mathbf{T}^\top\boldsymbol{\delta}) &= \mathbf{c}_1, \end{aligned}$$

$$\frac{1}{2}\boldsymbol{\delta} = (\mathbf{T}(\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \lambda\Omega_{\boldsymbol{\Theta}})^{-1}\mathbf{T}^\top)^{-1}[\mathbf{T}(\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \lambda\Omega_{\boldsymbol{\Theta}})^{-1}\boldsymbol{\Theta}^\top\mathbf{r} - \mathbf{c}_1]. \tag{1.17}$$

By inserting the value of $\boldsymbol{\delta}$ from equation (1.17) in equation (1.16),

$$\begin{aligned} \tilde{\boldsymbol{\gamma}} &= (\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \lambda\Omega_{\boldsymbol{\Theta}})^{-1}\{\boldsymbol{\Theta}^\top\mathbf{r} - \mathbf{T}^\top(\mathbf{T}(\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \lambda\Omega_{\boldsymbol{\Theta}})^{-1}\mathbf{T}^\top)^{-1}[\mathbf{T}(\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \lambda\Omega_{\boldsymbol{\Theta}})^{-1}\boldsymbol{\Theta}^\top\mathbf{r} - \mathbf{c}_1]\}, \\ &= \hat{\boldsymbol{\gamma}} - (\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \lambda\Omega_{\boldsymbol{\Theta}})^{-1}\mathbf{T}^\top\{\mathbf{T}(\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \lambda\Omega_{\boldsymbol{\Theta}})^{-1}\mathbf{T}^\top\}^{-1}(\mathbf{T}\hat{\boldsymbol{\gamma}} - \mathbf{c}_1). \end{aligned}$$

□

1.2 Three-Dimensional Bases

A 3D shape descriptor is greatly beneficial in many fields such as biometrics, biomedical imaging, and computer vision. Double Fourier series and spherical harmonics are widely common for representing the 3D objects. Here, we discuss spherical harmonics for 3D shape modeling.

1.2.1 Spherical Harmonics

Spherical harmonics expansion allows us to regard a closed 3D smooth surface as a function expanded in another space with parameters $\boldsymbol{\beta}$ appearing linearly in the expansion like the 2D case. Spherical harmonics are a natural and convenient choice of basis functions for representing any twice differentiable spherical function.

Let x, y and z denote the Cartesian object space coordinates and θ, ϕ, and r as spherical parameter space coordiantes, where θ is taken as the polar (colatitudinal) coordinate with $\theta \in [0, \pi]$, and ϕ as azimuthal (longitudinal) coordinate with $\phi \in [0, 2\pi]$. Spherical harmonics is somehow equivalent to a 3D extension of the Fourier series, which models r as a function of θ and ϕ. The basis for spherical harmonics $Y_l^m(\theta, \phi)$ of degree l and order m is defined as:

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta)\exp(im\phi), \tag{1.18}$$

where l and m are integers with $|m| \leq l$, and the associated Legendre polynomial P_l^m is defined by differential equation

$$P_l^m(x) = \frac{(-1)^m}{2^l l!}(1-x^2)^{\frac{m}{2}}\frac{d^{l+m}}{dx^{l+m}}(x^2-1)^l.$$

The above equation implies that

$$Y_l^{-m}(\theta, \phi) = (-1)^m Y_l^m(\theta, \phi)^*,$$

where $Y_l^m(\theta, \phi)^*$ is the complex conjugate of $Y_l^m(\theta, \phi)$. Most applications of spherical harmonics require only real valued spherical functions, which are defined as follows,

$$Y_l^m(\theta, \phi) = \begin{cases} \sqrt{2}\sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta)\cos(m\phi) & \text{for } m \geq 0, \\ \sqrt{2}\sqrt{\frac{2l+1}{4\pi}\frac{(l-|m|)!}{(l+|m|)!}} P_l^{|m|}(\cos\theta)\sin(|m|\phi) & \text{for } m < 0. \end{cases} \tag{1.19}$$

Using the basis function in equation (1.19), any spherical function $r(\theta, \phi)$ can be expanded as

$$r(\theta, \phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} c_l^m Y_l^m(\theta, \phi), \tag{1.20}$$

where the coefficients c_l^m are uniquely determined by

$$c_l^m = \int_0^{2\pi} Y_l^m(\theta, \phi) r(\theta, \phi)\sin(\theta) d\phi d\theta. \tag{1.21}$$

Given a function $r(\theta, \phi)$ and a specified maximum degree $L_{\max}$, the coefficients can be extracted by solving the integral in equation (1.21). Another approach is to write $r(\theta, \phi)$ as a linear expansion of Y_l^m's with possibly some measurement errors and find the coefficient of fit through the method of least squares. That is,

$$\boldsymbol{r} = \mathbf{H}_s \boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1.22}$$

where

$$\boldsymbol{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_K \end{pmatrix},$$

$$\mathbf{H}_s = \begin{pmatrix} Y_l^{-l}(\theta_1, \phi_1) & Y_l^{-(l-1)}(\theta_1, \phi_1) & \dots & Y_l^0(\theta_1, \phi_1) & \dots & Y_l^l(\theta_1, \phi_1) \\ Y_l^{-l}(\theta_2, \phi_2) & Y_l^{-(l-1)}(\theta_2, \phi_2) & \dots & Y_l^0(\theta_2, \phi_2) & \dots & Y_l^l(\theta_2, \phi_2) \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ Y_l^{-l}(\theta_N, \phi_N) & Y_l^{-(l-1)}(\theta_N, \phi_N) & \dots & Y_l^0(\theta_N, \phi_N) & \dots & Y_l^l(\theta_N, \phi_N) \end{pmatrix}, \tag{1.23}$$

$|m| \leq l \leq L_{\max}$, and $K = (L_{\max} + 1)^2$. The quality of fit improves as $L_{\max}$ increases, i.e. the more number of expansion terms.

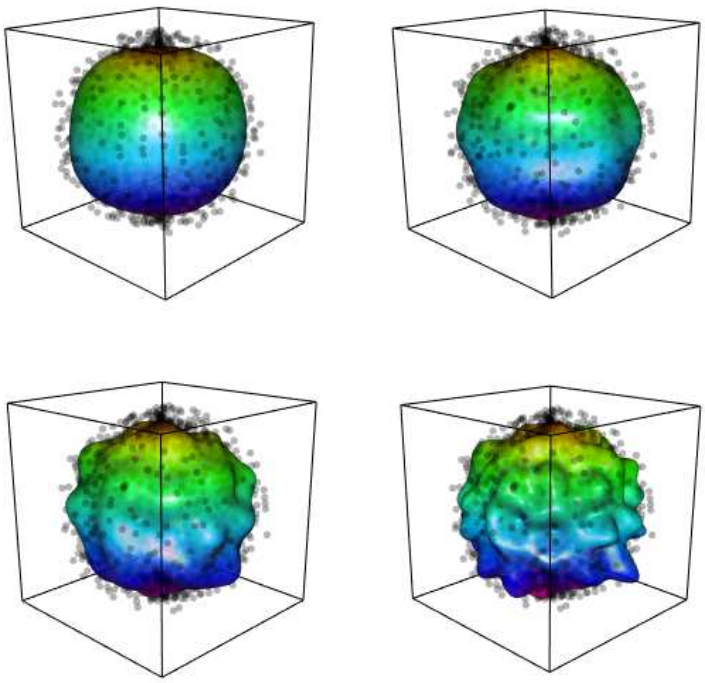

Figure 1.5 Reconstructing a random 3D shape from a sample of scattered points on its surface using spherical harmonics with different degrees. Top left: $L_{\max} = 1$. Top right: $L_{\max} = 2$. Bottom left: $L_{\max} = 3$. Bottom right: $L_{\max} = 4$.

Figure 1.5 shows how the approximation of surface of 3D shapes is feasible using spherical harmonics.

The model (1.22) is only suitable for surface modeling of stellar shapes. Different parametric form for surface modeling, regardless of the type of the shape, is suggested by Brechbühler *et al.* (1995); Duncan and Olson (1993). This parametric form gives us three explicit functions defining the surface of shape as:

$$\begin{pmatrix} x_s(\theta, \phi) \\ y_s(\theta, \phi) \\ z_s(\theta, \phi) \end{pmatrix},$$

where each of the coordinates is modeled as a function of spherical harmonic bases,

$$x_s(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} c_l^{m_x} Y_l^m(\theta,\phi),$$
$$y_s(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} c_l^{m_y} Y_l^m(\theta,\phi),$$
$$z_s(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} c_l^{m_z} Y_l^m(\theta,\phi).$$

Accordingly, the three following linear models are generated,

$$\mathbf{x}_s = \mathbf{H}_s\boldsymbol{\beta}_x + \boldsymbol{\varepsilon}_x,$$
$$\mathbf{y}_s = \mathbf{H}_s\boldsymbol{\beta}_y + \boldsymbol{\varepsilon}_y,$$
$$\mathbf{z}_s = \mathbf{H}_s\boldsymbol{\beta}_z + \boldsymbol{\varepsilon}_z.$$

The set of expansion coefficients $(\boldsymbol{\beta}_x, \boldsymbol{\beta}_y, \boldsymbol{\beta}_z)$ defines the shape completely. Assuming $\mathbf{x}_s$, $\mathbf{y}_s$ and $\mathbf{z}_s$ to be independent of each other, the above three models are equivalent to the following model

$$\begin{bmatrix}\mathbf{x}_s\\ \mathbf{y}_s\\ \mathbf{z}_s\end{bmatrix} = \mathbf{H}_s \begin{bmatrix}\boldsymbol{\beta}_x\\ \boldsymbol{\beta}_y\\ \boldsymbol{\beta}_z\end{bmatrix} + \begin{bmatrix}\boldsymbol{\varepsilon}_x\\ \boldsymbol{\varepsilon}_y\\ \boldsymbol{\varepsilon}_z\end{bmatrix}. \tag{1.24}$$

CHAPTER 2 SHAPE CLUSTERING

In Chapter 1, we discussed shape modeling through linear models using various basis functions. Having characterised shapes by functional forms, we can cluster shapes according to their estimated coefficients. Clustering shapes can be achieved simply by computing the Euclidean distance over their parameters of the models in equation (1.2). However, we show this heuristic method may lead to improper clusters, so we aim to develop a methodology using the fitted models. To this end, we present a likelihood-based approach for clustering shapes in this chapter. From the Bayesian point of view, the clustering procedure is enhanced by contemplating the distribution of parameters in equation (1.2). We asume some prior distributions over parameters of the fit $\boldsymbol{\beta}$, and calculate the marginal distribution of the model by integrating the likelihood with respect to the assumed prior. In this approach, the hierarchy of clusters is built up based on the posterior probabilities (Hartigan, 1990; Booth *et al.*, 2008; Fraley and Raftery, 2002; Yeung *et al.*, 2001). Agglomerative hierarchical clustering is used to establish the dendrogram in which curves are merged as long as the merge improves the posterior probabilities, as discussed in Heard *et al.* (2006).

This chapter is organized as follows. First, a brief sketch of calculating the marginal likelihood function with classical assumptions is provided in Section 2.1. In Section 2.2, we provide a trick to facilitate the computation of marginal likelihood for Gaussian models. In Section 2.3, we introduce a new Bayesian information criterion for clustering curves, we call it CLUSBIC (Mirshahi *et al.*, 2017c). CLUSBIC coincides with the problem of clustering curves using their marginal likelihoods. Considering $\mathbf{d}$ as a grouping vector, the posterior probability of the grouping is proportional to the marginal probabilities of curves. Therefore, clustering curves using posterior probability of the grouping is asymptotically equivalent to CLUSBIC. In Section 2.4, we discuss some robust models and calculations of the posterior probability for such models. In Section 2.5, the consistency of the CLUSBIC is proved. The applicability of the method is verified on a set of simulated and real dataset in 2D and 3D spaces in Section 2.9.

2.1 Gaussian Models

We consider the following linear model in polar coordinates

$$\boldsymbol{r}_{N\times 1} = \boldsymbol{\Theta}_{N\times K}\boldsymbol{\beta}_{K\times 1} + \boldsymbol{\varepsilon}_{N\times 1}, \tag{2.1}$$

where N and K indicate the number of observations and the attributes respectively. We assume $\boldsymbol{\varepsilon}$ is distributed according to the Gaussian distribution with mean $\mathbf{0}_{N\times 1}$ and covariance matrix $\sigma^2\mathbf{I}_N$, i.e. $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_N)$, where $\mathbf{I}_N$ is the identity matrix of size N.

In case of spherical harmonic expansions, equation (1.24), we assume that the error terms $\boldsymbol{\varepsilon}_x$, $\boldsymbol{\varepsilon}_y$, and $\boldsymbol{\varepsilon}_z$ are independent of each other and each follows the Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_N)$. Subsequently,

$$(\boldsymbol{\varepsilon}_x, \boldsymbol{\varepsilon}_y, \boldsymbol{\varepsilon}_z) \sim \mathcal{N}_{3N}(\mathbf{0}, \mathbf{I}_3 \otimes \sigma^2\mathbf{I}_N), \tag{2.2}$$

where the symbol $\otimes$ is the Kronecker product.

Given D distinct shapes, the model associated with this set of shapes is

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}, \tag{2.3}$$

where in case of 2D shapes

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \vdots \\ \boldsymbol{r}_D \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} \boldsymbol{\Theta}_1^1 & \mathbf{0}^2_{N_1\times K} & \dots & \mathbf{0}^D_{N_1\times K} \\ \mathbf{0}^1_{N_2\times K} & \boldsymbol{\Theta}_2^2 & \dots & \mathbf{0}^D_{N_2\times K} \\ \vdots & \vdots & & \vdots \\ \mathbf{0}^1_{N_D\times K} & \mathbf{0}^2_{n_D\times K} & \dots & \boldsymbol{\Theta}_D^D \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_D \end{bmatrix}, \ \mathbf{e} = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \vdots \\ \boldsymbol{\varepsilon}_D \end{bmatrix},$$

and 3D shapes, as it was shown in equation (1.22),

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \vdots \\ \boldsymbol{r}_D \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} \mathbf{H}_{s_1}^1 & \mathbf{0}^2_{N_1\times K} & \dots & \mathbf{0}^D_{N_1\times K} \\ \mathbf{0}^1_{N_2\times K} & \mathbf{H}_{s_2}^2 & \dots & \mathbf{0}^D_{N_2\times K} \\ \vdots & \vdots & & \vdots \\ \mathbf{0}^1_{N_D\times K} & \mathbf{0}^2_{n_D\times K} & \dots & \mathbf{H}_{s_D}^D \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_D \end{bmatrix}, \ \mathbf{e} = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \vdots \\ \boldsymbol{\varepsilon}_D \end{bmatrix},$$

where the matrix $\mathbf{0}^j_{n_i\times p}$ has all its entries zero. For 3D shapes as in equation (1.24), we have,

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}, \tag{2.4}$$

where

$$\mathbf{Y} = \begin{bmatrix} \mathbf{x}_s^1 & \mathbf{y}_s^1 & \mathbf{z}_s^1 \\ \mathbf{x}_s^2 & \mathbf{y}_s^2 & \mathbf{z}_s^2 \\ \vdots & & \\ \mathbf{x}_s^D & \mathbf{y}_s^D & \mathbf{z}_s^D \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} \mathbf{H}_{s_1}^1 & \mathbf{0}^2_{N_1\times K} & \dots & \mathbf{0}^D_{N_1\times K} \\ \mathbf{0}^1_{N_2\times K} & \mathbf{H}_{s_1}^2 & \dots & \mathbf{0}^D_{N_2\times K} \\ \vdots & \vdots & & \vdots \\ \mathbf{0}^1_{N_D\times K} & \mathbf{0}^2_{n_D\times K} & \dots & \mathbf{H}_{s_D}^D \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \boldsymbol{\beta}_x^1 & \boldsymbol{\beta}_y^1 & \boldsymbol{\beta}_z^1 \\ \boldsymbol{\beta}_x^2 & \boldsymbol{\beta}_y^2 & \boldsymbol{\beta}_z^2 \\ \vdots & & \\ \boldsymbol{\beta}_x^D & \boldsymbol{\beta}_y^D & \boldsymbol{\beta}_z^D \end{bmatrix}, \ \mathbf{E} = \begin{bmatrix} \boldsymbol{\varepsilon}_x^1 & \boldsymbol{\varepsilon}_y^1 & \boldsymbol{\varepsilon}_z^1 \\ \boldsymbol{\varepsilon}_x^2 & \boldsymbol{\varepsilon}_y^2 & \boldsymbol{\varepsilon}_z^2 \\ \vdots & & \\ \boldsymbol{\varepsilon}_x^D & \boldsymbol{\varepsilon}_y^D & \boldsymbol{\varepsilon}_z^D \end{bmatrix}.$$

We explain the methodology using the notation for the equation (2.3). Similar methodology applies to equation (2.4), taking into account the property in equation (2.2).

Suppose $\mathbf{d} = (d_1, d_2, \ldots, d_D)$ is a grouping vector, e.g. $\mathbf{d} = (1, 1, \ldots, 1)$ assigns all D shapes to one group and $\mathbf{d} = (1, 2, \ldots, D)$ assigns each shape to a singleton. The likelihood function for the model (2.3) given the grouping vector $\mathbf{d}$ is,

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{b}, \sigma^2, \mathbf{X}, \mathbf{d}) &= \prod_{i=1}^{\mathcal{C}(\mathbf{d})} p(\boldsymbol{r}_{(i)} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\Theta}_{(i)}) \\ &= \prod_{i=1}^{\mathcal{C}(\mathbf{d})} \frac{1}{(2\pi\sigma^2)^{\frac{N_{(i)}}{2}}} \exp\left\{ -\frac{1}{2\sigma^2} (\boldsymbol{r}_{(i)} - \boldsymbol{\Theta}_{(i)}\boldsymbol{\beta})^\top (\boldsymbol{r}_{(i)} - \boldsymbol{\Theta}_{(i)}\boldsymbol{\beta}) \right\}, \end{aligned} \tag{2.5}$$

where $\mathcal{C}(\mathbf{d})$ denotes the number of unique elements in $\mathbf{d}$ (or the number of clusters imposed by $\mathbf{d}$), $N_{(i)}$, $\boldsymbol{r}_{(i)}$, and $\boldsymbol{\Theta}_{(i)}$ represent the number of observations, vector of response, and matrix of covariates after combining the clusters caused by $\mathbf{d}$, and $\mathbf{b}$ is the vector of unknown parameters.

To clarify the notations $N_{(i)}$, $\boldsymbol{r}_{(i)}$, and $\boldsymbol{\Theta}_{(i)}$, suppose $D = 5$ and $\mathbf{d} = (1, 2, 2, 3, 1)$. Consequently, the number of unique elements in $\mathbf{d} = (1, 2, 2, 3, 1)$ is $\mathcal{C}(\mathbf{d}) = 3$. Therefore, the likelihood function given the grouping vector $\mathbf{d} = (1, 2, 2, 3, 1)$ is

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{b}, \sigma^2, \mathbf{X}, \mathbf{d}) &= \prod_{i=1}^{\mathcal{C}(\mathbf{d})} p(\boldsymbol{r}_{(i)} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\Theta}_{(i)}) \\ &= p(\boldsymbol{r}_{(1)} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\Theta}_{(1)}) \times p(\boldsymbol{r}_{(2)} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\Theta}_{(2)}) \times p(\boldsymbol{r}_{(3)} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\Theta}_{(3)}), \end{aligned}$$

where

$$\boldsymbol{r}_{(1)} = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_5 \end{bmatrix}, \boldsymbol{r}_{(2)} = \begin{bmatrix} \boldsymbol{r}_2 \\ \boldsymbol{r}_3 \end{bmatrix}, \boldsymbol{r}_{(3)} = \boldsymbol{r}_4,$$

$$\boldsymbol{\Theta}_{(1)} = \begin{bmatrix} \boldsymbol{\Theta}_1 \\ \boldsymbol{\Theta}_5 \end{bmatrix}, \boldsymbol{\Theta}_{(2)} = \begin{bmatrix} \boldsymbol{\Theta}_2 \\ \boldsymbol{\Theta}_3 \end{bmatrix}, \boldsymbol{\Theta}_{(3)} = \boldsymbol{\Theta}_4,$$

and $N_{(1)} = N_1 + N_5$, $N_{(2)} = N_2 + N_3$, and $N_{(3)} = N_4$.

For the sake of simplicity, we propose conjugate priors (Bernardo and Smith, 1994). Conjugate priors are associated with conjugate posteriors from the same family of distribution

as the prior. To begin with, σ^2 is assumed to be known (see Section 2.9 for further discussion and estimation of σ^2 when it is unknown). The standard conjugate prior imposed on $\boldsymbol{\beta}$, conditional on σ^2, is

$$\boldsymbol{\beta} \mid \sigma^2 \sim \mathcal{N}(\boldsymbol{\beta}_0, \sigma^2 \boldsymbol{V}_0), \tag{2.6}$$

where $\boldsymbol{\beta}_0$, and $\boldsymbol{V}_0$ are the prior mean, and prior covariance matrix for $\boldsymbol{\beta}$ respectively. For a detailed discussion of Bayesian methods see O'Hagan and Forster (2004). The marginal probability of $\boldsymbol{r}$ can be computed as follows,

$$p(\boldsymbol{r} \mid \sigma^2) = \int p(\boldsymbol{r} \mid \boldsymbol{\beta}, \sigma^2) p(\boldsymbol{\beta} \mid \sigma^2) d\boldsymbol{\beta}. \tag{2.7}$$

In case of conjugate priors, the model appears as the multivariate Gaussian distribution,

$$\begin{aligned} p(\boldsymbol{r} \mid \sigma^2) = & \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}} \mid \mathbf{I}_N + \boldsymbol{\Theta} \boldsymbol{V}_0 \boldsymbol{\Theta}^\top \mid^{\frac{1}{2}}} \\ & \times \exp\left\{ -\frac{1}{2\sigma^2} (\boldsymbol{r} - \boldsymbol{\Theta}\boldsymbol{\beta}_0)^\top (\mathbf{I}_N + \boldsymbol{\Theta} \boldsymbol{V}_0 \boldsymbol{\Theta}^\top)^{-1} (\boldsymbol{r} - \boldsymbol{\Theta}\boldsymbol{\beta}_0) \right\}, \end{aligned} \tag{2.8}$$

where $N = \sum_{i=1}^{D} N_i$ denotes the number of observations and $|\boldsymbol{A}|$ is the determinant of $\boldsymbol{A}$. The curves are assigned to a group with maximum $p(\mathbf{d} \mid \mathbf{y})$. By the Bayes theorem,

$$\begin{aligned} p(\mathbf{d}|\mathbf{y}) &= \frac{p(\mathbf{y} \mid \mathbf{d}) p(\mathbf{d})}{p(\mathbf{y})} \\ &= \frac{\prod_{i=1}^{\mathcal{C}(\mathbf{d})} p(\boldsymbol{r}_{(i)}) p(\mathbf{d})}{p(\mathbf{y})} \\ &\propto \prod_{i=1}^{\mathcal{C}(\mathbf{d})} p(\boldsymbol{r}_{(i)}) p(\mathbf{d}). \end{aligned} \tag{2.9}$$

The dendrogram exploits the posterior probability to build the hierarchy. It is expected that the posterior probability reaches its maximum over a reasonable grouping. Thus, the logical cut-off on dendrogram is when the posterior probability is maximized over the dendrogram. As an alternative method, one can use Markov chain Monte Carlo (MCMC) methods in order to find a grouping vector for which the posterior probability is maximized by generating samples from the posterior probability.

2.2 Computational Acceleration

In order to calculate the marginal likelihood $p(\boldsymbol{r} \mid \sigma^2)$ for each curve, one needs to compute the inverse of covariance matrix $(\mathbf{I}_N + \boldsymbol{\Theta} \boldsymbol{V}_0 \boldsymbol{\Theta}^\top)$ which is of size N, the number of observa-

tions. The computation of the inverse has the computational complexity of $\mathcal{O}(N^{2.37})$ (Davie and Stothers, 2013) at the best-case scenario. To circumvent the computational complexity involved in computing the inverse of covariance matrix, we modify the Gaussian model by adding σ^2 to the hierarchy of parameters. Assuming an inverse gamma distribution for σ^2, transform the Gaussian model to Student's t-model which does not require any matrix inversion of size N. Therefore, applying the specified modification,

$$p(\boldsymbol{\beta}, \sigma^2) = p(\boldsymbol{\beta} \mid \sigma^2)p(\sigma^2); \quad \boldsymbol{\beta} \mid \sigma^2 \sim \mathcal{N}(\boldsymbol{\beta}_0, \sigma^2 \boldsymbol{V}_0) \text{ and } \sigma^2 \sim \text{IG}(a, b), \tag{2.10}$$

where a and b are some constants (see further discussion below). The marginal probability of the model, consequently, appears as multivariate Student's t-distribution,

$$p(\boldsymbol{r}) = \frac{\Gamma(a + \frac{N}{2}) b^a \sqrt{|\boldsymbol{V}^*|}}{\Gamma(a)(2\pi)^{\frac{N}{2}} \sqrt{|\boldsymbol{V}_0|}} \left[b + \frac{1}{2} (\boldsymbol{\beta}_0^\top \boldsymbol{V}_0^{-1} \boldsymbol{\beta}_0 + \boldsymbol{r}^\top \boldsymbol{r} - \boldsymbol{\mu}^{*\top} \boldsymbol{V}^{*-1} \boldsymbol{\mu}) \right]^{-(a + \frac{N}{2})}, \tag{2.11}$$

where

$$\boldsymbol{V}^* = (\boldsymbol{V}_0^{-1} + \boldsymbol{\Theta}^\top \boldsymbol{\Theta})^{-1}, \boldsymbol{\mu}^* = \boldsymbol{V}^* (\boldsymbol{V}_0^{-1} \boldsymbol{\beta}_0 + \boldsymbol{\Theta}^\top \boldsymbol{r})$$
$$\text{and } \Gamma(x) = \int_0^\infty t^{x-1} \exp(-t) dt.$$

If σ^2 is nearly degenerate, i.e., $\text{E}(\sigma^2) = \mu_0$, and $\text{Var}(\sigma^2) \approx 0$, this model is, asymptotically, the same as the model (2.8). The computational complexity of this model is bounded by $\mathcal{O}(NK^2)$, which is a significant improvement over the Gaussian model (2.8). The proposed computational trick leads to a marginal probability which comes from a distribution with heavier tails than the Gaussian distribution. This trick is particularly helpful in modeling data containing outliers.

As Smith *et al.* (2008) discussed, agglomerative clustering using Student's t-distribution suffers from some instabilities under the common settings of the hyper-parameters. Here, the hyper-parameters of the inverse gamma distribution are determined such that it produces a nearly degenerate distribution.

The Bayesian approach discussed above cannot be directly applied to closed curve modeling as the estimation of $\boldsymbol{\beta}$ is conditional on a set of constraints of the form $\mathbf{T}\boldsymbol{\beta} = \mathbf{0}$. Inevitably, the prior distributions need to be modified (Davis, 1978; O'Hagan, 1973). Here, instead, we adjust the model so that the coefficients $\boldsymbol{\beta}$ can be estimated through ordinary least squares method with no constraint. Substituting the constraint $(\mathbf{T}\boldsymbol{\beta})^\top (\mathbf{T}\boldsymbol{\beta}) = \mathbf{0}$ for $\mathbf{T}\boldsymbol{\beta} = \mathbf{0}$ solves the issue because $(\mathbf{T}\boldsymbol{\beta})^\top (\mathbf{T}\boldsymbol{\beta}) = \mathbf{0}$ if and only if $\mathbf{T}\boldsymbol{\beta} = \mathbf{0}$. The least squares estimate of $\boldsymbol{\beta}$ in

the linear model $\boldsymbol{r} = \boldsymbol{\Theta}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ subject to the constraint of the form $(\mathbf{T}\boldsymbol{\beta})^\top(\mathbf{T}\boldsymbol{\beta}) = \mathbf{0}$ is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{\Theta}^\top\boldsymbol{\Theta} + \delta\mathbf{T}^\top\mathbf{T})^{-1}\boldsymbol{\Theta}^\top\boldsymbol{r},$$

where δ is estimated such that the constraint $(\mathbf{T}\hat{\boldsymbol{\beta}})^\top(\mathbf{T}\hat{\boldsymbol{\beta}}) \approx \mathbf{0}$ holds. The $\hat{\boldsymbol{\beta}}$ can be viewed as an ordinary least squares solution to the model $\boldsymbol{r}^* = \boldsymbol{\Theta}^*\boldsymbol{\beta} + \boldsymbol{\varepsilon}^*$, where

$$\boldsymbol{r}^* = \begin{bmatrix}\boldsymbol{r}\\ \mathbf{0}\end{bmatrix},\ \boldsymbol{\Theta}^* = \begin{bmatrix}\boldsymbol{\Theta}\\ \sqrt{\delta}\mathbf{T}\end{bmatrix},\ \boldsymbol{\varepsilon}^* = \begin{bmatrix}\boldsymbol{\varepsilon}\\ \mathbf{0}\end{bmatrix},$$

and $\mathbf{0}$ is the vector of 0's with the length equal to the number of the rows in the matrix $\mathbf{T}$, nrows($\mathbf{T}$). The same approach is valid when the estimation of $\boldsymbol{\beta}$ is conditional on extra constraints such as smoothing splines in Theorem 1.1. Similarly, for smoothing spline bases

$$\boldsymbol{r}^* = \begin{bmatrix}\boldsymbol{r}\\ \mathbf{0}\end{bmatrix},\ \boldsymbol{\Theta}^* = \begin{bmatrix}\boldsymbol{\Theta}\\ \sqrt{\delta}\mathbf{T}\\ \sqrt{\lambda}\boldsymbol{\Omega}_{\boldsymbol{\Theta}}^{\frac{1}{2}}\end{bmatrix},\ \boldsymbol{\varepsilon}^* = \begin{bmatrix}\boldsymbol{\varepsilon}\\ \mathbf{0}\end{bmatrix},$$

where λ is the penalization parameter of smoothing spline bases and $\boldsymbol{\Omega}_{\boldsymbol{\Theta}}^{\frac{1}{2}}$ is the square root of the penalty matrix for positive semi-definite matrices (Higham *et al.*, 1990).

2.3 Clustering Bayesian Information Criterion (CLUSBIC)

Here, we introduce a new criterion for clustering, based on the marginal probability, called *Clustering Bayesian Information Criterion* (CLUSBIC). CLUSBIC is similar to BIC in nature, but it is designed for hierarchical clustering purpose. When all data fall into one cluster, CLUSBIC coincides with BIC.

Zellner (1986) proposed a simple informative distribution on the coefficient $\boldsymbol{\beta}$ in Gaussian regression models. He introduced a conjugate Gaussian prior distribution

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, g\sigma^2(\boldsymbol{\Theta}^{*\top}\boldsymbol{\Theta}^*)^{-1}),$$

with mean $\boldsymbol{\beta}_0$ and a special covariance matrix, where $\boldsymbol{\Theta}^*$ is the adjusted covariate matrix, see Section 2.2. This covariance matrix is a scaled version of the covariance matrix of the maximum likelihood estimator of $\boldsymbol{\beta}$. In practice, $\boldsymbol{\beta}_0$ can be taken zero and g is an overdispersion parameter to be estimated or manually tuned. The parameter g can be set according to various common model selection methods such as AIC, BIC and RIC (George and Foster, 2000). Zellner (1983, 1986) suggested the use of prior distribution on g as a fully Bayesian

method (see also Liang *et al.* (2008)). Various other methods have been recommended in the literature as an optimum value for g, like using the empirical Bayes methods (Clyde and George, 2000; Hansen and Yu, 2001). Choosing the number of clusters according to the marginal distribution is analogous to using the BIC criterion in model selection problem if g equals the number of the observations N, $g = N$.

Denote by $\mathcal{M}$ the set of all possible models. The set $\mathcal{M}$ contains all the possible ways that one can assign D distinguishable shapes into $D, D-1, D-2, \ldots, 1$ indistinguishable clusters. In other words, $\mathcal{M}$ has

$$\left\{ {D \atop D} \right\} + \left\{ {D \atop D-1} \right\} + \ldots + \left\{ {D \atop 1} \right\}$$

possible models, where $\left\{ {n \atop k} \right\}$ indicates the Stirling number of the second kind which counts the number of ways to partition a set of n labeled objects into k non-empty unlabeled subsets.

In model selection problems, one selects a subset of covariates, from the data matrix $\mathbf{X}$, which gives lower prediction error for future observations by an information criterion. Particularly, in model selection, one examines the following hypothesis

$$H_0 : \boldsymbol{\beta}_{j_1} = \boldsymbol{\beta}_{j_2} = \ldots = \boldsymbol{\beta}_{j_k} = \mathbf{0} \text{ vs. } H_1 : \boldsymbol{\beta}_{j_1} \neq \mathbf{0}, \ldots, \boldsymbol{\beta}_{j_k} \neq \mathbf{0}, \tag{2.12}$$

for $1 \leq j_1 < j_2 < \ldots < j_k \leq D$. However, in clustering, one is interested in finding the different ways of combining the covariates. In other words, the hypothesis which is being tested in this case is

$$H_0 : \mathbf{Tb} = \mathbf{0}, \text{ vs. } H_1 : \mathbf{Tb} \neq \mathbf{0}, \tag{2.13}$$

where $\mathbf{T}_{qK \times DK}$ is the matrix of constraints such that $\mathbf{Tb} = \mathbf{0}$ is a set of q linear constraints on the $\boldsymbol{\beta}_j$'s for $1 \leq j \leq D$.

Example 2.1. Suppose $D = 2$. The linear model in equation (2.3) for the 2 shapes includes

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \boldsymbol{\Theta}_1^1 & \mathbf{0}_{N_1 \times K}^2 \\ \mathbf{0}_{N_2 \times K}^1 & \boldsymbol{\Theta}_2^2 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{bmatrix}, \text{ and } \mathbf{e} = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \end{bmatrix}.$$

In order to verify whether the 2 shapes belong to the same cluster, one needs to test the following hypothesis

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

Or equivalently, one can test

$$H_0 : \mathbf{Tb} = \mathbf{0} \text{ vs. } H_1 : \mathbf{Tb} \neq \mathbf{0},$$

where $\mathbf{T} = \begin{bmatrix} \mathbf{I}_K & -\mathbf{I}_K \end{bmatrix}$, with $\mathbf{I}_K$ being an identity matrix and $-\mathbf{I}_K$ being a diagonal matrix with -1 entries across the diagonal.

In the following theorem, we provide the derivation of CLUSBIC.

Theorem 2.1. The problem of clustering a set of D shapes coincides with the CLUSBIC if $g = N_i$, the number of observations, in the marginal likelihood for each shape $i = 1, 2, \ldots, D$.

Proof. The marginal likelihood for a set of D shapes is

$$p(\mathbf{y} \mid \mathbf{X}, \mathbf{d}, \sigma^2) = \int_{\mathbf{b}} p(\mathbf{y} \mid \mathbf{b}, \mathbf{X}, \mathbf{d}, \sigma^2) p(\mathbf{b} \mid \mathbf{d}, \sigma^2) d\mathbf{b}.$$

For simplicity, we use the notation $p(\mathbf{y} \mid \mathbf{b}, \sigma^2)$ instead of $p(\mathbf{y} \mid \mathbf{b}, \mathbf{X}, \mathbf{d}, \sigma^2)$. In order to obtain an approximation to this integral, we take a second order Taylor expansion of the log-likelihood at $\tilde{\mathbf{b}}$, which is the solution to the following constrained optimization problem,

$$\max \log\{p(\mathbf{y} \mid \mathbf{b}, \sigma^2)\}, \text{ subject to } \mathbf{T}\mathbf{b} = \mathbf{0}. \tag{2.14}$$

The $\tilde{\mathbf{b}}$ can be found using the method of Lagrange multiplier. The Lagrangian function for this problem is

$$\mathcal{L}(\mathbf{b}, \boldsymbol{\lambda}) = \log\{p(\mathbf{y} \mid \mathbf{b}, \sigma^2)\} + \boldsymbol{\lambda}^\top \mathbf{T}\mathbf{b}, \tag{2.15}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. Expanding $\mathcal{L}(\mathbf{b}, \boldsymbol{\lambda})$ about $\tilde{\mathbf{b}}$ and $\tilde{\boldsymbol{\lambda}}$.

$$\begin{aligned} \mathcal{L}(\mathbf{b}, \boldsymbol{\lambda}) = \mathcal{L}(\tilde{\mathbf{b}}, \tilde{\boldsymbol{\lambda}}) &+ \begin{bmatrix} (\mathbf{b} - \tilde{\mathbf{b}})^\top & (\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}})^\top \end{bmatrix} \begin{bmatrix} \frac{\partial \log\{p(\mathbf{y}|\mathbf{b}, \sigma^2)\}}{\partial \mathbf{b}}\Big|_{\mathbf{b}=\tilde{\mathbf{b}}} + \mathbf{T}^\top \boldsymbol{\lambda} \\ \mathbf{T}\tilde{\mathbf{b}} \end{bmatrix} \\ &+ \frac{1}{2} \begin{bmatrix} (\mathbf{b} - \tilde{\mathbf{b}})^\top & (\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}})^\top \end{bmatrix} \begin{bmatrix} \frac{\partial^2 \log\{p(\mathbf{y}|\mathbf{b}, \sigma^2)\}}{\partial \mathbf{b} \partial \mathbf{b}^\top} & \mathbf{T}^\top \\ \mathbf{T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{b} - \tilde{\mathbf{b}} \\ \boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}} \end{bmatrix} \qquad (2.16) \\ &+ \mathcal{O}_p(||\mathbf{b} - \tilde{\mathbf{b}}||^3). \qquad (2.17) \end{aligned}$$

$$\begin{aligned} \mathcal{L}(\mathbf{b}, \boldsymbol{\lambda}) = \log\{p(\mathbf{y} \mid \tilde{\mathbf{b}}, \sigma^2)\} &+ (\mathbf{b} - \tilde{\mathbf{b}})^\top \frac{\partial^2 \log\{p(\mathbf{y} \mid \mathbf{b}, \sigma^2)\}}{\partial \mathbf{b} \partial \mathbf{b}^\top}\Big|_{\mathbf{b}=\tilde{\mathbf{b}}} (\mathbf{b} - \tilde{\mathbf{b}}) \\ &+ 2(\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}})^\top \mathbf{T}(\mathbf{b} - \tilde{\mathbf{b}}) + \mathcal{O}_p(||\mathbf{b} - \tilde{\mathbf{b}}||^3). \end{aligned} \tag{2.18}$$

Under the assumption that $\mathbf{Tb} = \mathbf{0}$,

$$\log\{p(\mathbf{y} \mid \mathbf{b}, \sigma^2)\} = \log\{p(\mathbf{y} \mid \tilde{\mathbf{b}}, \sigma^2)\} + (\mathbf{b} - \tilde{\mathbf{b}})^\top \frac{\partial^2 \log\{p(\mathbf{y} \mid \mathbf{b}, \sigma^2)\}}{\partial \mathbf{b} \partial \mathbf{b}^\top}\bigg|_{\mathbf{b}=\tilde{\mathbf{b}}} (\mathbf{b} - \tilde{\mathbf{b}}) + \mathcal{O}_p(||\mathbf{b} - \tilde{\mathbf{b}}||^3). \tag{2.19}$$

In case of $\mathbf{y}$ having a Gaussian distribution, the above equation is exact as $\mathcal{O}_p(||\mathbf{b}-\tilde{\mathbf{b}}||^3) = 0$. Defining the average observed Fisher information matrix as

$$\bar{\mathbf{J}}(\tilde{\mathbf{b}}, \mathbf{y}) = -\frac{1}{\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i} \frac{\partial^2 \log\{p(\mathbf{y} \mid \mathbf{b}, \sigma^2)\}}{\partial \mathbf{b} \partial \mathbf{b}^\top}\bigg|_{\mathbf{b}=\tilde{\mathbf{b}}},$$

we have

$$\int_{\mathbf{b}} p(\mathbf{y} \mid \mathbf{b}, \sigma^2) p(\mathbf{b} \mid \sigma^2) d\mathbf{b} = p(\mathbf{y} \mid \tilde{\mathbf{b}}, \sigma^2) \int_{\mathbf{b}} \exp\{-\frac{1}{2}(\mathbf{b} - \tilde{\mathbf{b}})^\top \sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i \bar{\mathbf{J}}(\tilde{\mathbf{b}}, \mathbf{y})(\mathbf{b} - \tilde{\mathbf{b}})\} \times p(\mathbf{b} \mid \sigma^2) d\mathbf{b} + \mathcal{O}_p(||\mathbf{b} - \tilde{\mathbf{b}}||^3).$$

Considering $\mathbf{b} \sim \mathcal{N}(\tilde{\mathbf{b}}, \bar{\mathbf{J}}^{-1}(\tilde{\mathbf{b}}, \mathbf{y}))$, where $\bar{\mathbf{J}}^{-1}(\tilde{\mathbf{b}}, \mathbf{y}) = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1} \sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i$,

$$\begin{aligned}
\int_{\mathbf{b}} p(\mathbf{y} \mid \mathbf{b}, \sigma^2) p(\mathbf{b} \mid \sigma^2) d\mathbf{b} &= p(\mathbf{y} \mid \tilde{\mathbf{b}}, \sigma^2) \int_{\mathbf{b}} (2\pi)^{-\frac{K\mathcal{C}(\mathbf{d})}{2}} |\bar{\mathbf{J}}(\tilde{\mathbf{b}}, \mathbf{y})|^{\frac{1}{2}} \\
&\quad \times \exp\{-\frac{1}{2}(\mathbf{b} - \hat{\mathbf{b}})^\top (\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i + 1) \bar{\mathbf{J}}(\tilde{\mathbf{b}}, \mathbf{y})(\mathbf{b} - \tilde{\mathbf{b}})\} d\mathbf{b} \\
&\quad + \mathcal{O}_p(||\mathbf{b} - \tilde{\mathbf{b}}||^3) \\
&= p(\mathbf{y} \mid \tilde{\mathbf{b}}, \sigma^2) |\bar{\mathbf{J}}(\tilde{\mathbf{b}}, \mathbf{y})|^{\frac{1}{2}} |(\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i + 1) \bar{\mathbf{J}}(\tilde{\mathbf{b}}, \mathbf{y})|^{-\frac{1}{2}} \\
&\quad + \mathcal{O}_p(||\mathbf{b} - \tilde{\mathbf{b}}||^3) \\
&= p(\mathbf{y} \mid \tilde{\mathbf{b}}, \sigma^2) (\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i + 1)^{-\frac{K\mathcal{C}(\mathbf{d})}{2}} |\bar{\mathbf{J}}(\tilde{\mathbf{b}}, \mathbf{y})|^{\frac{1}{2}} |\bar{\mathbf{J}}(\tilde{\mathbf{b}}, \mathbf{y})|^{-\frac{1}{2}} \\
&\quad + \mathcal{O}_p(||\mathbf{b} - \tilde{\mathbf{b}}||^3).
\end{aligned}$$

Consequently, as $\tilde{\mathbf{b}} \xrightarrow{p} \mathbf{b}$ (Newey and McFadden, 1994), we have

$$-2\log\{p(\mathbf{y} \mid \mathbf{b}, \sigma^2)\} = -2\log\{p(\mathbf{y} \mid \tilde{\mathbf{b}}, \sigma^2)\} + K\mathcal{C}(\mathbf{d}) \log(\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i + 1). \tag{2.20}$$

Given that $\log(\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i) \approx \log(\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i + 1)$ for large values of $\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i$, the CLUSBIC expression is derived. □

The proof justifies our choice of linkage in hierarchical clustering as it closely reflects the notion of Ward's linkage (Ward, 1963).

Ward's linkage is a popular method in hierarchical clustering which is based on minimizing variance after the merge. Ward's method merges two clusters A and B that minimize the sum of squares after the merge. Define

$$\Delta(A, B) = \sum_{k \in (AB)} ||x_k - m_{(AB)}||^2 - \sum_{k \in A} ||x_k - m_A||^2 - \sum_{k \in B} ||x_k - m_B||^2,$$

where m_j is the center of cluster j. Here, $\Delta(A, B)$ is the merging cost of combining the clusters A and B. Ward's method keeps this growth as small as possible, i.e. merges the closest clusters in terms of variance.

The CLUSBIC is an approximation to the logarithm of marginal likelihood. In case of Gaussian models, CLUSBIC is explicitly equal to the logarithm of marginal likelihood. Using CLUSBIC in clustering suggests merging the groups i and j together only if that leads to decrease in the CLUSBIC. The CLUSBIC is calculated for different combinations of shapes. A pair of grouping that minimizes CLUSBIC is a merging candidate. As an example, the groups i and j are merged together in the second level of the hierarchy only if that decreases the total CLUSBIC comparing with one level before.

$$\begin{aligned}
\text{CLUSBIC}^{(2)} - \text{CLUSBIC}^{(1)} = & -2\{\ell_1 + \ell_2 + \ldots + \ell_{(ij)} + \ldots + \ell_{\mathcal{C}(\mathbf{d})}\} \\
& + K(\mathcal{C}(\mathbf{d}) - 1) \log(\sum_{q=1}^{\mathcal{C}(\mathbf{d})-1} N_q + 1) \\
& + 2\{\ell_1 + \ell_2 + \ldots + \ell_i + \ldots + \ell_j + \ldots + \ell_D\} \\
& - K\mathcal{C}(\mathbf{d}) \log(\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i + 1),
\end{aligned}$$

where $\ell_i = \log\{p(\boldsymbol{r} \mid \boldsymbol{\beta}, \boldsymbol{\Theta}, \sigma^2)\}$ and $\ell_{(ij)}$ is the log-likelihood after merging the group i with group j. As the total number of observations is fixed at each level of hierarchy, i.e.

$\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i = \sum_{q=1}^{\mathcal{C}(\mathbf{d})-1} N_q$,

$$\text{CLUSBIC}^{(2)} - \text{CLUSBIC}^{(1)} = -2\{\ell_{(ij)} - (\ell_i + \ell_j)\} - K \log(\sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i + 1).$$

The relation between the CLUSBIC and Ward's linkage is reported in form of a corollary as follows.

Corollary 2.1. For Gaussian models, the Ward's linkage is closely related to CLUSBIC. Minimizing the CLUSBIC between each two consecutive levels of dendrogram leads to minimizing the following metric,

$$c_{ij} = \ell_i + \ell_j - \ell_{(ij)}; \;\; i, j = 1, 2, \ldots D.$$

Now, considering the model $\boldsymbol{r} = \boldsymbol{\Theta}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$,

$$\begin{aligned} \log\Big\{p(\boldsymbol{r}_i \mid \hat{\boldsymbol{\beta}}_i, \boldsymbol{\Theta}, \sigma^2)\Big\} &= -\frac{N_i}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{r}_i - \boldsymbol{\Theta}_i\hat{\boldsymbol{\beta}})^\top(\boldsymbol{r}_i - \boldsymbol{\Theta}_i\hat{\boldsymbol{\beta}}_i) \\ &= -\frac{N_i}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}||\boldsymbol{r}_i - \hat{\mathrm{E}}(r_i)||_2^2. \end{aligned}$$

Substituting the above equation in the c_{ij} metric, we have

$$\begin{aligned} c_{ij} &= \frac{1}{2\sigma^2}\{||r_{(ij)} - \hat{\mathrm{E}}(r_{(ij)})||_2^2 - ||r_i - \hat{\mathrm{E}}(r_i)||_2^2 - ||r_j - \hat{\mathrm{E}}(r_j)||_2^2\}, \\ &\approx \Delta(i, j) \end{aligned}$$

When $\boldsymbol{\Theta}^\top\boldsymbol{\Theta}$ is diagonal, c_{ij} equals $\Delta(i, j)$.

2.4 Heavy-Tailed Models

In the presence of outliers in the data, or small model departures, the Gaussian assumption on $\boldsymbol{\varepsilon}$ is debatable and lacks robustness. Consequently, one requires to consider a heavy-tailed distribution in order to guard against the sensitivity of Gaussian distribution to outliers. Here, we investigate how to reduce the sensitivity of marginal probability with respect to uncertain inputs. In order to perform clustering according to CLUSBIC, different distribution assumptions on the model, while fixing the priors, are discussed.

The general family of scale mixtures of Gaussian (Andrews and Mallows, 1974) is a sub-class of symmetric distributions that includes a large class of heavy-tailed distributions. Suppose $\mathbf{x}$ is a k-dimensional zero-mean Gaussian attribute with covariance matrix $\mathbf{Q}_{k\times k}$

and z is a positive random scale. The random vector $\mathbf{y}$ is a scale mixture of Gaussian if $\mathbf{y} \stackrel{d}{=} z\mathbf{x}$, where $\mathbf{x}$ and z are independent, and $\stackrel{d}{=}$ denotes equality in distribution. Depending on the distribution of z, $\mathbf{y}$ takes on different heavy-tailed distributions such as Student's t-distribution, Laplace, contaminated Gaussian etc. This family has been extensively used in robust Bayesian modeling (Box and Tiao, 2011; Gonçalves *et al.*, 2015).

By taking scale mixture of Gaussian approach on the linear model , $\boldsymbol{r}_{N\times 1} = \boldsymbol{\Theta}_{N\times K}\boldsymbol{\beta}_{K\times 1} + \boldsymbol{\varepsilon}_{N\times 1}$, the marginal distribution of model, $p(\boldsymbol{r})$, is not necessarily analytically tractable. For this purpose, the marginal probability is approximated through Markov Chain Monte Carlo (MCMC) methods (Chib and Jeliazkov, 2001). For instance,

$$\boldsymbol{r} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\Theta}, \mathbf{u}, \mathbf{d} \sim \mathcal{N}(\boldsymbol{\Theta}\boldsymbol{\beta}, \sigma^2 \text{diag}\{\frac{1}{u_1}, \frac{1}{u_2}, \ldots, \frac{1}{u_N}\}),$$

$$\text{and } u_i \sim \text{G}(\frac{\nu}{2}, \frac{\nu}{2}) \text{ for } i = 1, 2, \ldots, n,$$

leads to marginal distribution $\boldsymbol{r} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\Theta}, \mathbf{d} \sim t_\nu(\boldsymbol{\Theta}\boldsymbol{\beta}, \sigma^2\mathbf{I})$ where $t_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the multivariate Student's t-distribution with ν degrees of freedom, location $\boldsymbol{\mu}$ and scale matrix $\boldsymbol{\Sigma}$. Adding some prior for $\boldsymbol{\beta}$ and σ^2 as in Section 2.1, provides full conditional distribution and one can compute marginal probability using Gibbs sampling, see Chib (1995). The new set of distributional assumptions brings robustness with the price of increasing complexity of computations. The heavy-tailed models are not used for analysis in this thesis and they require further studies as future research.

2.5 Consistency of CLUSBIC

In this section, we show that the CLUSBIC decision rule is consistent in choosing the true clustering as $N \to \infty$. Assuming that there exists a model $m_0 \in \mathcal{M}$ that represents the true clustering. The CLUSBIC, developed in Theorem 2.1, is said to be a consistent measure if

$$\lim_{N\to\infty} p_N(m_0) = \lim_{N\to\infty} p_N(\hat{m} = m_0) = 1, \tag{2.21}$$

where $\hat{m}$ is the model selected by CLUSBIC.

In regression setting, the space of models $\mathcal{M}$ can be partitioned into two sub-spaces of under-specified and over-specified models, in a rather straightforward fashion. The space of under-specified models $\mathcal{M}_1$ contains all models that mistakenly exclude the attributes of the true models. On the other hand, the space of over-specified models $\mathcal{M}_2$ contains all models that include more attributes besides the true model's attributes. In other words, the sub-spaces $\mathcal{M}_1$ and $\mathcal{M}_2$ can be effortlessly established considering the presence or the absence of

the attributes of the true model. Therefore, more formally, we have

$$\mathcal{M}_1 = \{m \in \mathcal{M} \mid m \not\supseteq m_0\}, \text{ and } \mathcal{M}_2 = \{m \in \mathcal{M} \mid m \supseteq m_0\}.$$

Consequently, for each model that belongs to $\mathcal{M}_2$, the dimension of the model, K, is always greater than the true model.

The definition of under-specified and over-specified models needs to be adjusted for the clustering context. Consider the general linear model $\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$ in equation (2.3). Let $N = \sum_{i=1}^{\mathcal{C}(\mathbf{d})} N_i$, and $K_{\mathbf{d}} = K\mathcal{C}(\mathbf{d})$ be the total number of observations, and the number of parameters in the model respectively. We define $\mathcal{M}_1$ and $\mathcal{M}_2$ as follows for the clustering problem,

$$\mathcal{M}_1 = \{m \in \mathcal{M} \mid \mathbf{T}_m\mathbf{b} \neq \mathbf{0}\}, \text{ and } \mathcal{M}_2 = \{m \in \mathcal{M} \mid \mathbf{T}_m\mathbf{b} = \mathbf{0}\},$$

where $\mathbf{T}_m$ is the matrix of constraints for the model $m \in \mathcal{M}$.

Example 2.2. Suppose $D = 4$, the set of all possible models contains

$$\left\{ {4 \atop 1} \right\} + \left\{ {4 \atop 2} \right\} + \left\{ {4 \atop 3} \right\} + \left\{ {4 \atop 4} \right\} = 1 + 7 + 6 + 1 = 15$$

models in total. Let $\mathbf{d}$ denote the grouping vector and $\mathcal{C}(\mathbf{d})$ be the number of clusters in $\mathbf{d}$. The space of $\mathcal{M}$ contains the following models,

- $\mathcal{C}(\mathbf{d}) = 1 \Longrightarrow \mathbf{d}_1 = (1, 1, 1, 1)$.
- $\mathcal{C}(\mathbf{d}) = 2 \Longrightarrow \mathbf{d}_2 = (1, 1, 1, 2), \mathbf{d}_3 = (1, 2, 2, 2), \mathbf{d}_4 = (1, 2, 1, 1), \mathbf{d}_5 = (1, 1, 2, 1), \mathbf{d}_6 = (1, 1, 2, 2), \mathbf{d}_7 = (1, 2, 1, 2), \mathbf{d}_8 = (1, 2, 2, 1)$.
- $\mathcal{C}(\mathbf{d}) = 3 \Longrightarrow \mathbf{d}_9 = (1, 1, 2, 3), \mathbf{d}_{10} = (1, 2, 1, 3), \mathbf{d}_{11} = (1, 2, 3, 1), \mathbf{d}_{12} = (1, 2, 2, 3), \mathbf{d}_{13} = (1, 2, 3, 2), \mathbf{d}_{14} = (1, 2, 3, 3)$.
- $\mathcal{C}(\mathbf{d}) = 4 \Longrightarrow \mathbf{d}_{15} = (1, 2, 3, 4)$.

Assume the grouping associated to the true model is $\mathbf{d}_7 = (1, 2, 1, 2)$. Therefore,

$$\mathbf{T}_{m_0} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I} \end{bmatrix},$$

where $\mathbf{I}$, $\mathbf{0}$, and $-\mathbf{I}$ are an identity matrix, a matrix with zero entries and a diagonal matrix with -1 entries across the diagonal respectively, which all are of size $K \times K$. According to the definition, the sets of $\mathcal{M}_1$, and $\mathcal{M}_2$ for this example are

$$\mathcal{M}_1 = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_6, \mathbf{d}_8, \mathbf{d}_9, \mathbf{d}_{11}, \mathbf{d}_{12}, \mathbf{d}_{14}\},$$

$$\mathcal{M}_2 = \{\mathbf{d}_7, \mathbf{d}_{10}, \mathbf{d}_{13}, \mathbf{d}_{15}\}.$$

In Figure 2.1, you can see the comparison between the true model and over-specified models through dendrogram.

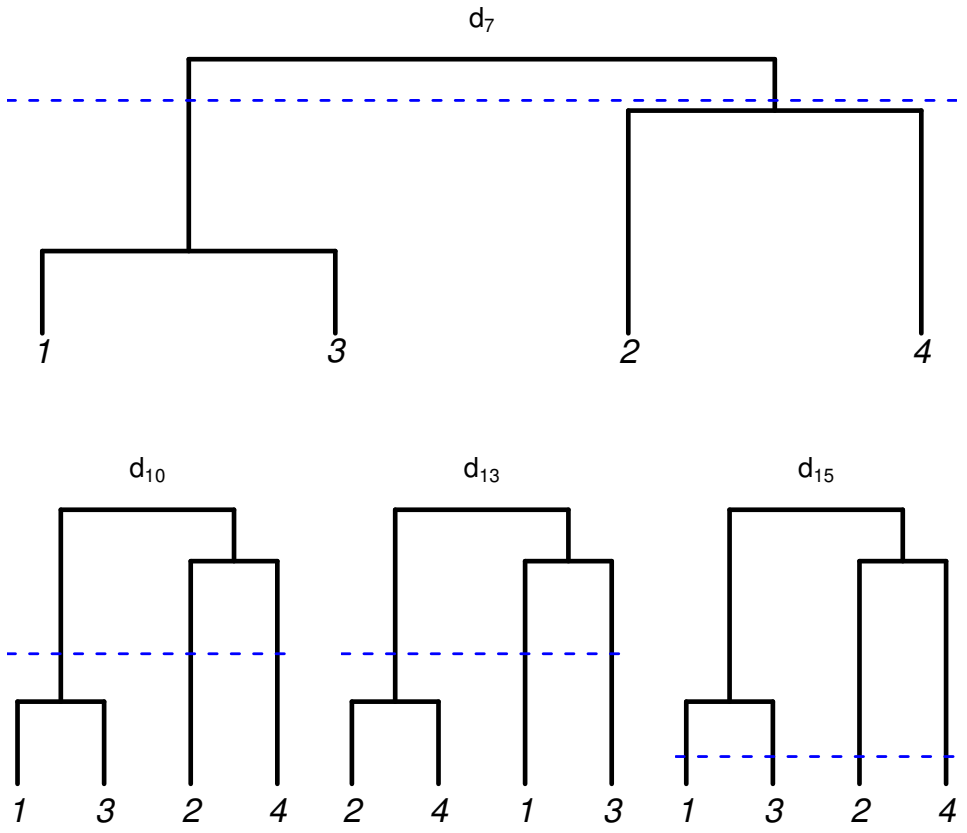

Figure 2.1 Top panel: the dendrogram represents the true model m_0. Bottom panel: the dendrograms associated with $m \in \mathcal{M}_2 - \{m_0\}$. The dashed blue line indicates the appropriate cutting point for each of the dendrograms.

For the consistency of CLUSBIC, we need to prove the following two conditions,

a) $\lim_{N\to\infty} p_N(m) = 0$ for $m \in \mathcal{M}_1$.

b) $\lim_{N\to\infty} p_N(m) = 0$ for $m \in \mathcal{M}_2 - \{m_0\}$.

It should be noted that the condition a) is in fact

a*) $\lim_{N\to\infty} N^h p_N(m) = 0$ for $m \in \mathcal{M}_1$, and for any positive h.

The consistency of BIC in model selection has been developed in the literature considering different assumptions, see Shao (1997); Nishi (1984); Rao and Wu (1989). Here, we aim to

prove the consistency of CLUSBIC in clustering problem, i.e. testing the hypothesis of equation (2.13). Estimating the parameters under the null hypothesis in equation (2.13) leads to constrained least squares. Consequently,

$$\begin{aligned}
\tilde{\mathbf{b}} &= \hat{\mathbf{b}} - (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}^\top[\mathbf{T}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}^\top]^{-1}\mathbf{T}\hat{\mathbf{b}},\\
\hat{\mathbf{y}} &= \mathbf{X}\tilde{\mathbf{b}}\\
&= \{\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top - \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}^\top[\mathbf{T}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}^\top]^{-1}\mathbf{T}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\}\mathbf{y}\\
&= \{\mathbf{Q}_1 - \mathbf{Q}_2\}\mathbf{y}\\
&= \mathbf{Q}\mathbf{y}.
\end{aligned}$$

The matrix $\mathbf{Q}$ is the projection matrix which is symmetric and idempotent, see the proof of Lemma 2.2 for the details. The estimate of variance matrix for model m is,

$$\hat{\sigma^2}(m) = \frac{\mathbf{y}^\top\{\mathbf{I} - \mathbf{Q}(m)\}\mathbf{y}}{N}.$$

We take the same line of attack as Nishi (1984) by considering K and D to be fixed. The two following assumptions are required for the proof,

1. $\mathbf{X}^\top\mathbf{X}$ is positive definite.
2. $\mathbf{H} = \lim_{N\to\infty}\frac{\mathbf{X}^\top\mathbf{X}}{N}$ is positive definite.

The validity of the two assumptions relies on the validity of the following two assumptions for all models, i.e. $\forall i \in \{1, 2, \ldots, D\}$

1. $\mathbf{\Theta}_i^\top\mathbf{\Theta}_i$ is positive definite.
2. $\mathbf{H}_i = \lim_{N_i\to\infty}\frac{\mathbf{\Theta}_i^\top\mathbf{\Theta}_i}{N_i}$ is positive definite.

In case where the error terms are distributed according to the Gaussian distribution, one can easily show that the CLUSBIC has the following form, similar to BIC, see Priestley (1982),

$$\begin{aligned}
\text{CLUSBIC}(m) &= N\log\hat{\sigma^2}(m) + K_{\mathbf{d}}(m)\log N\\
&= N\log\frac{\mathbf{b}^\top\mathbf{X}^\top\{\mathbf{I} - \mathbf{Q}(m)\}\mathbf{X}\mathbf{b} + \mathbf{e}^\top\{\mathbf{I} - \mathbf{Q}(m)\}\mathbf{e} + 2\mathbf{e}^\top\{\mathbf{I} - \mathbf{Q}(m)\}\mathbf{X}\mathbf{b}}{N}\\
&\quad + K_{\mathbf{d}}(m)\log N. \qquad (2.22)
\end{aligned}$$

Here, we provide some supplementary technical materials useful in the proof of the following lemmas.

(i) The column space of a matrix $\mathbf{A}$ is denoted by $\mathrm{C}(\mathbf{A})$, and defined as the space spanned by the columns of $\mathbf{A}$.

(ii) The rank of $\mathbf{A}$ is defined to be the dimension of $\mathrm{C}(\mathbf{A})$, $\dim\{\mathrm{C}(\mathbf{A})\}$, i.e. the number of linearly independent columns of $\mathbf{A}$.

$$\mathrm{rank}(\mathbf{A}) = \mathrm{rank}(\mathbf{A}^\top) = \mathrm{rank}(\mathbf{A}^\top\mathbf{A}) = \mathrm{rank}(\mathbf{A}\mathbf{A}^\top).$$

(iii) Orthogonal complement of the sub-space is defined as

$$\mathbf{V}^\perp = \{\mathbf{v} \in \mathbf{R}^n \mid \mathbf{v} \perp \mathbf{V}\}.$$

(iv) If $\mathbf{V} \subset \mathbf{W}$, then $\mathbf{V}^\perp \cap \mathbf{W} = \{\mathbf{v} \in \mathbf{W} \mid \mathbf{v} \perp \mathbf{V}\}$ is called the orthogonal complement of $\mathbf{V}$ with respect to $\mathbf{W}$.

$$\mathrm{rank}(\mathbf{W}) = \mathrm{rank}(\mathbf{V}) + \mathrm{rank}(\mathbf{V}^\perp \cap \mathbf{W}).$$

(v) $\mathbf{Q}_\mathbf{X} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ is called projection matrix onto $\mathrm{C}(\mathbf{X})$. The matrix $\mathbf{Q}_\mathbf{X}$ is symmetric ($\mathbf{Q}_\mathbf{X} = \mathbf{Q}_\mathbf{X}^\top$) and idempotent ($\mathbf{Q}_\mathbf{X}\mathbf{Q}_\mathbf{X} = \mathbf{Q}_\mathbf{X}$).

(vi) If $\mathbf{Q}_\mathbf{X}^1$ and $\mathbf{Q}_\mathbf{X}^2$ are projection matrices with $\mathrm{C}(\mathbf{Q}_\mathbf{X}^1) \subset \mathrm{C}(\mathbf{Q}_\mathbf{X}^2)$, then $\mathbf{Q}_\mathbf{X}^2 - \mathbf{Q}_\mathbf{X}^1$ is also a projection matrix onto the orthogonal complement of $\mathrm{C}(\mathbf{Q}_\mathbf{X}^1)$ with respect to $\mathrm{C}(\mathbf{Q}_\mathbf{X}^2)$.

(vii) Let $\mathbf{A}$ be $k \times k$ matrix of constants and $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$. If $\mathbf{A}$ is idempotent with rank p, then

$$\frac{\mathbf{y}^\top\mathbf{A}\mathbf{y}}{\sigma^2} \sim \chi^2_{p,\lambda},$$

where $\lambda = \frac{\boldsymbol{\mu}^\top\mathbf{A}\boldsymbol{\mu}}{\sigma^2}$.

Lemma 2.1. For $m \in \mathcal{M}_1$, and for any positive h, $\lim_{N\to\infty} N^h p_N(m) = 0$, using CLUSBIC.

Proof. In order to prove the lemma, we work with the difference between the CLUSBIC of the true model and any arbitrary model in $\mathcal{M}_1$. We decompose this difference into several random variables. Taking into account the properties of multivariate Gaussian distribution and the quadratic forms from the same family, we proceed with the proof. Let $m \in \mathcal{M}_1$,

$$\begin{aligned} p_N(m) &= \Pr\{\mathrm{CLUSBIC}(m) < \mathrm{CLUSBIC}(m^*); m^* \in \mathcal{M}\} \\ &\leq \Pr\{\mathrm{CLUSBIC}(m) < \mathrm{CLUSBIC}(m_0)\} \\ &= \Pr\{X + Y_N + N^{\frac{1}{2}}c_N - \sigma^2(\lambda_N N)^{-\frac{1}{2}}b_N \leq Z_N\}, \end{aligned} \tag{2.23}$$

where

$$
\begin{aligned}
X &= 2(\lambda_N N)^{-\frac{1}{2}} \mathbf{e}^\top \mathbf{Q}^* \mathbf{X}\mathbf{b}, \\
Y_N &= (\lambda_N N)^{-\frac{1}{2}} \mathbf{e}^\top \mathbf{Q}^* \mathbf{e}, \\
Z_N &= b_N (\lambda_N N)^{-\frac{1}{2}} N^{-1} [\mathbf{e}^\top \{\mathbf{I} - \mathbf{Q}(m_0)\}\mathbf{e} - \sigma^2 N], \\
c_N &= \lambda_N^{-\frac{1}{2}} N^{-1} \mathbf{b}^\top \mathbf{X}^\top \mathbf{Q}^* \mathbf{X}\mathbf{b}, \\
\lambda_N &= 4\sigma^2 N^{-1} \mathbf{b}^\top \mathbf{X}^\top \mathbf{Q}^{*^2} \mathbf{X}\mathbf{b}, \\
b_N &= N \left\{ \exp\left(\frac{\log(N)\mathrm{p}}{N} \right) - 1 \right\},
\end{aligned}
$$

and $\mathrm{p} = K_{\mathbf{d}}(m_0) - K_{\mathbf{d}}(m)$, $\mathbf{Q}^* = \mathbf{Q}(m_0) - \mathbf{Q}(m)$. Since $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, the following properties can be easily verified.

1. $$\mathbf{e}^\top \mathbf{Q}^* \mathbf{X}\mathbf{b} \sim \mathcal{N}(0, \sigma^2 \mathbf{b}^T \mathbf{X}^\top \mathbf{Q}^{*^2} \mathbf{X}\mathbf{b}) \Longrightarrow X \sim \mathcal{N}(0, 1).$$

2. Y_N is a quadratic form from the Gaussian distribution.

3. Since $\{\mathbf{I} - \mathbf{Q}(m_0)\}$ is a symmetric, idempotent matrix,

   $$\frac{\mathbf{e}^\top \{\mathbf{I} - \mathbf{Q}(m_0)\}\mathbf{e}}{\sigma^2} \sim \chi^2_{N-[D-q(m_0)]K},$$

   where χ^2_b is the chi-squared distribution with b degrees of freedom, see the proof of Lemma 2.2 for the details.

The validity of equation (2.23) can be easily verified through the definition of CLUSBIC in equation (2.22). By the Fréchet inequality,

$$\max\{0, p(A_1) + \ldots + p(A_n) - (n-1)\} \leq p(A_1 \cap \ldots \cap A_n) \leq \min\{p(A_1), \ldots, p(A_n)\},$$

where A_i's are some events, the equation (2.23) is bounded by

$$\Pr(X \leq -N^{\frac{1}{2}} c_N + \sigma^2 (\lambda_N N)^{-\frac{1}{2}} b_N + 2N^{\frac{1}{4}}) + p(-Y_N > N^{\frac{1}{4}}) + p(Z_N > N^{\frac{1}{4}}). \tag{2.24}$$

Using the assumptions 1 and 2,

$$\begin{aligned}\lim_{N\to\infty}\lambda_N &= 4\sigma^2(\mathbf{H}^{\frac{1}{2}}\mathbf{b})^\top\{\mathbf{Q}_H^*\}^2\mathbf{H}^{\frac{1}{2}}\mathbf{b},\\ &= \lambda,\\ \lim_{N\to\infty}c_N &= \lambda^{-\frac{1}{2}}(\mathbf{H}^{\frac{1}{2}}\mathbf{b})^\top\mathbf{Q}_H^*\mathbf{H}^{\frac{1}{2}}\mathbf{b}\\ &= c,\\ b_N &= \mathcal{O}(\log N),\end{aligned}$$

where $\mathbf{Q}_H^* = \mathbf{Q}_H(m_0) - \mathbf{Q}_H(m)$, and

$$\begin{aligned}\mathbf{Q}_H(m) = \mathbf{H}^{\frac{1}{2}}(m_0)\{\mathbf{H}(m_0)\}^{-1}\mathbf{H}^{\frac{1}{2}}(m_0) - \mathbf{H}^{\frac{1}{2}}(m_0)\{\mathbf{H}(m_0)\}^{-1}\mathbf{T}^\top(\mathbf{T}\{\mathbf{H}(m_0)\}^{-1}\mathbf{T}^\top)^{-1}\mathbf{T}\\ \{\mathbf{H}(m_0)\}^{-1}\mathbf{H}^{\frac{1}{2}}(m_0),\ \text{ for } m\in\mathcal{M}.\end{aligned}$$

Let $d_N = c_N - 2N^{-\frac{1}{4}} - \sigma^2(\lambda_N)^{-\frac{1}{2}}N^{-1}b_N$. One can easily show that $d_N = \mathcal{O}(1)$ as $\lim_{N\to\infty}c_N = c$ and $\sigma^2(\lambda_N)^{-\frac{1}{2}}N^{-1}b_N = o(1)$. Using the characteristics of the standard Gaussian distribution function,

$$\begin{aligned}\Pr(X \le -N^{\frac{1}{2}}c_N + \sigma^2(\lambda_N N)^{-\frac{1}{2}}b_N + 2N^{\frac{1}{4}}) &= p(X \le -N^{\frac{1}{2}}d_N)\\ &\le N^{-\frac{1}{2}}d_N^{-1}\phi(N^{\frac{1}{2}}d_N)\\ &= o\left(\exp\left\{-\frac{Nc_N^2}{2}\right\}\right),\end{aligned} \tag{2.25}$$

where $\phi(.)$ is the density function of the standard Gaussian distribution.

Given that Y_N is a quadratic form, using the definition of moment generating functions for quadratic forms (Mathai and Provost, 1992), we have

$$\begin{aligned}\Pr(-Y_N > N^{\frac{1}{4}}) &\le \exp\{-N^{\frac{1}{4}}\}\mathrm{E}(\exp\{-Y_N\}),\\ &= \exp\{-N^{\frac{1}{4}}\}|\mathbf{I} + 2\sigma^2(N\lambda_N)^{-\frac{1}{2}}\mathbf{Q}^*|^{-\frac{1}{2}}\\ &= o(\exp\{-N^{\frac{1}{4}}\}).\end{aligned} \tag{2.26}$$

By the property 3,

$$\begin{aligned}\Pr(Z_N > N^{\frac{1}{4}}) &\le \exp\{-N^{\frac{1}{4}}\}\mathrm{E}(\exp\{Z_N\}),\\ &= \exp\{-N^{\frac{1}{4}} - \sigma^2 b_N(\lambda_N N)^{-\frac{1}{2}}\}[1 - 2\sigma^2 b_N\lambda_N^{-\frac{1}{2}}N^{-\frac{3}{2}}]^{-\frac{N-K_{\mathbf{d}}(m_0)}{2}}\\ &= o(\exp\{-N^{\frac{1}{4}}\}).\end{aligned} \tag{2.27}$$

The equations (2.25), (2.26) and (2.27) complete the proof. □

Lemma 2.2. For $m \in \mathcal{M}_2 - \{m_0\}$, $\lim_{N\to\infty} p_N(m) = 0$, using CLUSBIC.

Proof. Here, we follow a similar approach as Lemma 2.1. For $m \in \mathcal{M}_2 - \{m_0\}$,

$$\begin{aligned} p_N(m) &\leq \Pr\{\text{CLUSBIC}(m) < \text{CLUSBIC}(m_0)\} \\ &= \Pr(\chi \geq N^{-1} b_N \chi_N) \\ &\leq \Pr\left(\chi \geq b_N \left[1 - \frac{1}{\sqrt{\log N}}\right]\right) + \Pr\left(\chi_N \leq N\left[1 - \frac{1}{\sqrt{\log N}}\right]\right), \end{aligned} \tag{2.28}$$

where

$$\begin{aligned} \chi &= \frac{N\{\hat{\sigma^2}(m_0) - \hat{\sigma^2}(m)\}}{\sigma^2} = \frac{\mathbf{e}^\top\{\mathbf{Q}(m) - \mathbf{Q}(m_0)\}\mathbf{e}}{\sigma^2} \\ \chi_N &= \frac{N\hat{\sigma^2}(m)}{\sigma^2} = \frac{\mathbf{e}^\top\{\mathbf{I} - \mathbf{Q}(m)\}\mathbf{e}}{\sigma^2}. \end{aligned}$$

By definition, we have

$$\begin{aligned} \mathbf{Q}(m) - \mathbf{Q}(m_0) &= \mathbf{Q}_1 - \mathbf{Q}_2(m) - [\mathbf{Q}_1 - \mathbf{Q}_2(m_0)] \\ &= \mathbf{Q}_2(m_0) - \mathbf{Q}_2(m) \\ &= \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}_{m_0}^\top[\mathbf{T}_{m_0}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}_{m_0}^\top]^{-1}\mathbf{T}_{m_0}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top \\ &\quad - \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}_m^\top[\mathbf{T}_m(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}_m^\top]^{-1}\mathbf{T}_m(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top. \end{aligned}$$

Under H_0, for an arbitrary model, we have $\mathbf{Tb} = \mathbf{0}$,

$$\begin{aligned} \implies \mathbf{T}(\mathbf{X}^\top\mathbf{X})^{-1}(\mathbf{X}^\top\mathbf{X})\mathbf{b} &= \mathbf{0} \\ \mathbf{T}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\mu} &= \mathbf{0} \\ \mathbf{T^*}^\top\boldsymbol{\mu} &= \mathbf{0}. \end{aligned}$$

Under H_0, $\boldsymbol{\mu} \in \text{C}(\mathbf{X}) = \mathbf{V}$ and $\boldsymbol{\mu} \perp \text{C}(\mathbf{T}^*)$, or $\boldsymbol{\mu} \in \text{C}(\mathbf{T}^*)^\perp \cap \text{C}(\mathbf{X}) = \mathbf{V}_0$ which is the orthogonal complement of $\text{C}(\mathbf{T}^*)$ with respect to $\text{C}(\mathbf{X})$.

$$\text{rank}(\mathbf{T}^*) = \text{rank}(\mathbf{T^*}^\top) \geq \text{rank}(\mathbf{T^*}^\top\mathbf{X}) = \text{rank}(\mathbf{T}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}) = \text{rank}(\mathbf{T}) = qK,$$

$$\text{rank}(\mathbf{T}^*) = \text{rank}(\mathbf{T^*}^\top\mathbf{T}^*) = \text{rank}(\mathbf{T}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}^\top) \leq qK.$$

Therefore, $\text{rank}(\mathbf{T}^*) = qK$. By definition of projection matrix (v),

$$\mathbf{Q}_{\mathbf{V}_0} = \mathbf{Q}_{\text{C}(\mathbf{X})} - \mathbf{Q}_{\text{C}(\mathbf{T}^*)}.$$

$$\dim(\mathbf{V}_0) = \dim\{\text{C}(\mathbf{X})\} - \dim\{\text{C}(\mathbf{T}^*)\} = \text{rank}(\mathbf{X}) - \text{rank}(\mathbf{T}^*) = DK - qK. \tag{2.29}$$

By the property (vi), $\mathbf{Q}(m) - \mathbf{Q}(m_0)$ is a projection matrix. Thus, using equation (2.29), χ has chi-squared distribution with p degrees of freedom,

$$\begin{aligned} \text{p} = \text{rank}(\mathbf{Q}(m) - \mathbf{Q}(m_0)) &= DK - q_m K - [DK - q_{m_0} K] \\ &= [q_{m_0} - q_m]K > 0, \end{aligned}$$

since $q_{m_0} > q_m$. Similarly, χ_N has chi-squared distribution with $N - [DK - q_m K]$ degrees of freedom.

Back to equation (2.28), since $\lim_{N\to\infty} b_N[1 - \frac{1}{\sqrt{\log N}}] = \infty$,

$$\Pr\left(\chi \geq b_N\left[1 - \frac{1}{\sqrt{\log N}}\right]\right) = o(1).$$

For the second term in equation (2.28), one can show that

$$\Pr\left(\chi_N \leq N\left\{1 - \frac{1}{\sqrt{\log N}}\right\}\right) \leq \exp\left\{-\frac{1}{4}\frac{N}{\log N}\right\},$$

using an inequality on chi-squared distribution, see Shibata (1981). This completes the proof. □

In the following two theorems, we show, first, that the CLUSBIC is a consistent measure for Gaussian models and then extend the consistency for any general model regardless of distributional assumptions.

Theorem 2.2. The CLUSBIC is a consistent clustering measure for Gaussian models.

Proof. The equation (2.21) follows from Lemma 2.1 and Lemma 2.2.

The risk, or expected loss, for the model is

$$\begin{aligned} \text{R}_N &= \text{E}||\mathbf{X}\mathbf{b} - \mathbf{X}\tilde{\mathbf{b}}||^2 \\ &= \mathbf{b}^\top \mathbf{X}^\top \{\mathbf{I} - \mathbf{Q}(m)\}\mathbf{X}\mathbf{b}.p_N(m) + \text{E}[\mathbf{e}^\top \mathbf{Q}(m)\mathbf{e}.\text{I}_{\hat{m}=m}] \\ &= A_1 + A_2, \end{aligned}$$

where $\mathrm{I}_{\hat{m}=m}$ is the indicator function. By the Cauchy-Schwartz's inequality,

$$\begin{aligned} A_2 &\leq \sqrt{\mathrm{E}\{\mathbf{e}^\top \mathbf{Q}(m)\mathbf{e}\}^2}\sqrt{p_N(m)} \\ &= \sigma^2\sqrt{2(D-q_m)K + (D-q_m)^2K^2}\sqrt{p_N(m)}. \end{aligned}$$

For $m \in \mathcal{M}_1$,

$$\lim_{N\to\infty} \frac{1}{N}\mathbf{b}^\top \mathbf{X}^\top\{\mathbf{I} - \mathbf{Q}(m)\}\mathbf{X}\mathbf{b} = \mathbf{b}^\top\{\mathbf{H} - \mathbf{H}^{\frac{1}{2}^\top}\mathbf{Q}_H(m)\mathbf{H}^{\frac{1}{2}}\}\mathbf{b} > 0.$$

Equivalently,

$$\mathbf{b}^\top \mathbf{X}^\top\{\mathbf{I} - \mathbf{Q}(m)\}\mathbf{X}\mathbf{b} = \mathcal{O}(N).$$

Therefore, A_1 and A_2 tend to 0 as $N \to \infty$ by condition (a).

For $m \in \mathcal{M}_2 - \{m_0\}$, $A_2 \to 0$ as $N \to \infty$ by condition (b). In addition,

$$\begin{aligned} A_1 &= \mathbf{b}^\top \mathbf{X}^\top\{\mathbf{I} - \mathbf{Q}(m)\}\mathbf{X}\mathbf{b} \\ &= \mathbf{b}^\top \mathbf{X}^\top\{\mathbf{I} - \mathbf{Q}_1 + \mathbf{Q}_2(m)\}\mathbf{X}\mathbf{b}, \\ &= \mathbf{b}^\top \mathbf{X}^\top\{\mathbf{Q}_2(m)\}\mathbf{X}\mathbf{b} \\ &= \mathbf{b}^\top \mathbf{T}_m^\top[\mathbf{T}_m(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{T}_m^\top]^{-1}\mathbf{T}_m\mathbf{b} \\ &= 0, \end{aligned}$$

since $\mathbf{T}_m\mathbf{b} = \mathbf{0}$ for models in $\mathcal{M}_2$.

Consequently, $\lim_{N\to\infty} \mathrm{R}_N = 0$ for both $m \in \mathcal{M}_1$ and $m \in \mathcal{M}_2 - \{m_0\}$. Therefore, if conditions (a) and (b) are satisfied,

$$\lim_{N\to\infty} \mathrm{R}_N = \lim_{N\to\infty} \mathrm{R}_N(m_0) = \sigma^2[D - q_{m_0}]K.$$

This completes the proof of Theorem 2.2. □

Next, we extend Theorem 2.2 to non-Gaussian models, showing that the consistency of the CLUSBIC is still preserved. In the above proof, we relied mostly on properties of the Guassian distribution. The proof of the following result relies on a quadratic approximation to the logarithm of the likelihood suggested by Hong and Preston (2012).

Theorem 2.3. The CLUSBIC is a consistent clustering measure.

Proof. The CLUSBIC for non-Gaussian models has the following form,

$$\mathrm{CLUSBIC}(m) = -2\log\{p_m(\mathbf{y} \mid \tilde{\mathbf{b}})\} + K_{\mathbf{d}}(m)\log(N).$$

We mainly need to show that $\lim_{N\to\infty} p_N(m) = 0$ for $m \in \mathcal{M}_1$ and $m \in \mathcal{M}_2 - \{m_0\}$. In other words, one needs to show that

$$\lim_{N\to\infty} \Pr[\text{CLUSBIC}(m_0) < \text{CLUSBIC}(m)] = 1,$$

for $m \in \mathcal{M}_1$ and $m \in \mathcal{M}_2 - \{m_0\}$.

$$\begin{aligned}
\text{CLUSBIC}(m_0) - \text{CLUSBIC}(m) &= -2\log\{p_{m_0}(\mathbf{y} \mid \tilde{\mathbf{b}})\} + K_{\mathbf{d}}(m_0)\log(N) \\
&\quad + 2\log\{p_m(\mathbf{y} \mid \tilde{\mathbf{b}})\} - K_{\mathbf{d}}(m)\log(N) \\
&= -2[\log\{p_{m_0}(\mathbf{y} \mid \tilde{\mathbf{b}})\} - \log\{p_m(\mathbf{y} \mid \tilde{\mathbf{b}})\}] \\
&\quad + [K_{\mathbf{d}}(m_0) - K_{\mathbf{d}}(m)]\log(N). \qquad (2.30)
\end{aligned}$$

For any $m \in \mathcal{M}$ and the true value of parameter $\mathbf{b}_0$, one can write the following decomposition

$$\begin{aligned}
\log\{p_m(\mathbf{y} \mid \tilde{\mathbf{b}})\} &= \underbrace{\log\{p_m(\mathbf{y} \mid \tilde{\mathbf{b}})\} - \log\{p_m(\mathbf{y} \mid \mathbf{b}_0)\}}_{①} \\
&\quad + \underbrace{\log\{p_m(\mathbf{y} \mid \mathbf{b}_0)\} - N\mathrm{E}\left(\log\{p_m(y \mid \mathbf{b}_0)\}\right)}_{②} \\
&\quad + \underbrace{N\mathrm{E}\left(\log\{p_m(y \mid \mathbf{b}_0)\}\right)}_{③}. \qquad (2.31)
\end{aligned}$$

Applying the second order Taylor expansion to $\log\{p_m(\mathbf{y} \mid \mathbf{b}_0)\}$ at the point $\tilde{\mathbf{b}}$, equation (2.19),

$$\begin{aligned}
\log\{p_m(\mathbf{y} \mid \tilde{\mathbf{b}})\} - \log\{p_m(\mathbf{y} \mid \mathbf{b}_0)\} &= -\frac{1}{2}(\tilde{\mathbf{b}} - \mathbf{b}_0)^\top \frac{\partial^2 \log\{p_m(\mathbf{y} \mid \mathbf{b})\}}{\partial\mathbf{b}\partial\mathbf{b}^\top}\bigg|_{\mathbf{b}=\bar{\mathbf{b}}} (\tilde{\mathbf{b}} - \mathbf{b}_0) \\
&= -\frac{1}{2}\sqrt{N}(\tilde{\mathbf{b}} - \mathbf{b}_0)^\top \frac{1}{N}\frac{\partial^2 \log\{p_m(\mathbf{y} \mid \mathbf{b})\}}{\partial\mathbf{b}\partial\mathbf{b}^\top}\bigg|_{\mathbf{b}=\bar{\mathbf{b}}} \\
&\quad \times \sqrt{N}(\tilde{\mathbf{b}} - \mathbf{b}_0).
\end{aligned}$$

Under the usual regularity conditions, see Sen and Singer (1994, page 209), we have

$$\frac{1}{N}\frac{\partial^2 \log\{p_m(\mathbf{y} \mid \mathbf{b})\}}{\partial\mathbf{b}\partial\mathbf{b}^\top}\bigg|_{\mathbf{b}=\mathbf{b}_0} \xrightarrow{p} \mathrm{E}\{\frac{\partial^2 \log\{p_m(\mathbf{y} \mid \mathbf{b})\}}{\partial\mathbf{b}\partial\mathbf{b}^\top}\} = -\mathrm{I}(\mathbf{b}_0), \qquad (2.32)$$

$$\frac{1}{\sqrt{N}}\frac{\partial \log\{p(\mathbf{y} \mid \mathbf{b}, \sigma^2)\}}{\partial\mathbf{b}}\bigg|_{\mathbf{b}=\mathbf{b}_0} \xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathrm{I}(\mathbf{b}_0)), \qquad (2.33)$$

where $\mathrm{I}(\mathbf{b}_0)$ is the Fisher information matrix, see Amemiya (1985) for details. On the other

hand, by the definition of constrained optimization problem equations (2.14), and (2.15),

$$\begin{bmatrix} \frac{\partial \log\{p(\mathbf{y}|\mathbf{b})\}}{\partial \mathbf{b}}\Big|_{\mathbf{b}=\tilde{\mathbf{b}}} + \mathbf{T}^\top \boldsymbol{\lambda} \\ \mathbf{T}\tilde{\mathbf{b}} \end{bmatrix} = \mathbf{0}. \tag{2.34}$$

Expanding the score function around the point $\mathbf{b}_0$, and $\boldsymbol{\lambda}_0 = \mathbf{0}$, we have

$$\begin{aligned} \mathbf{0} &= \begin{bmatrix} \frac{\partial \log\{p(\mathbf{y}|\mathbf{b})\}}{\partial \mathbf{b}}\Big|_{\mathbf{b}=\mathbf{b}_0} + \mathbf{T}^\top \boldsymbol{\lambda}_0 \\ \mathbf{T}\mathbf{b}_0 \end{bmatrix} + \begin{bmatrix} \frac{\partial^2 \log\{p(\mathbf{y}|\mathbf{b})\}}{\partial \mathbf{b}\partial \mathbf{b}^\top}\Big|_{\mathbf{b}=\bar{\mathbf{b}}} & \mathbf{T}^\top \\ \mathbf{T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{b}} - \mathbf{b}_0 \\ \tilde{\boldsymbol{\lambda}} - \boldsymbol{\lambda}_0 \end{bmatrix} \\ \sqrt{N} \begin{bmatrix} \tilde{\mathbf{b}} - \mathbf{b}_0 \\ \tilde{\boldsymbol{\lambda}} \end{bmatrix} &= -\sqrt{N} \begin{bmatrix} \frac{\partial^2 \log\{p(\mathbf{y}|\mathbf{b})\}}{\partial \mathbf{b}\partial \mathbf{b}^\top}\Big|_{\mathbf{b}=\bar{\mathbf{b}}} & \mathbf{T}^\top \\ \mathbf{T} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \log\{p(\mathbf{y}|\mathbf{b})\}}{\partial \mathbf{b}}\Big|_{\mathbf{b}=\mathbf{b}_0} \\ \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \left(\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{T}(\mathbf{T}^\top \mathbf{A}^{-1}\mathbf{T})^{-1}\mathbf{T}^\top \mathbf{A}^{-1}\right) \frac{1}{\sqrt{N}} \frac{\partial \log\{p(\mathbf{y}|\mathbf{b})\}}{\partial \mathbf{b}}\Big|_{\mathbf{b}=\mathbf{b}_0} \\ (\mathbf{T}^\top \mathbf{A}^{-1}\mathbf{T})^{-1}\mathbf{T}^\top \mathbf{A}^{-1} \frac{1}{\sqrt{N}} \frac{\partial \log\{p(\mathbf{y}|\mathbf{b})\}}{\partial \mathbf{b}}\Big|_{\mathbf{b}=\mathbf{b}_0} \end{bmatrix}, \end{aligned}$$

where $\mathbf{A} = \frac{\partial^2 \log\{p(\mathbf{y}|\mathbf{b})\}}{\partial \mathbf{b}\partial \mathbf{b}^\top}\Big|_{\mathbf{b}=\bar{\mathbf{b}}}$. Taking into account equations (2.32) and (2.33), one can show that

$$\begin{aligned} \sqrt{N}(\tilde{\mathbf{b}} - \mathbf{b}_0) &\xrightarrow{d} \mathcal{N}(\mathbf{0}, \mathrm{I}^{-1}(\mathbf{b}_0) - \mathrm{I}^{-1}(\mathbf{b}_0)\mathbf{T}(\mathbf{T}^\top \mathrm{I}^{-1}(\mathbf{b}_0)\mathbf{T})^{-1}\mathbf{T}^\top \mathrm{I}^{-1}(\mathbf{b}_0)), \\ &\xrightarrow{d} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\tilde{\mathbf{b}}}), \end{aligned} \tag{2.35}$$

where $\mathrm{I}^{-1}(\mathbf{b}_0)$ is the inverse of the Fisher information matrix. By equations (2.35) and (2.32), and the fact that $\mathrm{I}(\mathbf{b}_0)\boldsymbol{\Sigma}_{\tilde{\mathbf{b}}}$ is an idempotent matrix, one can conclude that

$$-\frac{1}{2}\sqrt{N}(\tilde{\mathbf{b}} - \mathbf{b}_0)^\top \frac{1}{N} \frac{\partial^2 \log\{p_m(\mathbf{y} \mid \mathbf{b})\}}{\partial \mathbf{b}\partial \mathbf{b}^\top}\Big|_{\mathbf{b}=\bar{\mathbf{b}}} \sqrt{N}(\tilde{\mathbf{b}} - \mathbf{b}_0) \xrightarrow{d} \frac{1}{2}\chi_{\dim(\mathbf{b}_0)}, \tag{2.36}$$

where χ has chi-squared distribution with $\dim(\mathbf{b}_0)$ degrees of freedom. As the convergence in distribution implies boundedness in probability, component (1) in (2.31) is of order $\mathcal{O}_p(1)$.

As for component (2) in (2.31), by central limit theorem,

$$\frac{1}{\sqrt{N}} \left[\log\{p_m(\mathbf{y} \mid \mathbf{b}_0)\} - N\mathrm{E}(\log\{p_m(y \mid \mathbf{b}_0)\})\right] \xrightarrow{d} N(\mathbf{0}, \boldsymbol{\Sigma}^*), \tag{2.37}$$

where $\boldsymbol{\Sigma}^* = \mathrm{Var}\left\{\frac{1}{\sqrt{N}} \left[\log\{p_m(\mathbf{y} \mid \mathbf{b}_0)\} - N\mathrm{E}(\log\{p_m(y \mid \mathbf{b}_0)\})\right]\right\}$. Accordingly, the component

(2) in (2.31) is of order $\mathcal{O}_p(\sqrt{N})$.

Now coming back to the equation (2.30), for both $m \in \mathcal{M}_1$ and $m \in \mathcal{M}_2 - \{m_0\}$,

$$\begin{aligned}\text{CLUSBIC}(m_0) - \text{CLUSBIC}(m) = -2[\mathcal{O}_p(1) + \mathcal{O}_p(\sqrt{N}) + N[\text{E}\left(\log\{p_{m_0}(y \mid \mathbf{b}_0)\}\right) \\ - \text{E}\left(\log\{p_m(y \mid \mathbf{b}_0)\}\right)] + [K_{\mathbf{d}}(m_0) - K_{\mathbf{d}}(m)]\log(N).\end{aligned} \tag{2.38}$$

Using Jensen's inequality,

$$\begin{aligned}\text{E}\left(\log\{p_m(y \mid \mathbf{b}_0)\}\right) - \text{E}\left(\log\{p_{m_0}(y \mid \mathbf{b}_0)\}\right) &= \text{E}\left(\log\{\frac{p_m(y \mid \mathbf{b}_0)}{p_{m_0}(y \mid \mathbf{b}_0)}\}\right) \\ &\leq \log\left\{\text{E}\left(\frac{p_m(y \mid \mathbf{b}_0)}{p_{m_0}(y \mid \mathbf{b}_0)}\right)\right\} \\ &\leq \log\left\{\int p_m(y \mid \mathbf{b}_0)dy\right\} \\ &\leq 0\end{aligned}$$

and hence

$$\text{CLUSBIC}(m_0) - \text{CLUSBIC}(m) = -\mathcal{O}(N). \tag{2.39}$$

Therefore, as N tends to infinity,

$$\Pr\{\text{CLUSBIC}(m_0) < \text{CLUSBIC}(m)\} = 1$$

for models in both $\mathcal{M}_1$ and $\mathcal{M}_2 - \{m_0\}$. This completes the proof. □

The proof indicates that regardless of the distribution of error terms, the CLUSBIC is a consistent criterion.

2.6 Clustering Prior

Clustering curves using CLUSBIC is asymptorically equivalant to clustering using the postrior probability of grouping $p(\mathbf{d} \mid \mathbf{y})$. The CLUSBIC measure is used for illustrating the asymptotic properties of clustering through $p(\mathbf{d} \mid \mathbf{y})$. Bayesian clustering according to the $p(\mathbf{d} \mid \mathbf{y})$ (equation (2.9)) requires modeling $p(\mathbf{y} \mid \mathbf{d})$ and $p(\mathbf{d})$. Here, we follow the structure suggested in Heard *et al.* (2006). The random vector $\mathbf{d}$ denotes the possible groupings for a set of shapes. Suppose $\mathcal{C}(\mathbf{d})$ is the total number of clusters at each step and $n_1, n_2, \ldots, n_{\mathcal{C}(\mathbf{d})}$ are the total number of shapes in each of the clusters. Suppose that $\mathcal{C}(\mathbf{d})$ is uniformly distributed over the set $\{1, 2, \ldots, D\}$, where D is the total number of shapes and the n_j's, $j = 1, 2, \ldots, D$,

are distributed according to multinomial-Dirichlet with parameter $\boldsymbol{\pi}$,

$$p(\mathbf{d} \mid \boldsymbol{\pi}) = \frac{1}{D} \frac{\Gamma(\sum_{i=1}^{\mathcal{C}(\mathbf{d})} \pi_i) \prod_{i=1}^{\mathcal{C}(\mathbf{d})} \Gamma(n_i + \pi_i)}{\prod_{i=1}^{\mathcal{C}(\mathbf{d})} \Gamma(\pi_i) \Gamma(D + \sum_{i=1}^{\mathcal{C}(\mathbf{d})} \pi_i)}.$$

A uniform setting on the parameter vector $\boldsymbol{\pi}$ leads to

$$p(\mathbf{d}) = \frac{(\mathcal{C}(\mathbf{d}) - 1)! n_1! n_2! \ldots n_{\mathcal{C}(\mathbf{d})}!}{D(D + \mathcal{C}(\mathbf{d}) - 1)!}.$$

The following example illustrates the use of likelihood function in clustering.

Example 2.3. Consider five distinct shapes, $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_5$, and their respective matrices of independent attributes, $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_5$. The agglomerative hierarchical clustering goes through 4 steps for clustering the five shapes. Let $\ell_i = \log\{p(\boldsymbol{r}_i)\}$, and $\ell^{(j)} = \sum_{i=1}^{\mathcal{C}(\mathbf{d})} \ell_{(i)}$ for $i = 1, 2, \ldots, 5$ and $j = 0, 1, \ldots, 4$, where the index j indicates the steps of dendrogram. Allocate each shape to a different group, $\mathbf{d} = (1, 2, \ldots, 5)$, and calculate the likelihood function associated with each of the shapes. The total log-likelihood is $\ell^{(0)} = \sum_{i=1}^{5} \ell_i$, which builds the bottom line of the dendrogram. The steps of hierarchical clustering using the agglomerative method are as follows.

Step 1 Calculating the distance matrix for $\binom{5}{2}$ different combinations in order to find the first possible grouping.

$$\mathbf{C}_1 = \begin{bmatrix} 0 & & & & \\ c_{21} & 0 & & & \\ c_{31} & c_{32} & 0 & & \\ c_{41} & c_{42} & c_{43} & 0 & \\ c_{51} & c_{52} & c_{53} & c_{54} & 0 \end{bmatrix},$$

where the $(i, j)^{th}$ entry of the matrix $\mathbf{C}_1$, $c_{ij} = \ell_i + \ell_j - \ell_{(ij)}$, $i, j = 1, 2, \ldots, 5$, and $\ell_{(ij)}$ is the log-likelihood after combining the observations of group i with group j. The smallest nonzero entry defines the first merging cluster. Let's assume that c_{52} is the smallest which corresponds to $\mathbf{d} = (1, 2, 3, 4, 2)$. Therefore, the individuals 2 and 5 are joined together to form a two-member cluster. If $\ell^{(2)} = \ell_{(52)} + \ell_1 + \ell_3 + \ell_4$, then the first cluster is joined at the height $h_1 = |\ell^{(1)} - \ell^{(0)}|$ on the dendrogram. If the sign of the $\ell^{(2)} - \ell^{(1)}$ is positive then the two individuals are grouped with a black solid line. Otherwise, they are grouped with a gray line indicating the decrease of marginal probability over the agglomerative path.

Step 2 Recalculate the distance matrix with the four existing groups: 1, (25), 3, and 4

where by (25) we mean a cluster that is formed by the individuals 2 and 5,

$$\mathbf{C}_2 = \begin{bmatrix} 0 & & & \\ c_{(25)1} & 0 & & \\ c_{31} & c_{3(25)} & 0 & \\ c_{41} & c_{4(25)} & c_{43} & 0 \end{bmatrix}.$$

Assuming that $c_{(25)1}$ is the smallest, the individual 1 is joined with the cluster (25), $\mathbf{d} = (1, 1, 2, 3, 1)$, at height $h_2 = |\ell^{(2)} - \ell^{(1)}| + |\ell^{(1)} - \ell^{(0)}|$, where $\ell^{(2)} = \ell_{(251)} + \ell_3 + \ell_4$.

Step 3 Recalculate the distance matrix with the three existing groups: (125), 3, and 4,

$$\mathbf{C}_3 = \begin{bmatrix} 0 & & \\ c_{3(251)} & 0 & \\ c_{4(251)} & c_{43} & 0 \end{bmatrix}.$$

Suppose c_{43} is the smallest, but $\ell^{(3)} - \ell^{(2)}$ is negative. In this case, the individual 3 is joined with the individual 4, $\mathbf{d} = (1, 1, 2, 2, 1)$, at height $h_3 = |\ell^{(3)} - \ell^{(2)}| + |\ell^{(2)} - \ell^{(1)}| + |\ell^{(1)} - \ell^{(0)}|$, but it is shown by a gray line in the dendrogram.

Step 4 In this step, the distance matrix looks as follows for the two existing groups: (125), and (34),

$$\mathbf{C}_4 = \begin{bmatrix} 0 & \\ c_{(34)(251)} & 0 \end{bmatrix}.$$

Here, all the individuals are grouped together, $\mathbf{d} = (1, 1, 1, 1, 1)$, at height $h_4 = |\ell^{(4)} - \ell^{(3)}| + \ldots + |\ell^{(1)} - \ell^{(0)}|$. Assuming that $\ell^{(4)} - \ell^{(3)}$ is negative, the two groups are merged together by a gray line, see Figure 2.2.

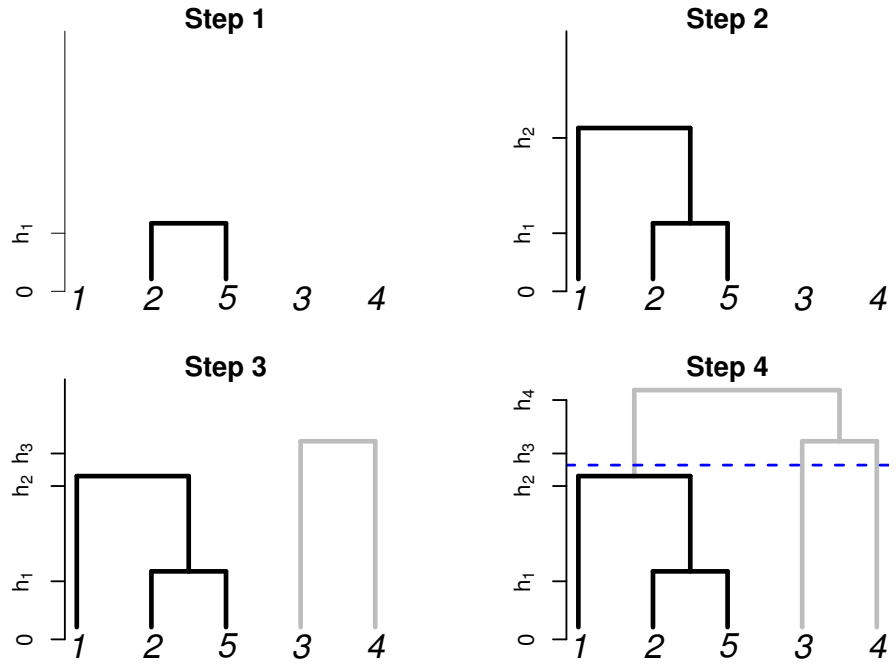

Figure 2.2 The step by step illustration of hierarchical agglomerative clustering using likelihood functions. The solid black and the gray lines show the improvement and degradation in total likelihood in comparison to previous step. The dashed blue line shows the maximum a posteriori cutting point for the dendrogram.

As it is visualized in Figure 2.2, only two clusters are shown by the solid black lines. In other words, only these two clusters lead to the increase in the total marginal probability. We use the the sign of $\ell^{(i)} - \ell^{(j)}$ such that $i - j = 1$ as a general hint to decide about the number of clusters. Thus the dendrogram is cut at a height in (h_2, h_3) so that the five shapes of this example are assigned into three distinct clusters of $\{1, 2, 5\}$, $\{3\}$ and $\{4\}$.

2.7 Markov Chain Monte Carlo (MCMC)

In many Bayesian hierarchical models, it is impossible to sample from a density directly, but the density is known up to a normalizing constant. Markov chain Monte Carlo (MCMC) is a method of generating Markov samples that have similar behavior to the independent identically distributed samples drawn from the target distribution in a long run. It is straightforward to introduce Markov chains on the finite state space, e.g. $d^{(i)}$, that can take on the discrete values of $\{d_1, d_2, \ldots, d_D\}$. The stochastic process $d^{(i)}$ is called a Markov chain if

$$p(d^{(i)} \mid d^{(i-1)}, \ldots, d^{(1)}) = p(d^{(i)} \mid d^{(i-1)}).$$

In other words, the evolution of the chain only depends on the current state of the chain. The chain is homogeneous if $p(d^{(i)} \mid d^{(i-1)})$ remains invariant for all i with $\sum_i p(d^{(i)} \mid d^{(i-1)}) = 1$. The chain converges to the target distribution of $p(\mathbf{d})$, if the stochastic transition matrix is irreducible and aperiodic. If there is a positive probability of visiting all other states from any particular state, the transition matrix is irreducible. The aperiodicity refers to the chain not being trapped in any cycles created over the states. The detailed balance or reversibility is a sufficient, but not necessary condition to guarantee the convergence of the chain

$$p(d^{(i)})p(d^{(i-1)} \mid d^{(i)}) = p(d^{(i-1)})p(d^{(i)} \mid d^{(i-1)}).$$

MCMC samplers are irreducible and aperiodic Markov chains with the target distribution as the invariant distribution (Berg, 2005).

2.7.1 Metropolis-Hastings Sampler

The Metropolis-Hastings algorithm is the most popular MCMC method. In Metropolis-Hastings algorithm, one samples a candidate value $\mathbf{d}^*$ given the current value $\mathbf{d}$ according to a proposal distribution $q(\mathbf{d}^* \mid \mathbf{d})$. The Markov chain moves toward $\mathbf{d}^*$ with acceptance probability

$$\mathrm{A}(\mathbf{d}^*, \mathbf{d}) = \min\left\{1, \frac{p(\mathbf{d}^*)q(\mathbf{d} \mid \mathbf{d}^*)}{p(\mathbf{d})q(\mathbf{d}^* \mid \mathbf{d})}\right\},$$

otherwise it remains at $\mathbf{d}$. If the tail of the proposal distribution is too narrow, only a local mode of $p(\mathbf{d})$ might be visited. On the other hand, if its tails are too wide, the rejection rate can be very high which results in high correlation. The pseudo code for the Metropolis-Hastings sampler can be found below,

Metropolis-Hastings

1: Initialize $\mathbf{d}^{(0)}$ $i = 0$ to $T - 1$
2: Sample $u \sim \mathrm{U}_{[0,1]}$.
3: Sample $\mathbf{d}^* \sim q(\mathbf{d}^* \mid \mathbf{d}^{(i)})$.
4: **if** $u < \mathrm{A}(\mathbf{d}^{(i)}, \mathbf{d}^*)$ **then**
5: $\quad \mathbf{d}^{(i+1)} = \mathbf{d}^*$
6: **else**
7: $\quad \mathbf{d}^{(i+1)} = \mathbf{d}^{(i)}$
8: **end if**

where $\mathrm{U}_{[a,b]}$ is a continuous uniform distribution over the interval $[a, b]$.

2.7.2 Gibbs Sampling

Assume $\mathbf{d}$ to be a D-dimensional vector such that full conditional probabilities for its elements, $p(d_j \mid \mathbf{d}_{-j}) = p(d_j \mid d_1, d_2, \ldots, d_{j-1}, d_{j+1}, \ldots, d_D)$, are available. The Gibbs sampler (Geman and Geman, 1984) incorporates the full conditionals into the proposal distribution in the steps of Metropolis-Hastings as follows,

$$q(\mathbf{d}^* \mid \mathbf{d}) = \begin{cases} p(d_j^* \mid \mathbf{d}_{-j}) & \text{if } \mathbf{d}_{-j}^* = \mathbf{d}_{-j}, \\ 0 & \text{otherwise.} \end{cases}$$

The acceptance probability for the proposed distribution is

$$\begin{aligned} \mathrm{A}(\mathbf{d}, \mathbf{d}^*) &= \min\left\{1, \frac{p(\mathbf{d}^*)q(\mathbf{d} \mid \mathbf{d}^*)}{p(\mathbf{d})q(\mathbf{d}^* \mid \mathbf{d})}\right\} \\ &= \min\left\{1, \frac{p(\mathbf{d}^*)p(d_j \mid \mathbf{d}_{-j})}{p(\mathbf{d})p(d_j^* \mid \mathbf{d}_{-j}^*)}\right\} \\ &= \min\left\{1, \frac{p(\mathbf{d}_{-j}^*)}{p(\mathbf{d}_{-j})}\right\} = 1. \end{aligned}$$

The pseudo code related to a full scan Gibbs sampler is as follows.

Full Scan Gibbs Sampling

Initialize $\mathbf{d}^{(0)}$
for $i = 0$ to $T - 1$ **do**
  Sample $d_1^{(i+1)} \sim p(d_1 \mid d_2^{(i)}, d_3^{(i)}, \ldots, d_D^{(i)})$.
  Sample $d_2^{(i+1)} \sim p(d_2 \mid d_1^{(i+1)}, d_3^{(i)}, \ldots, d_D^{(i)})$.
  $\vdots$
  Sample $d_j^{(i+1)} \sim p(d_j \mid d_1^{(i+1)}, \ldots, d_{j-1}^{(i+1)}, d_{j+1}^{(i)}, \ldots, d_D^{(i)})$.
  $\vdots$
  Sample $d_D^{(i+1)} \sim p(d_D \mid d_1^{(i+1)}, d_2^{(i+1)}, \ldots, d_{D-1}^{(i+1)})$.
end for

2.7.3 Random Scan Gibbs Sampling

The full scan Gibbs sampler updates all the components, one after the other. The complete cycle over all possible components may lead to a slow convergence. As an alternative to the full scan Gibbs sampling, one can randomly select the component to be updated at each cycle. The pseudo code related to the random scan Gibbs sampler is provided below.

Random Scan Gibbs Sampling

1: Initialize $\mathbf{d}^{(0)}$
2: **for** $i = 0$ to $T - 1$ **do**
3: Draw j uniformly from $\{1, 2, \ldots, D\}$.
4: Sample $d_j^{(i+1)} \sim p(d_j \mid d_1^{(i)}, \ldots, d_{j-1}^{(i)}, d_{j+1}^{(i)}, \ldots, d_D^{(i)})$.
5: Update $\mathbf{d}^{(i+1)} = (d_1^{(i)}, \ldots, d_{j-1}^{(i)}, d_j^{(i+1)}, d_{j+1}^{(i)}, \ldots, d_D^{(i)})$.
6: **end for**

2.7.4 Split-Merge Gibbs Sampling

The implementation of Gibbs sampling carries no complication, but Gibbs sampler may occasionally suffer from poor mixing. This undesirable behaviour arises from the chain being trapped in a local mode associated with an untrue clustering of data. Jain and Neal (2007) modified the Gibbs sampling procedure by coupling split-merge updates to Metropolis-Hastings at each cycle of sampling. Two different scenarios for split-merge are proposed. The first scenario is based on random split of the subset of data into two separate clusters. The second scenario is an improvement of former scenario in terms of producing more sensible splits of clusters that are more likely to be accepted as a move. Considering $\mathcal{A} = \{1, 2, \ldots, D\}$ to be the set of objects for clustering and $\mathbf{d} = \{d_1, d_2, \ldots, d_D\}$ to be their corresponding set of groupings, the pseudo codes for each of the scenarios are as follows.

Random Split-Merge

1: Select two distinct observations, i and j, uniformly at random.
2: Let $\mathcal{S}$ be the subset $\mathcal{A} \setminus \{i, j\}$ such that its grouping is in either d_i or d_j.
3: **if** $d_i = d_j$ **then**
4: Split the two objects, i and j, into two different groupings as d_i^{split} and d_j^{split} as follows:

- Let d_i^{split} be a new grouping such that $d_i^{\text{split}} \notin \{d_1, \ldots, d_D\}$.
- Let $d_j^{\text{split}} = d_j$
- To each $k \in \mathcal{S}$, independently assign d_i^{split} or d_j^{split} with equal probabilities.
- To each $k \notin \mathcal{S} \cup \{i, j\}$ assign $d_k^{\text{split}} = d_k$.

Evaluate the split proposal by Metropolis-Hastings acceptance probability

$$\begin{aligned} \mathrm{A}(\mathbf{d}^{\text{split}}, \mathbf{d}) &= \min\left\{1, \frac{q(\mathbf{d}^{\text{split}} \mid \mathbf{d}) p(\mathbf{d}^{\text{split}})}{q(\mathbf{d} \mid \mathbf{d}^{\text{split}}) p(\mathbf{d})}\right\} \\ &= \min\left\{1, \frac{p(\mathbf{d}^{\text{split}})}{2^{n_{d_i^{\text{split}}} + n_{d_j^{\text{split}}} - 2} p(\mathbf{d})}\right\}, \end{aligned}$$

where $n_{d_i^{split}}$ and $n_{d_j^{split}}$ denote the number of observations that belong to each split mixture component. If the proposal is accepted, $\mathbf{d}^{\text{split}}$ becomes the next state in the Markov chain and in case of rejection, the chain remains unchanged.
5: **end if**
6: **if** $d_i \neq d_j$ **then**
7: Merge the two objects, i and j, into one grouping as $\mathbf{d}^{\text{merge}}$ as follows:

- Let $d_i^{\text{merge}} = d_j$.
- Let $d_j^{\text{merge}} = d_j$.
- To each $k \in \mathcal{S}$, assign $d_k^{\text{merge}} = d_j$.
- To each $k \notin \mathcal{S} \cup \{i, j\}$ assign $d_k^{\text{merge}} = d_k$.
- Evaluate the merge proposal by Metropolis-Hastings acceptance probability

$$\begin{aligned} \mathrm{A}(\mathbf{d}^{\text{merge}}, \mathbf{d}) &= \min\left\{1, \frac{q(\mathbf{d}^{\text{merge}} \mid \mathbf{d}) p(\mathbf{d}^{\text{merge}})}{q(\mathbf{d} \mid \mathbf{d}^{\text{merge}}) p(\mathbf{d})}\right\} \\ &= \min\left\{1, \frac{2^{n_{d_i} + n_{d_j} - 2} p(\mathbf{d}^{\text{merge}})}{p(\mathbf{d})}\right\}. \end{aligned}$$

If the proposal is accepted, $\mathbf{d}^{\text{merge}}$ becomes the next state in the Markov chain and in case of rejection, the chain remains unchanged.

8: **end if**
9: Repeat the above steps for T cycles.

Restricted Gibbs Sampling Split-Merge

1: Select two distinct observations, i and j, uniformly at random.
2: Let $\mathcal{S}$ be the subset $\mathcal{A} \setminus \{i, j\}$ such that its grouping is in either d_i or d_j.
3: Define the launch state, $\mathbf{d}^{\text{launch}}$, that will be used to compute Gibbs sampling probabilities.

- Set d_i^{launch} to a new component such that $d_i^{\text{launch}} \notin \{d_1, \ldots, d_D\}$ if $d_i = d_j$, else $d_i^{\text{launch}} = d_i$.
- Set $d_j^{\text{launch}} = d_j$.
- To each $k \in \mathcal{S}$ assign either d_i^{launch} or d_j^{launch} as follows:
  1. Select an initial state by independently assigning d_i^{launch} or d_j^{launch} with equal probabilities to each of d_k^{launch}.
  2. Modify $\mathbf{d}^{\text{launch}}$ by performing t intermediate restricted Gibbs sampling scans.

4: **if** $d_i = d_j$ **then**
5: Split the two objects, i and j, into two different groupings as d_i^{split} and d_j^{split} as follows:

- Let $d_i^{\text{split}} = d_i^{\text{launch}}$.
- Let $d_j^{\text{split}} = d_j^{\text{launch}}$
- To each $k \in \mathcal{S}$, assign either d_i^{split} or d_j^{split} by running one Gibbs sampling scan from the launch state $\mathbf{d}^{\text{launch}}$.
- To each $k \notin \mathcal{S} \cup \{i, j\}$ assign $d_k^{\text{split}} = d_k$.
- Set $q(\mathbf{d}^{\text{split}} \mid \mathbf{d})$ to the transition probability of Gibbs sampling from the launch state to the final proposed state of $\mathbf{d}^{\text{split}}$. The Gibbs sampling transition probability is the product over $k \in \mathcal{S}$ of the probabilities of setting each $\mathbf{d}_k^{\text{split}}$ to its final value in the final Gibbs sampling scan.

Evaluate the split proposal by Metropolis-Hastings acceptance probability

$$\mathrm{A}(\mathbf{d}^{\text{split}}, \mathbf{d}) = \min\left\{1, \frac{q(\mathbf{d}^{\text{split}} \mid \mathbf{d}) p(\mathbf{d}^{\text{split}})}{q(\mathbf{d} \mid \mathbf{d}^{\text{split}}) p(\mathbf{d})}\right\}.$$

If the proposal is accepted, $\mathbf{d}^{\text{split}}$ becomes the next state in the Markov chain and in case of rejection, the chain remains unchanged.
6: **end if**
7: **if** $d_i \neq d_j$ **then**
8: Merge the two objects, i and j, into one grouping as $\mathbf{d}^{\text{merge}}$ as follows:

- Let $d_i^{\text{merge}} = d_j$.
- Let $d_j^{\text{merge}} = d_j$
- To each $k \in \mathcal{S}$, assign $d_k^{\text{merge}} = d_j$.
- To each $k \notin \mathcal{S} \cup \{i, j\}$ assign $d_k^{\text{merge}} = d_k$.
- Set $q(\mathbf{d} \mid \mathbf{d}^{\text{merge}})$ to the transition probability of Gibbs sampling from the launch state to the original split configuration of $\mathbf{d}$. The Gibbs sampling transition probability is the product over $k \in \mathcal{S}$ of the probabilities of setting each $\mathbf{d}_k$ to its original value in a Gibbs sampling scan from launch state.

Restricted Gibbs Sampling Split-Merge Cont'd

Evaluate the merge proposal by Metropolis-Hastings acceptance probability

$$\mathrm{A}(\mathbf{d}^{\text{merge}}, \mathbf{d}) = \min\left\{1, \frac{q(\mathbf{d}^{\text{merge}} \mid \mathbf{d})p(\mathbf{d}^{\text{merge}})}{q(\mathbf{d} \mid \mathbf{d}^{\text{merge}})p(\mathbf{d})}\right\}.$$

If the proposal is accepted, $\mathbf{d}^{\text{merge}}$ becomes the next state in the Markov chain and in case of rejection, the chain remains unchanged.

9: **end if**

10: Repeat the above steps for T cycles.

The MCMC sampling methods explained above are particularly applicable to the problem of clustering through $p(\mathbf{d} \mid \mathbf{y})$. One may explore the whole space of $\mathbf{d}$ employing MCMC sampler methods to spot the value of $\mathbf{d}$ for which the $p(\mathbf{d} \mid \mathbf{y})$ is maximum. For this purpose, one needs to replace $p(\mathbf{d})$ with $p(\mathbf{d} \mid \mathbf{y})$ in the above samplers.

2.8 Simulation

In order to verify the ability of the proposed method in modeling and clustering the 2D, and 3D shapes, we simulated some data from each model respectively. The results are presented in Section 2.8.1 and Section 2.8.2.

2.8.1 2D Shapes

Here, we simulated set of binary images of simple geometrical objects, Figure 2.3. No noise is included in the boundary data of objects in these shapes. In Figure 2.3, image 5 (image 7) is rotated version of image 4 (image 6) by 90° (180°). As a result, these images represent only five distinct objects. The images need to be aligned to assure orientation-free data extraction using object recognition or image registration methods, see Zitovà and Flusser (2003).

1 2 3 4 5 6 7

Figure 2.3 Binary images of some closed geometrical objects.

First, the coordinate of data pixels on the boundary of the objects in Cartesian coordinates system is extracted. A sample of 500 observations from the boundary of each image is taken for the modeling and clustering purpose. Based on the value of marginal probability for each

object, one can choose the suitable basis function as well as K, the number of terms in the expansion.

Table 2.1 The marginal log-likelihood values when $\mathbf{d} = \{1, 2, \ldots, 7\}$ over different values of basis functions and K for the simulation dataset in Figure 2.3.

$\log\{p(\boldsymbol{r})\}$	Basis Function			
K	Fourier	Circular Harmonics	Wavelets	Smoothing Splines
5	529.94	385.30	395.91	411.07
9	595.50	481.86	553.74	578.84
17	604.17	572.63	567.25	605.25
33	562.22	555.02	504.48	550.37
65	464.57	463.19	405.72	356.22
129	265.64	265.45	208.86	7.65

According to the values of $\log\{p(\boldsymbol{r})\}$ reported in the Table 2.1, it is observed that that Fourier, and smoothing splines basis functions with $K = 17$ are equally suitable for the simulation dataset. In Table 2.1, the increase in number of expansion terms K does not necessary increase the marginal log-likelihood as the marginal probability is being penalized by K, see equation (2.20). After selecting the appropriate basis function, for instance Fourier bases with $K = 17$, a model is fitted to each object in database. The clustering can be performed using the Euclidean distance between the coefficients obtained from each model, $\boldsymbol{\beta}_i$'s.

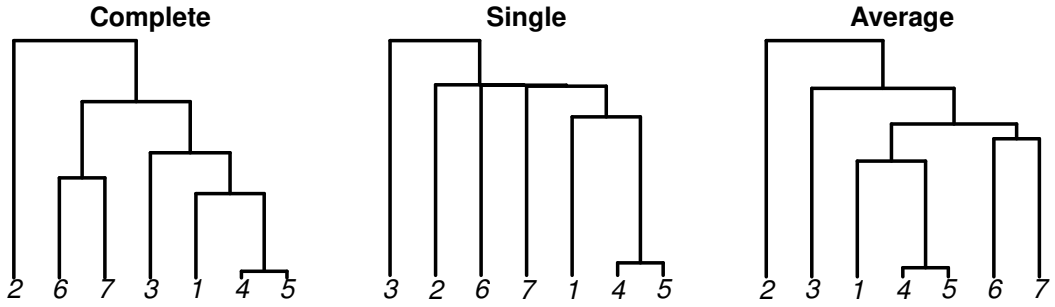

Figure 2.4 Clustering the simulated objects in Figure 2.3 using hierarchical clustering with different metrics over the coefficient of fits, $\boldsymbol{\beta}_i$'s, in each model.

It is expected that the following clusters to be produced, for the simulated shapes, using

any proper method of clustering,

$$\{1\}, \{2\}, \{3\}, \{4, 5\}, \{6, 7\}.$$

Clustering result using different metrics over the coefficients of models fails to produce meaningful trees. There is no cutting point for any of the dendrograms in Figure 2.4 that can produce the mentioned clusters. The dendrogram employing the Ward's distance can be seen in the following Figure.

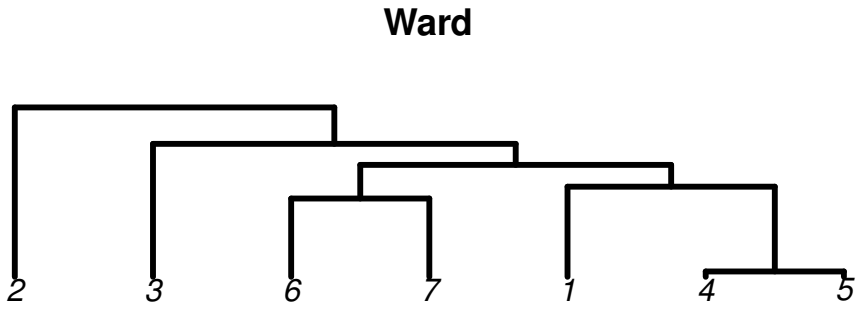

Figure 2.5 Clustering the simulated objects in Figure 2.3 using hierarchical clustering with Ward's distance.

In Figure 2.5, cutting the dendrogram after two merges, produces the desired clusters. It should be noted that this cutting point should be decided visually. This toy example shows that the clustering algorithm that employs the marginal probability is a better alternative.

The results of the modeling and the clustering are summarized in Figure 2.6 using Fourier bases. In Figure 2.6, the seven objects are correctly assigned to five dissimilar groups. The same result is produced using other bases. In terms of the modeling, however, Fourier and circular harmonics produce better fits visually; they avoid over smoothing of the edges and they are less sensitive to sudden variations of data.

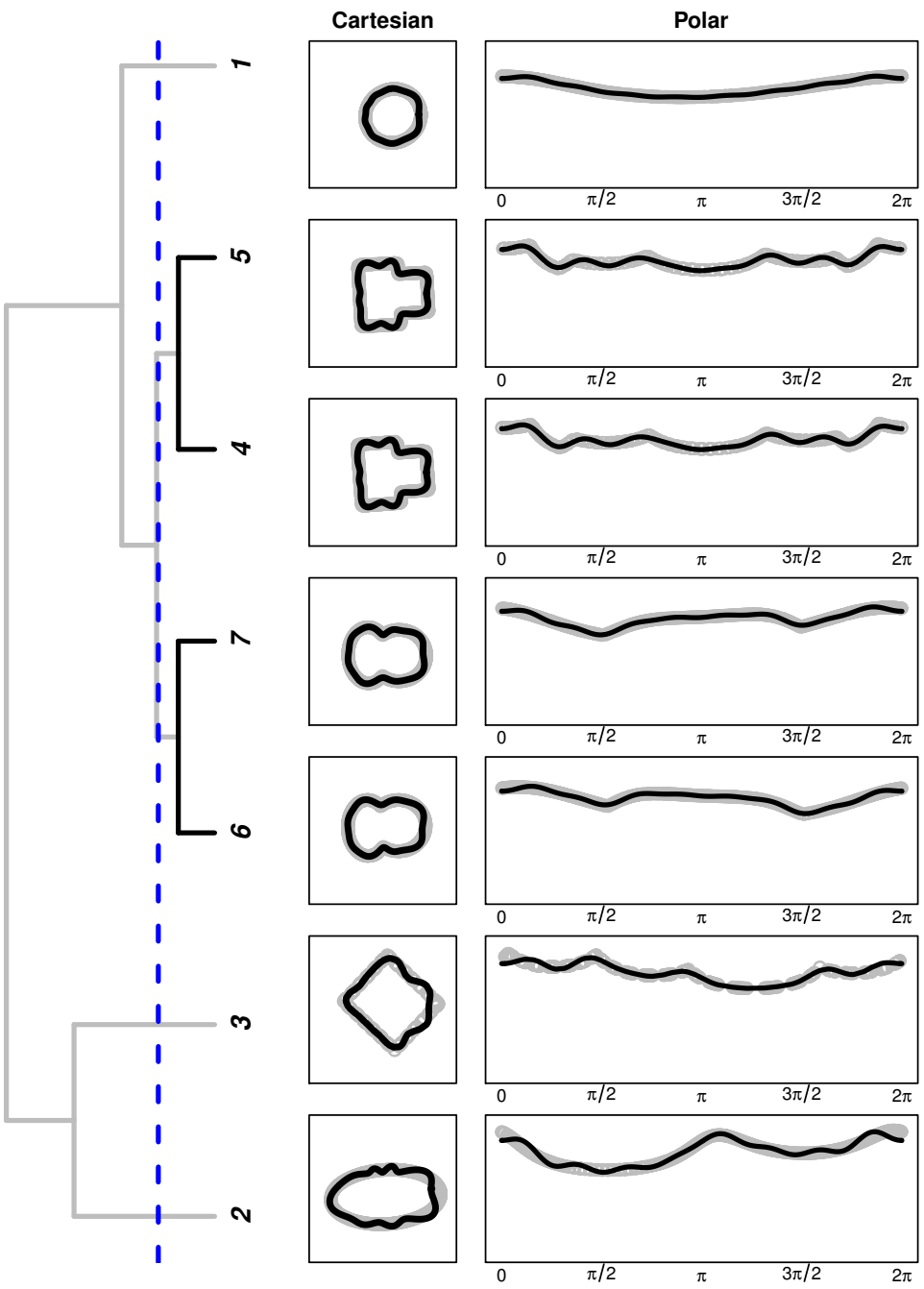

Figure 2.6 Left panel, dendrogram of posterior probability associated with each simulated closed curve using circular harmonic basis functions with 17 terms in equation (1.2). Black lines represent the improvement in the posterior probability and gray lines depict the deterioration in the posterior probability. The dashed blue line indicates the maximum a posteriori cutting point for the dendrogram. Middle panel, the fitted curves to each of simulated curves used in the dendrogram in 2D space. The gray points represent the boundary data used in modeling. Right panel, the same curves as the middle panel are graphed in 1D space.

As an alternative to building the dendrogram, the Markov chain Monte Carlo method can be used over all possible states of $\mathbf{d}$. Here, we opt for random scan Gibbs sampling as an example. The random scan Gibbs sampling over $\mathbf{d}$ values after a minimum of 11 cycles results in the same number of groupings produced by the dendrogram. Constructing the dendrogram for the seven images goes through 6 complete cycles such that in each cycle it performs $\binom{\mathcal{C}(\mathbf{d})}{2}$ comparisons, i.e. 56 comparisons in total. The Gibbs sampling in each cycle performs $\mathcal{C}(\mathbf{d})$ comparisons. Depending on the $\mathcal{C}(\mathbf{d})$ at each cycle, the number of comparisons vary.

As the data are in form of pixels extracted from the images, outliers can occur typically. To avoid any inaccurate results, one can use heavier-tailed models comparing Gaussian models after analyzing the error terms for each model.

2.8.2 3D Shapes

Here, we simulated some data from two random closed 3D object. Five samples with different error terms are obtained from either of the objects to be served for modeling and clustering procedure using our proposed method. The results are summarized in the following graph. The samples are directly being modeled in 3D space, unlike the 2D shapes, due to characteristics of spherical harmonic bases.

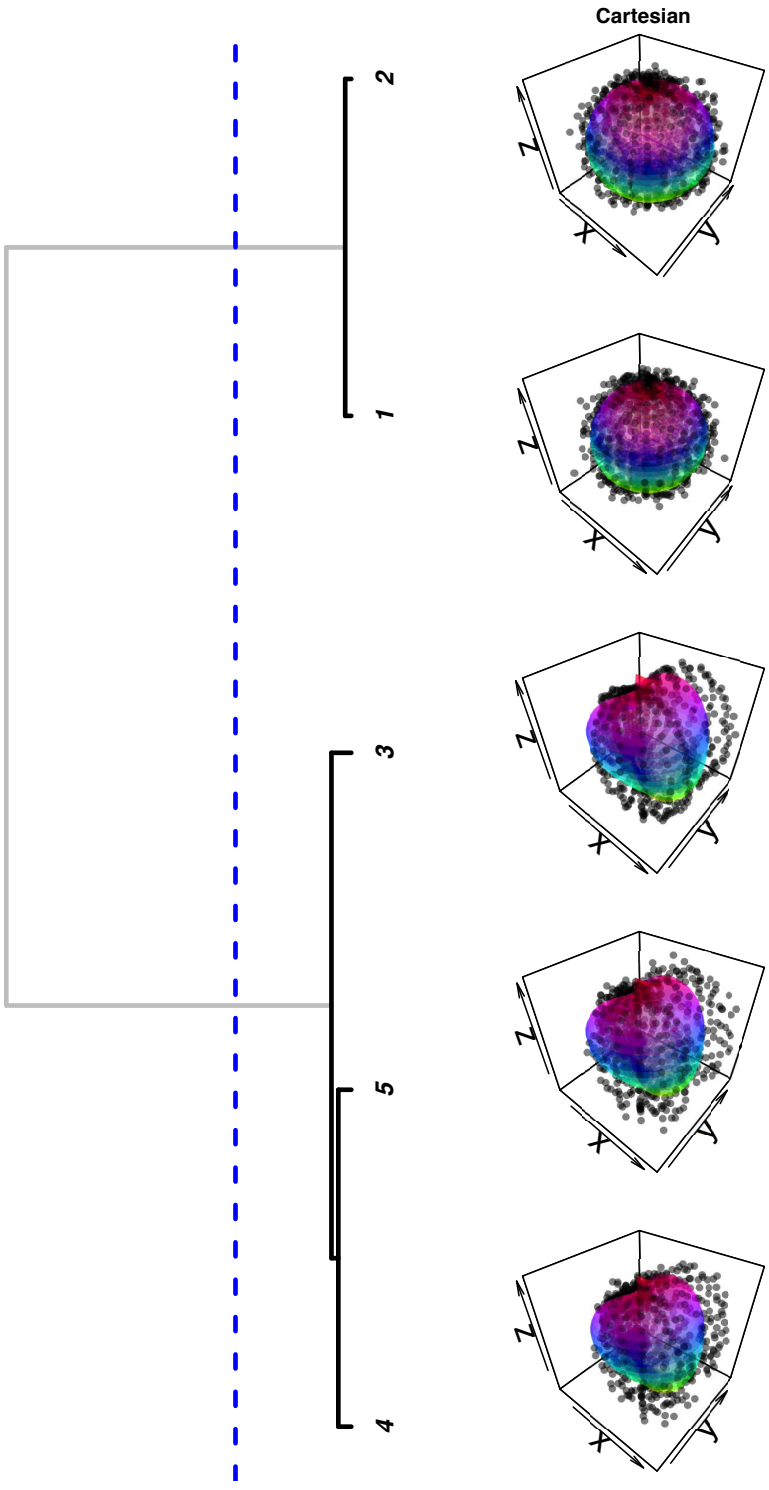

Figure 2.7 Left panel, dendrogram of posterior probability associated with each simulated 3D objects using spherical harmonics with $L_{\max} = 2$. Black lines represent the improvement in the posterior probability and gray lines depict the deterioration in the posterior probability. The dashed blue line indicates the maximum a posteriori cutting point for the dendrogram. Right panel, the 3D data and the corresponding fit in the Cartesian coordinates.

In the Figure 2.7, the objects are correctly allocated to different clusters using our proposed method.

2.9 Application

In this section, the application of our method is tested on the biological cell data obtained from Murphy lab [1]. The database contains 3D images from HeLa cell line captured by laser-scanning microscope. For this study, the images which are labeled as monoclonal antibody against an outer membrane protein of mitochondria are utilized. There are fifty data folders each representing the data from a distinct cell. Each folder contains four sub-folders and the data corresponding to the cell and crop image folders are used for this study. The cell folder has various images of a specific cell, taken at different depths called confocal plane. For instance, for a cell of 1.6 micrometer thickness, one can have 16 high-resolution images of 0.1 micrometer thickness from bottom to top (z-axis sections). These images can then be stacked one on top of the other to create a single 2D image or to reconstruct a 3D image of the sample, see Figure 2.17.

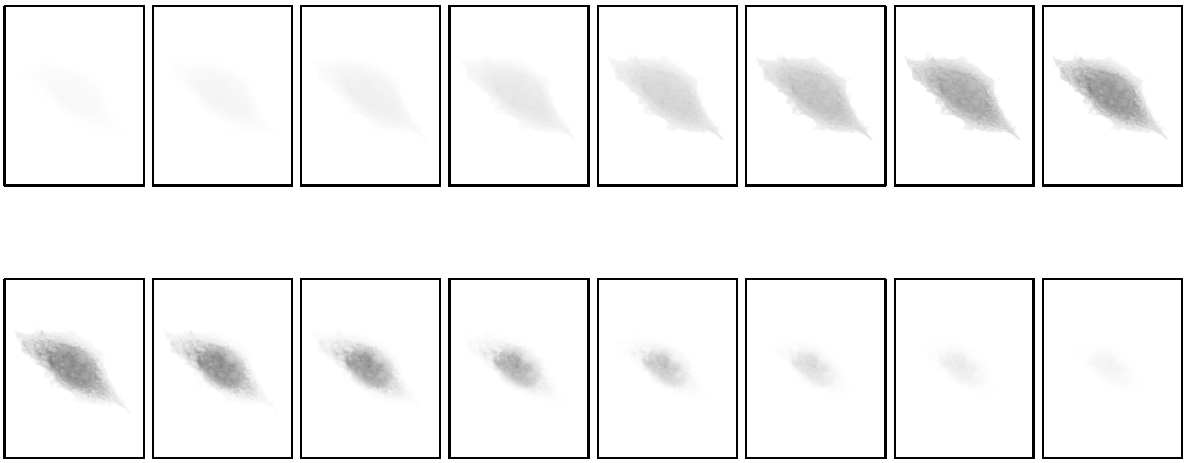

Figure 2.8 An example of confocal laser scanning microscopy for a single cell in the dataset.

More information on various cell imaging techniques can be found in Dailey *et al.* (2006). First, a specific stack common over all cells is chosen. Afterwards, all different stacks for each cell are considered and the cell is treated as a 3D object.

[1] http://murphylab.web.cmu.edu/data/#3DHeLa

2.9.1 2D Shapes

In order to represent the usefulness of our method in clustering of 2D shapes, a single stack, common over all cells, is selected. The appointed stack, thereafter, is segmented using the designed crop image such that there is only one cell per image, see Figure 2.9. To obtain a true representation of a shape from each image, location, scale, and rotational effects should be filtered out. For this purpose, we aligned cells such that their centroids are located in the center of images. In addition, the cells are rotated in direction of their main principal component axes to assure rotation free analysis. Consequently, the coordinate of pixels on the boundary of the cell is extracted for modeling purpose. We use MATLAB standard methods including segmentation, boundary detection and Savitzky–Golay smoothing filter functions from MATLAB toolboxes (MATLAB, 2013a,b) for detecting the boundary of the cell. We call the associated line the *oracular boundary* since it, supposedly, represents the true boundary for each cell. Besides, we propose another method for boundary detection which enables us to take into account the associated uncertainty, see Figure 2.10.

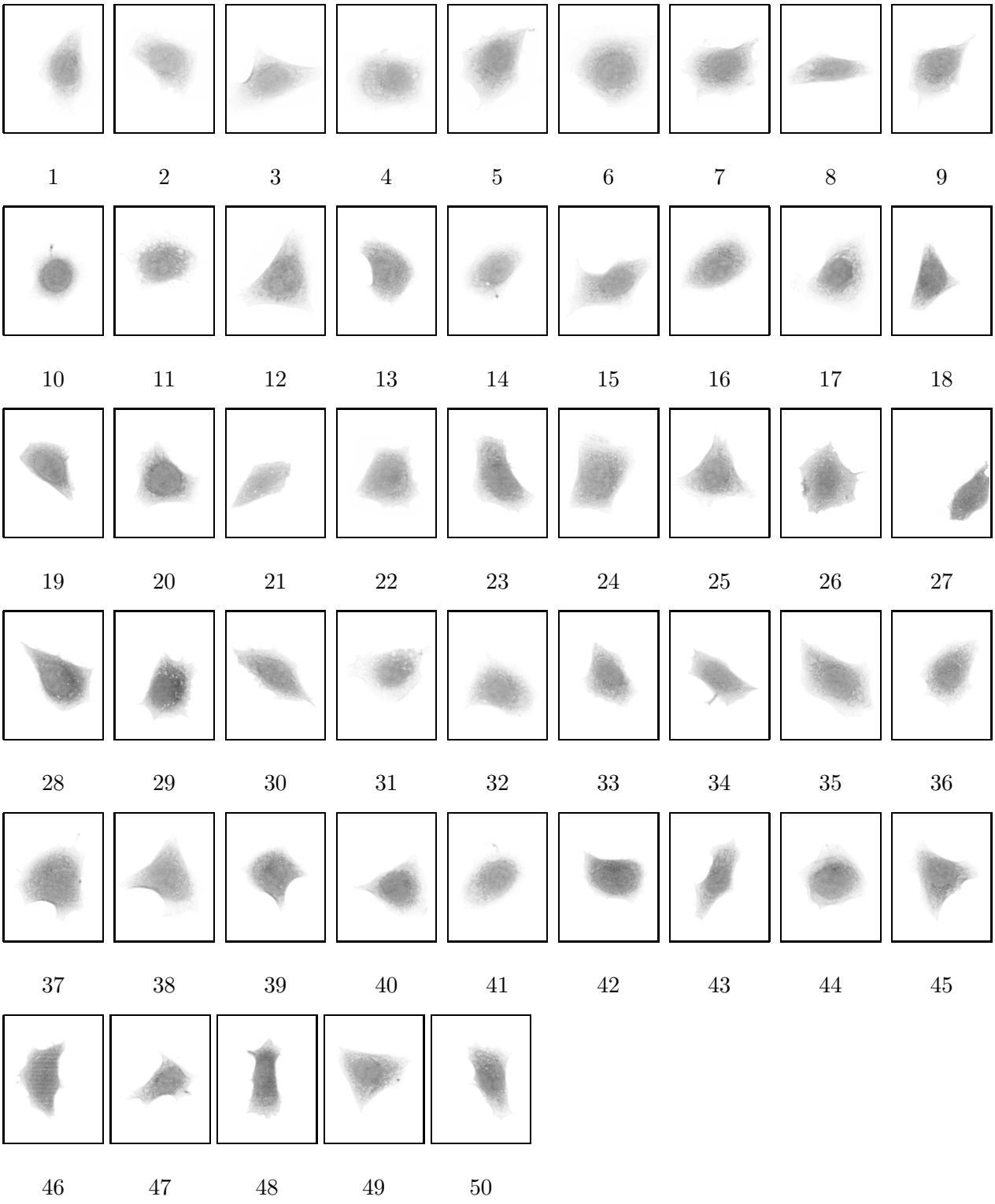

Figure 2.9 The raw images of fifty cells used for clustering throughout this thesis. The number assigned to each cell matches with its order in dataset.

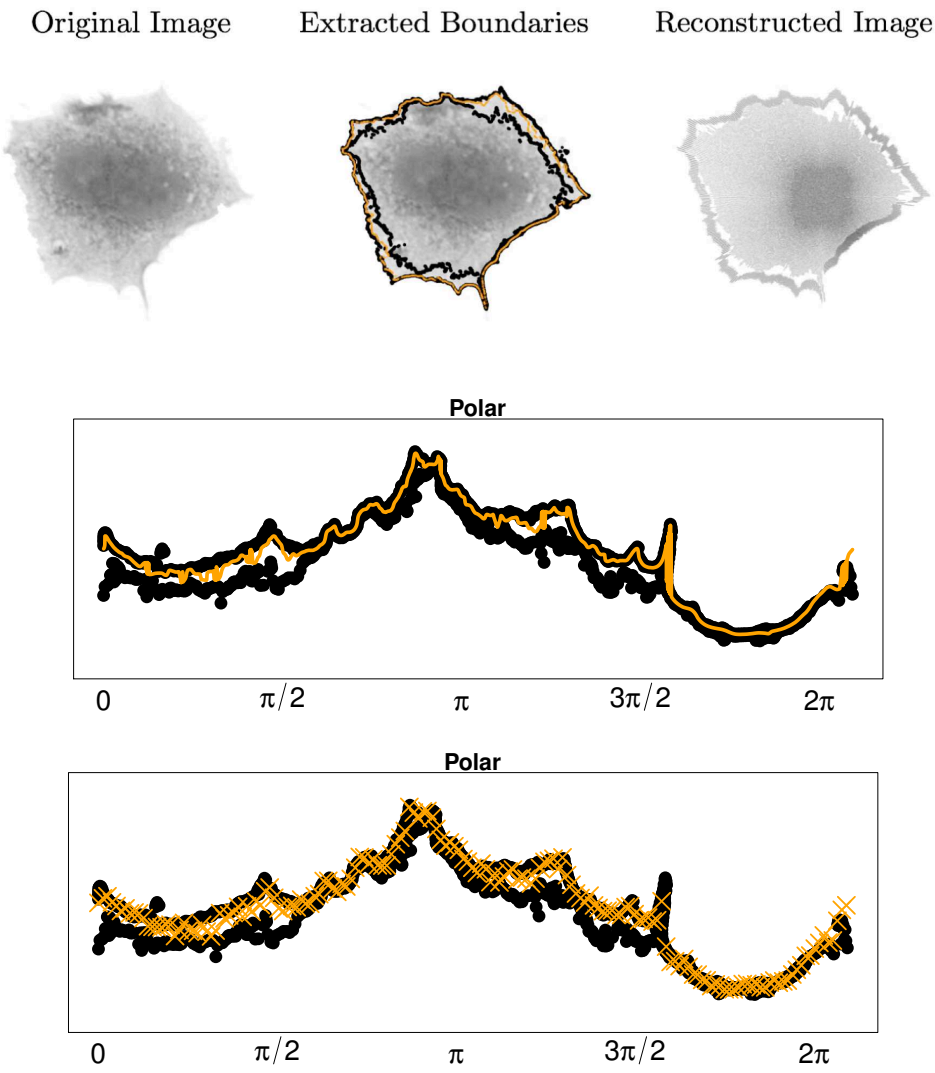

Figure 2.10 Top panel, left: the original cell image. Middle: orange line indicates the oracular boundary using MATLAB standard method. The black points represents the upper and lower bounds for the oracular boundary. Right: the reconstructed image using the boundary data on the left panel and the Gaussian samples. Middle panel, the boundaries extracted from the middle image, are transferred to the polar coordinates. The black points represent the lower and upper boundaries data and the orange line represents the oracular boundary. Bottom panel, samples generated from a Gaussian distribution, centered around the oracular boundary (the orange line in the middle panel) with the common variance over all cells, are plotted with the orange crosses.

For each observation on the oracular boundary, the uncertainty (variance) is calculated

based on the data points on the lower and the upper boundaries in polar coordinates. Lower and upper boundaries are treated as 95% confidence interval and

$$\hat{\sigma}_i \approx \frac{\mathrm{UCL}_i - \mathrm{LCL}_i}{4}, \text{ for } i = 1, 2, \ldots, N,$$

gives an estimation of standard deviation for point i, where UCL_i and LCL_i represent the lower boundary and upper boundary values at the point i respectively. Then the median of the $\hat{\sigma}_i$ is treated as the common standard deviation to be used for all the computations throughout our clustering algorithm. A Gaussian sample centered around each observation on oracular boundary with the so called common variance is generated, Figure 2.10 left and bottom panels.

In order to check the effect of the hyper-parameters on the final grouping, the clustering is carried out over a grid of various values of the hyper-parameters. Consider a re-parameterization of the inverse gamma distribution from $\mathrm{IG}(a, b)$ to $\mathrm{IG}(\mu, \tau)$, in terms of its mean and variance, where

$$\mu = \frac{b}{a-1} \text{ for } a > 1 \text{ , and } \tau = \frac{b^2}{(a-1)^2(a-2)} \text{ for } a > 2.$$

The clustering results obtained is summarized in the following table.

Table 2.2 The effect of prior distribution $\mathrm{IG}(\mu, \tau)$ on the number of groupings using Fourier basis functions with $K = 33$ as an example.

$\mathcal{C}(\mathbf{d})$	τ			
μ	0.25	0.5	1	2
0.5	40	42	43	45
1	15	21	28	38
2	3	5	9	17
4	1	1	2	4

The starting values of $\mu_0 = 0.5$ and $\tau_0 = 0.25$ in Table 2.2 are empirical moment estimation using the data. These values allocate each cell to a separate cluster. Considering an overdispersion parameter α as $\mu = \alpha\mu_0$, which allows for contemplating any prior information about the number of clusters. For instance, setting $\alpha = 2$ leads to 15 clusters in total. Exceedingly shifting the prior distribution of σ^2 while keeping the variance, τ, low, overwhelms the likelihood function and merges all shapes into a single cluster. Therefore,

the hyper-parameter should be tuned with caution.

For the sake of readability, in Figure 2.11, the dendrogram is reported for only ten random cells. Each cell is assigned a number corresponding to its order in the database. To have a fair comparison, the number of terms in linear expansion, K, is kept fixed across the basis functions. Setting $\alpha = 1$, all the basis functions yield the same 7 clusters, see Figure 2.11.

The same procedure is applied for all fifty cells, see Figure 2.12 and Figure 2.13. Setting the overdispersion parameter $\alpha = 4$, such that $\mu = 2$ and $\tau = 0.5$ in Table 2.2, leads to significantly lower number of clusters, see Figure 2.14 and Figure 2.15.

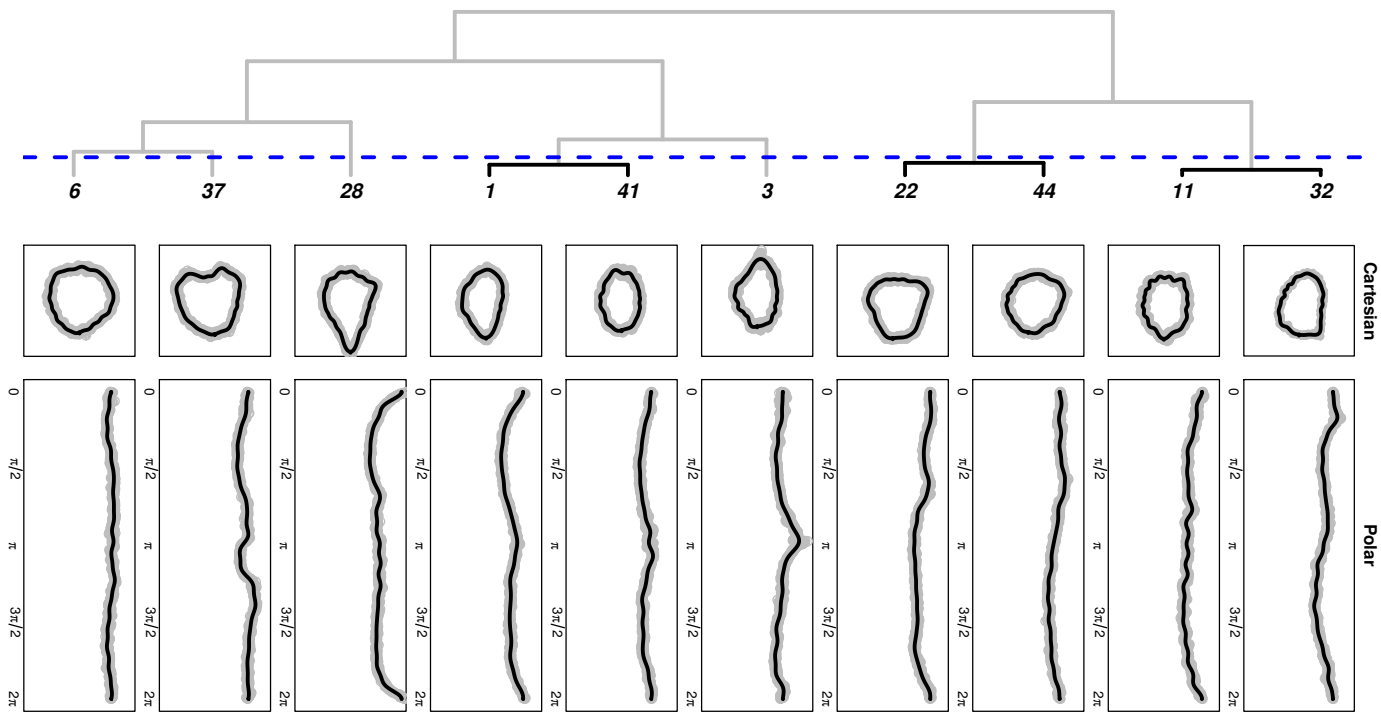

Figure 2.11 Top panel, dendrogram of posterior probability associated with each cell using Fourier basis functions with $K = 33$ terms in equation (1.2). Black lines represent the improvement in the posterior probability and gray lines depict the deterioration in the posterior probability. The dashed blue line indicates the maximum a posteriori cutting point for the dendrogram. Middle panel, the fitted curves to each of ten random selected cells used in dendrogram in 2D space. The gray points represent the boundary data used in modeling. Bottom panel, the same curves as the middle panel are depicted a in 1D space.

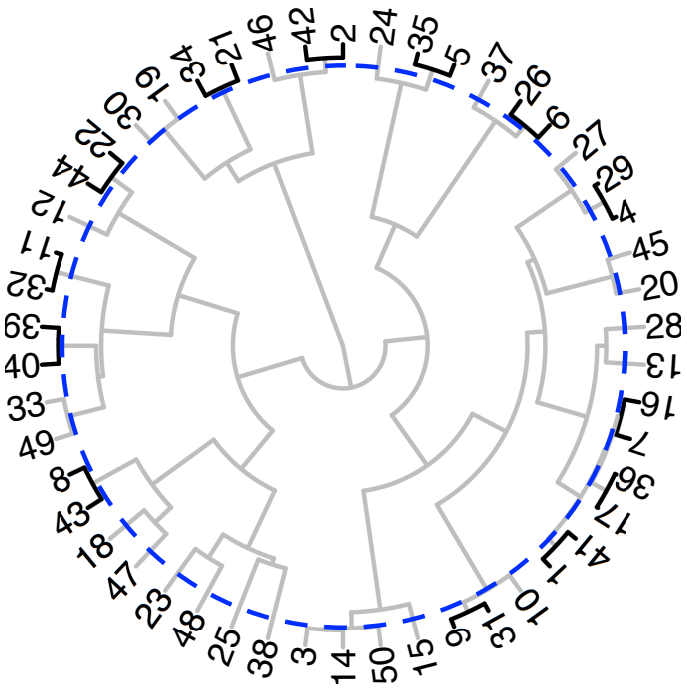

Figure 2.12 The dendrogram of posterior probabilities associated with 50 cells using smoothing spline basis functions with $K = 33$ terms in equation (1.2). Black lines represent the improvement in the posterior probability and gray lines depict the deterioration in the posterior probability. The dashed blue circle indicates the maximum a posteriori cutting point for the dendrogram.

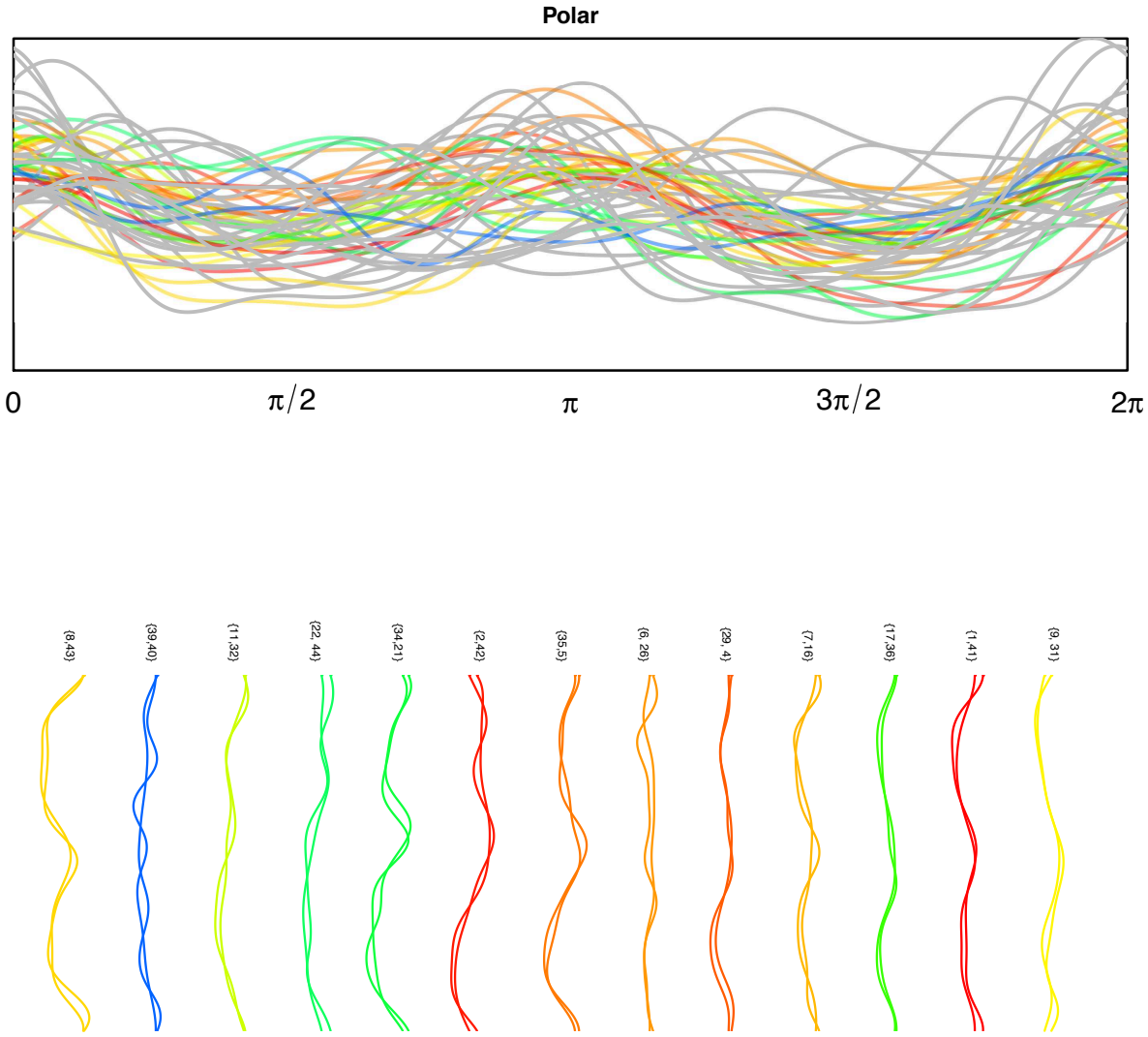

Figure 2.13 Top panel, the set of curves fitted to the 50 cells in Figure 2.12 using smoothing splines with $K = 33$ terms in equation (1.2). The curves in singleton clusters are plotted in gray and the mutli-member clusters each are illustrated with distinct colors. Bottom panel shows curves involved in each multi-member cluster defined by the dendrogram in Figure 2.12.

Different basis functions produce different groupings across the cells. Fourier and circular harmonics basis functions produce the same 40 clusters on the cells, while smoothing splines and wavelets lead to 37 and 38 clusters respectively. As an example, the circular harmonic

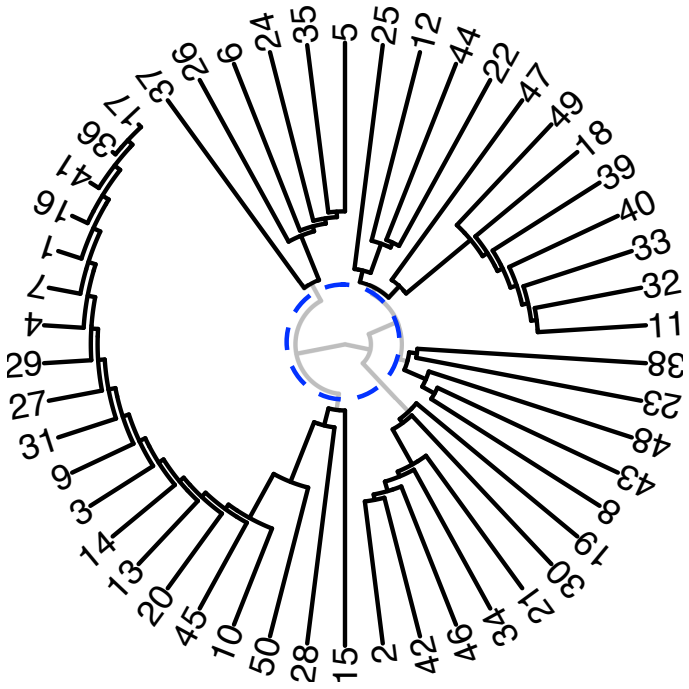

Figure 2.14 The dendrogram of posterior probabilities associated to the same cells as in Figure 2.12 when $\alpha = 4$.

basis functions render the following two-member clusters.

$$\{1, 41\}, \{5, 35\}, \{6, 26\}, \{4, 29\}, \{7, 16\}, \{17, 36\}, \{11, 32\}, \{2, 42\}, \{9, 31\}, \{22, 44\}.$$

The over smoothing of the edges in smoothing spline causes cells with global similar features to be grouped together. The smoothing splines, in other words, ignore some details in boundaries of the shapes. On the other hand, wavelets are greatly sensitive to small variations in data. One may choose among the bases by considering a reasonable trade-off based on the clustering objective.

Similarly, one may run an MCMC over the space of $\mathbf{d}$ to determine the grouping for which the posterior probability reaches its maximum value. The random scan Gibbs sampling after 60 cycles produces the same groupings for the Fourier and circular harmonics, but different

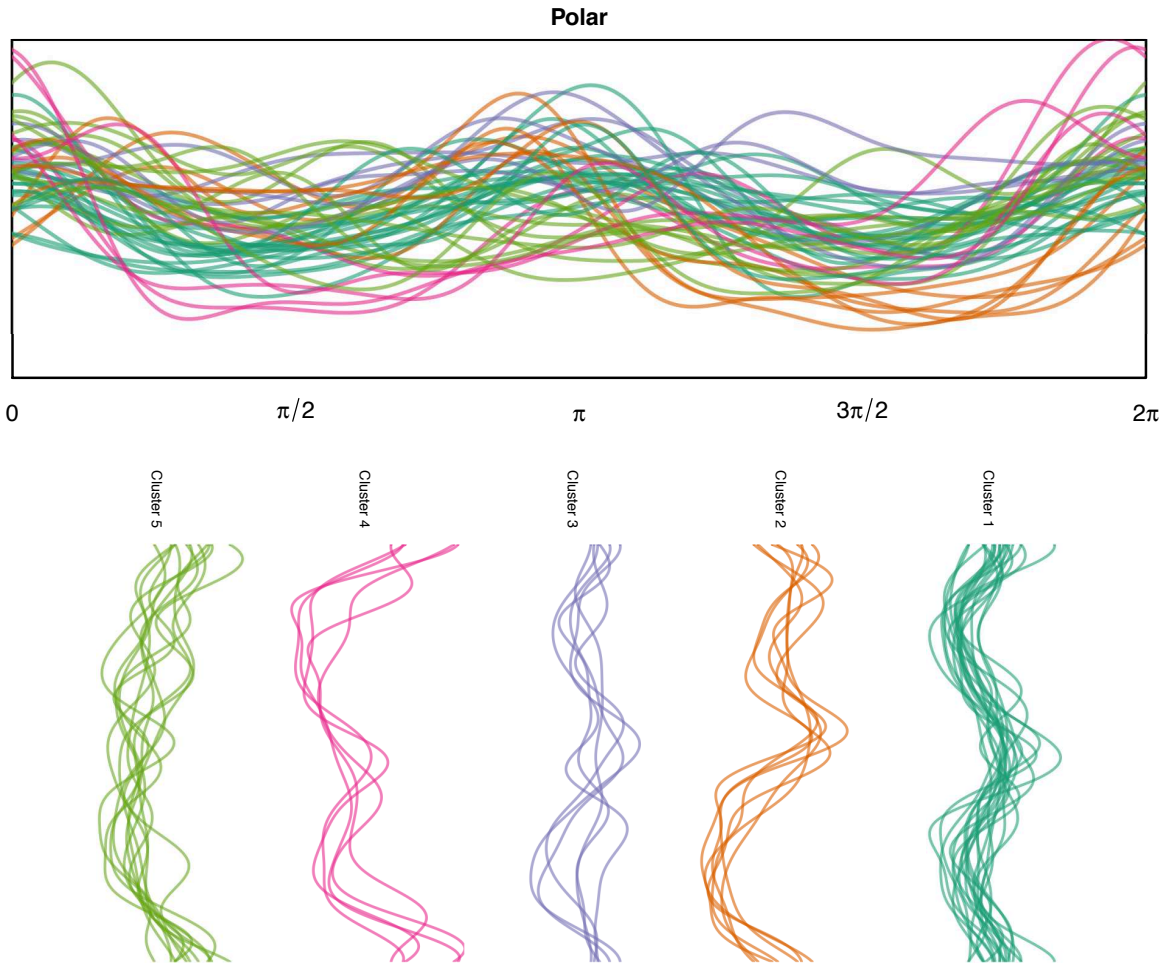

Figure 2.15 Top panel, the set of curves fitted to the 50 cells in Figure 2.14 using smoothing splines with $K = 33$ terms in equation (1.2). The mutli-member clusters each are illustrated with distinct colors. Bottom panel shows curves involved in each multi-member cluster defined by the dendrogram in Figure 2.14.

groupings for the wavelets and smoothing splines. The results of clustering methods and the differences between the methods using Rand index are reported in Figure 2.16.

The Rand index is a popular measure for cluster validation where the 0 reflects complete similarity and 1 reflects no similarity between two methods. The Rand index values, overall, are small in Figure 2.16, which means that the difference between these basis functions is negligible in this example.

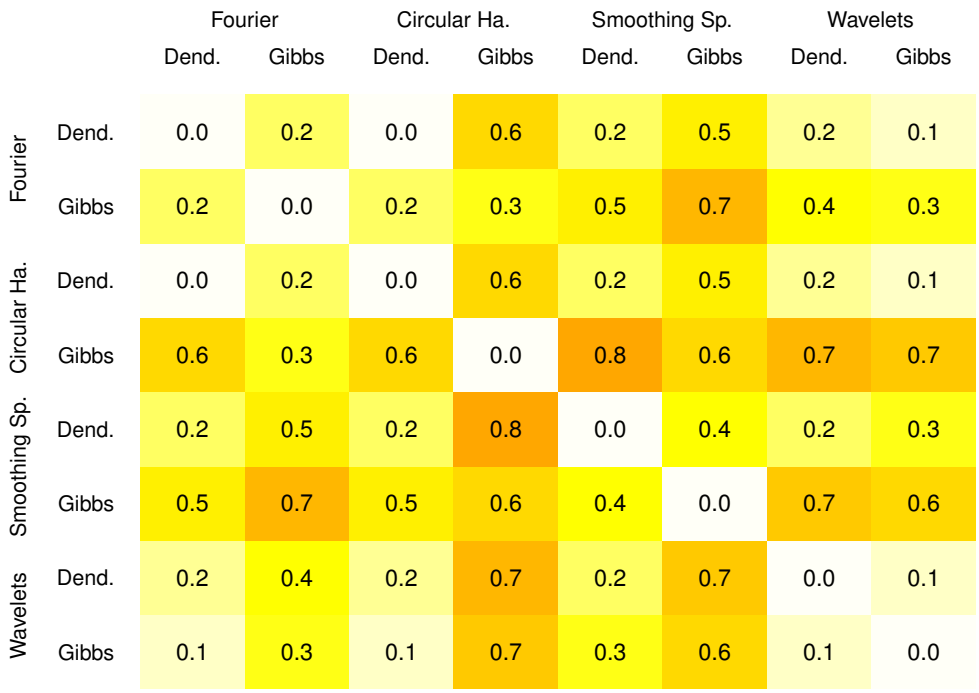

Figure 2.16 The Rand index values $\times 100$, as distance measures, between clusters obtained from dendrogram and Gibbs sampling methods using different basis functions. Top panel, the Gibbs sampling is run for a total of 25 cycles. Bottom panel, the Gibbs sampling is run for total of 50 cycles.

2.9.2 3D Shapes

The shape studies according to only 2D cross sections of cells may produce misleading results, as it lacks the full information about the 3D shape. In this section, we take into account the information from all stacks. In order to obtain the 3D Cartesian coordinates associated to each voxel on the surface, we do as follows. First, the boundary data for each image stack is extracted separately, following the same procedure. Second, the 2D coordinates of each stack are combined all together to create the 3D coordinates of the cell shapes. Afterwards, we take into account the image spacing information. For this dataset, the voxel spacing is $(0.049\mu m, 0.049\mu m, 0.203\mu m)$ (Peng and Murphy, 2011). It should be noted that the final extracted data must locate within a sphere of unit radius to be suitable for modeling using spherical harmonics.

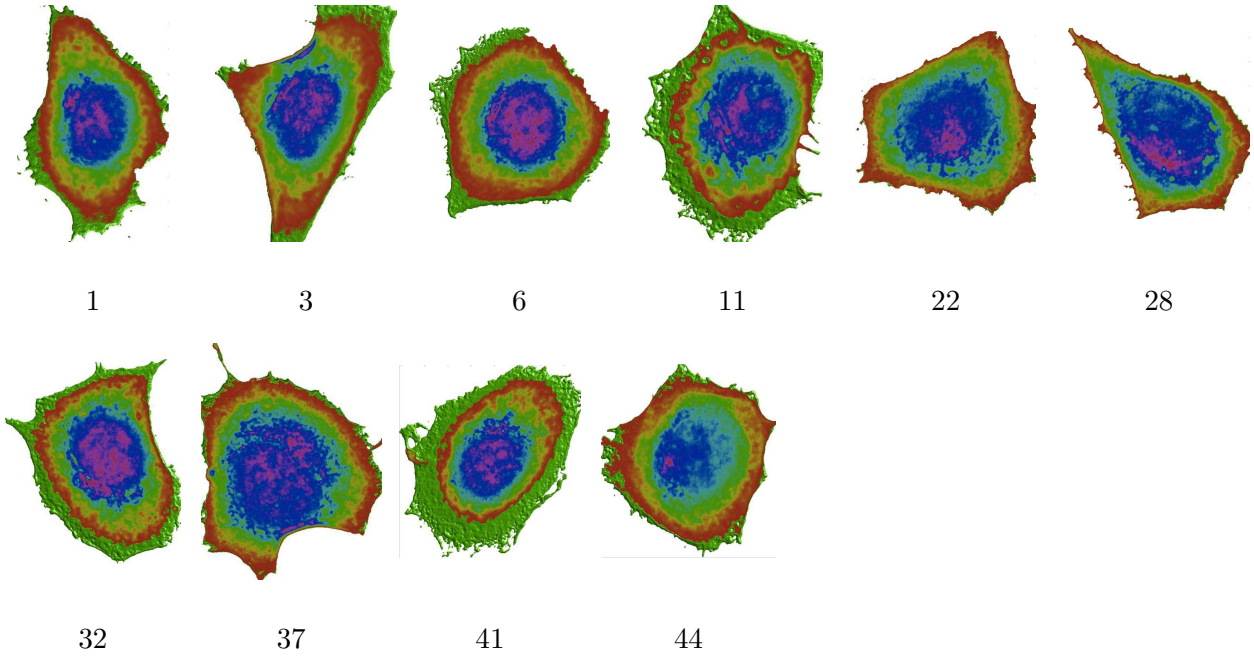

Figure 2.17 Reconstructing the selected cell shapes by embedding the stack of 2D images. The number assigned to each cell matches with its order in dataset.

In Figure 2.18, one can see how spherical harmonic bases accomplish in modeling cell shapes.

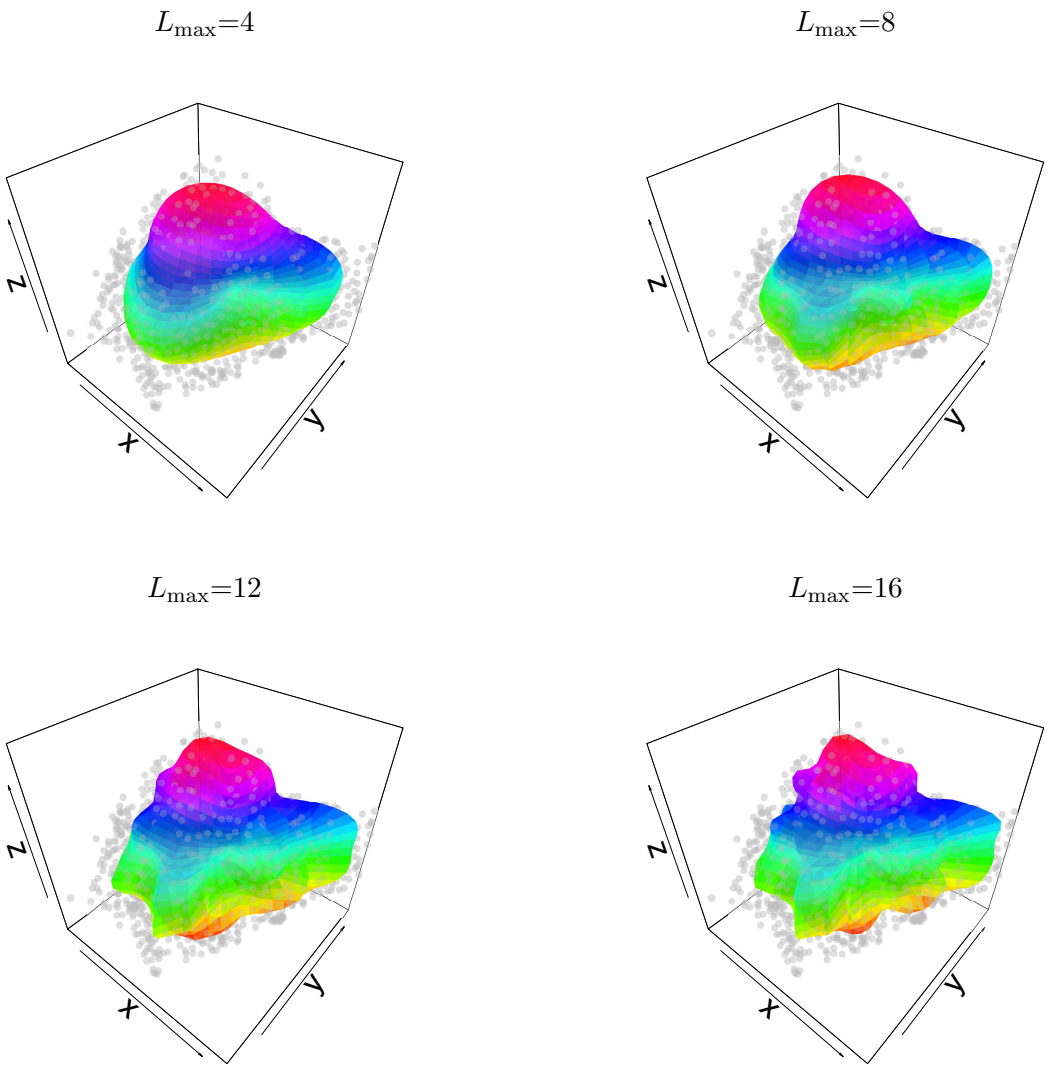

Figure 2.18 3D modeling of a cell shape using spherical harmonic basis functions with different values of $L_{\max}$. As $L_{\max}$ increases, the model become more flexible and complex.

Similar to the 2D case, the clustering procedure is applied to the 3D data extracted from cells. The result of clustering for the same 10 random cells as in Section 2.9.1 are reported in Figure 2.19.

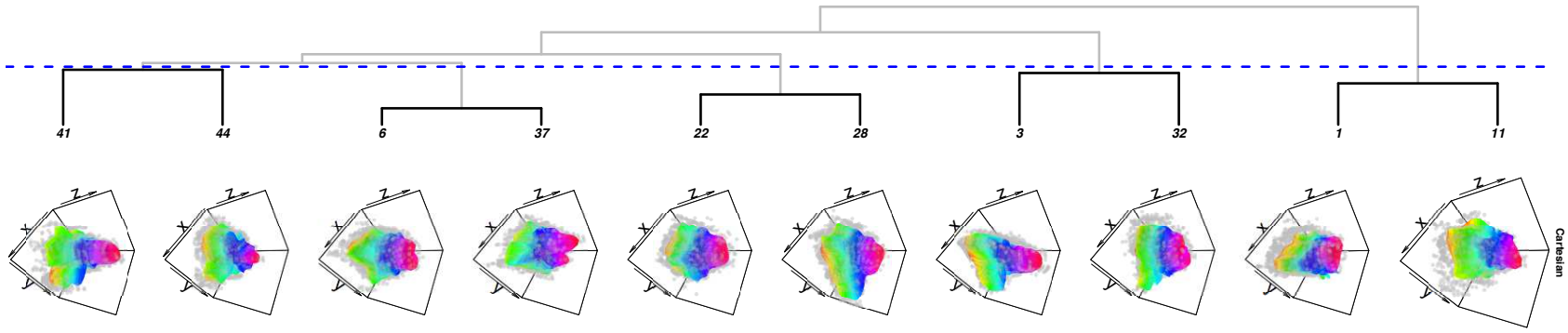

Figure 2.19 Top panel, dendrogram of posterior probability associated with each cell using spherical harmonics with $L_{\max} = 12$. Black lines represent the improvement in the posterior probability and gray lines depict the deterioration in the posterior probability. The dashed blue line indicates the maximum a posteriori cutting point for the dendrogram. Bottom panel, the 3D data and the corresponding fit in the Cartesian coordinates.

In Figure 2.19, the clustering results differ comparing to its 2D counterpart in Figure 2.11. As in the 3D case, we allow for more information, the clustering results are assumed to reflect the true grouping better. We repeat the same procedure considering all 50 cells. The clustering results are reported in Figure 2.20. As we discussed in Section 2.3, the number of all possible groupings is

$$\sum_{k=1}^{50} \left\{ {50 \atop k} \right\} \approx 10^{47}.$$

In practice, it is not feasible to explore all possible groupings when D is relatively large. We run the random Gibbs sampling for 8000 cycles as an example. The convergence behavior of the sampling throughout the 8000 cycles is reported in Figure 2.21. In Figure 2.21, middle panel, the chain seems to stabilize at 5 clusters after 700 cycles. However, the grouping suggested by sampling at each cycle varies, see Figure 2.21. The grouping generated from random Gibbs sampling after the 8000 cycles is as follows,

$$\text{Cluster } 1 = \{4, 12, 33, 38\}, \text{Cluster } 2 = \{1, 13, 26, 48\}, \text{Cluster } 3 = \{5, 8, 11, 21, 22, 32\},$$

$$\text{Cluster } 4 = \{2, 3, 14, 18, 20, 24, 27, 29, 30, 34, 35, 40, 43, 44, 45, 46, 47\},$$

$$\text{Cluster } 5 = \{6, 7, 9, 10, 15, 17, 19, 23, 25, 28, 31, 36, 37, 39, 41, 42, 49, 50\}.$$

In order to have a visual comparison, we show in Figure 2.22 the members of each cluster using their 2D image as in Figure 2.9.

In Section 2.9.1, we explained that the data are generated by some Gaussian error around the oracular boundary. Now, we verify the Gaussian assumption for the residuals of fits considering the grouping imposed by the Gibbs sampling, Figure 2.22. In Figure 2.23, the Quantile-Quantile (Q-Q) plot and the histogram for the residuals of fits after clustering, are reported. Figure 2.23 indicates that the Gaussian assumption is almost valid.

As a dendrogram does not explore the whole space of possible groupings, the results of clustering for Gibbs sampling and that of the dendrogram in Figure 2.20 differ.

In this chapter, we proposed a new methodology for clustering cells shapes employing their likelihood functions. We demonstrated that the suitable grouping of cell shapes happens when the posterior probability of grouping, $p(\mathbf{d} \mid \mathbf{y})$, attains its maximum. Two methods have been suggested for searching the possible value of maximum a posteriori of grouping: 1) dendrogram, 2) MCMC methods. In dendrograms, one can only explore a subspace of all possible groupings in order to approximate the maximum value of $p(\mathbf{d} \mid \mathbf{y})$. On the other hand, MCMC methods enables us to traverse the whole space to obtain the maximum value. However, as the number of shapes increase, the search over whole space become cumbersome.

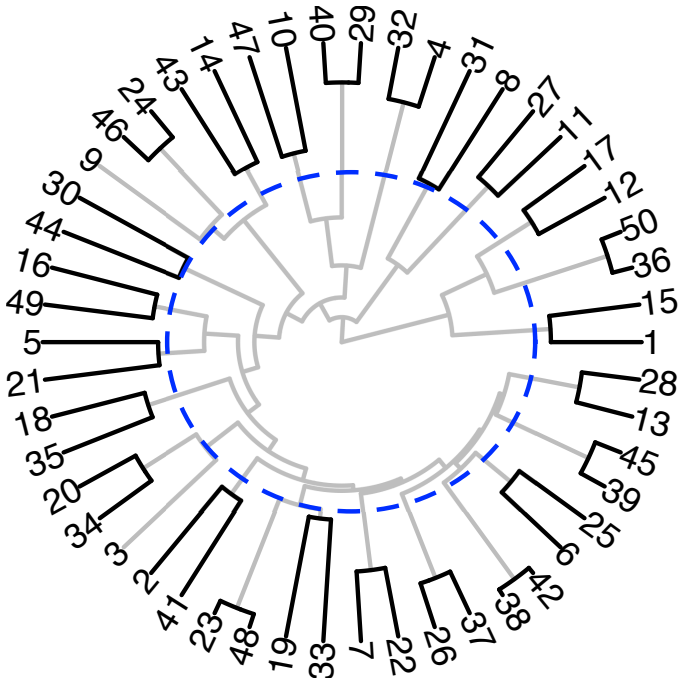

Figure 2.20 The dendrogram of posterior probability associated with 50 cells using spherical harmonics with $L_{\max} = 12$. Black lines represent the improvement in the posterior probability and gray lines depict the deterioration in the posterior probability. The dashed blue circle indicates the maximum a posteriori cutting point for the dendrogram.

Generally, cells are dissimilar in terms of their physical structure, even if they belong to the same tissue of a body. Our methodology helps in clustering the cells taking into account their shapes. The huge gap between different clusters signals the existence of a hidden factor that needs to be studied further. The alteration in the intra-cellular activities due to cancer is one of the possible factors.

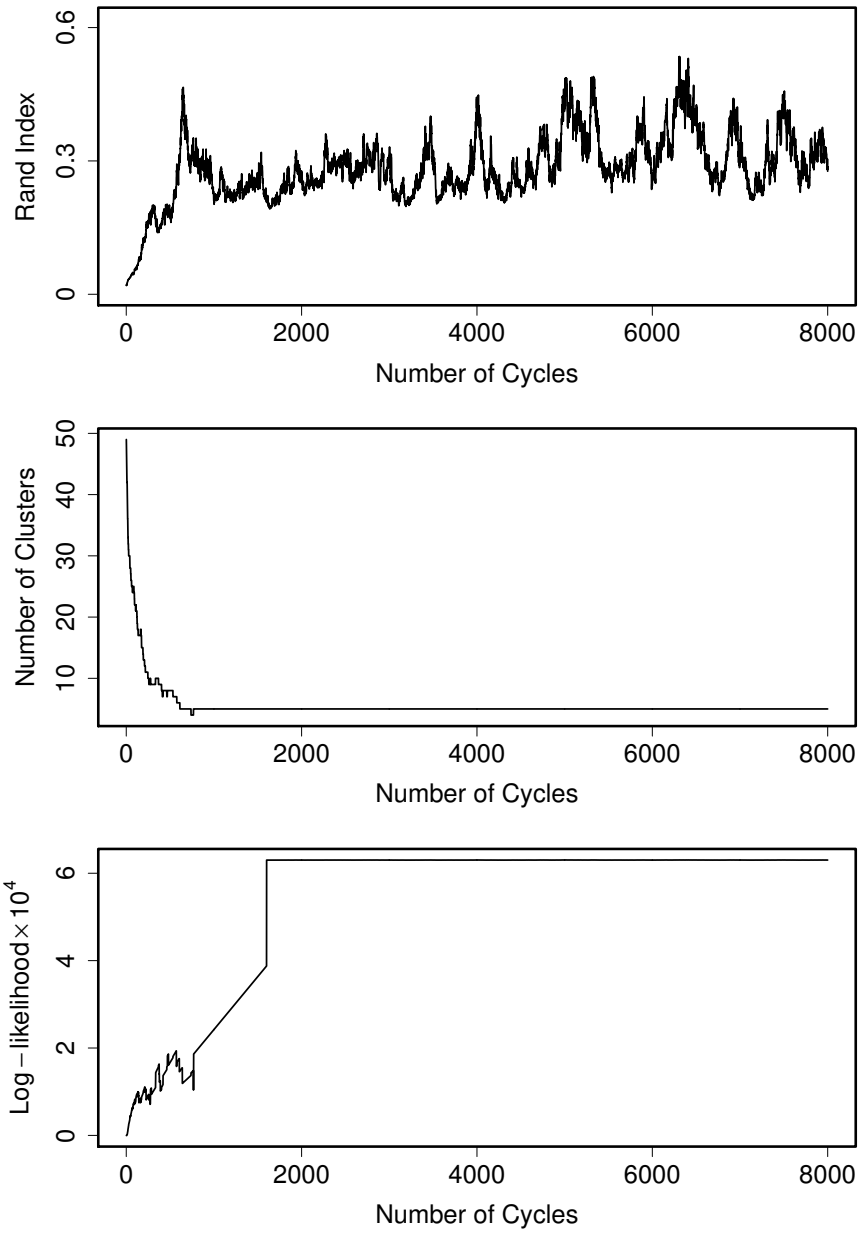

Figure 2.21 The result of random Gibbs sampling for the same cells as in Figure 2.20. Top panel: the Rand index between the grouping suggested at each cycle of random Gibbs sampling with the grouping produced by the dendrogram in Figure 2.20. Middle panel: the number of clusters produced through 8000 cycles of random Gibbs sampling. Bottom panel: the logarithm of posterior probability values for 8000 cycles of random Gibbs sampling.

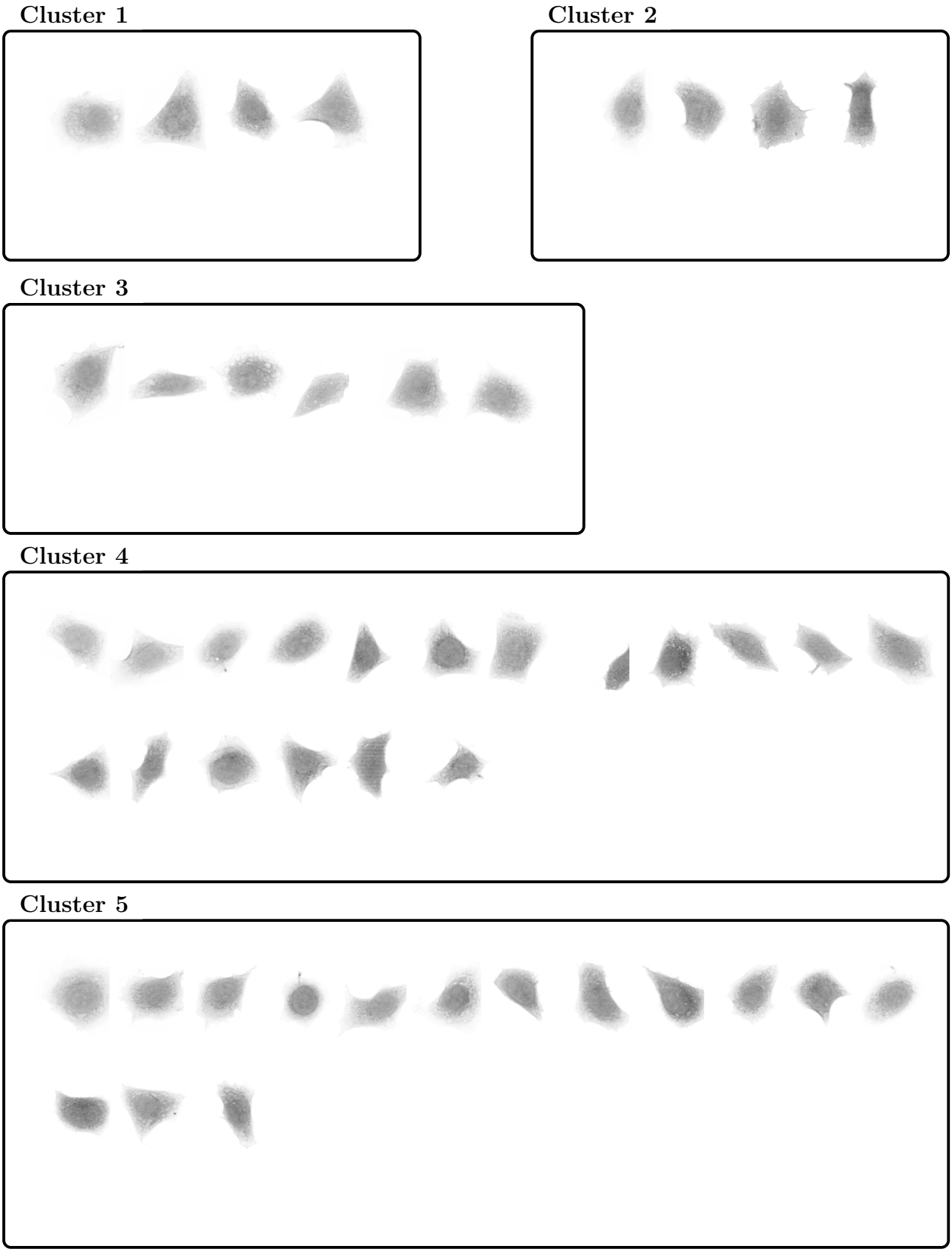

Figure 2.22 Clusters produced by Gibbs sampling after 8000 cycles. Each cluster member is presented by the 2D image of the corresponding cell.

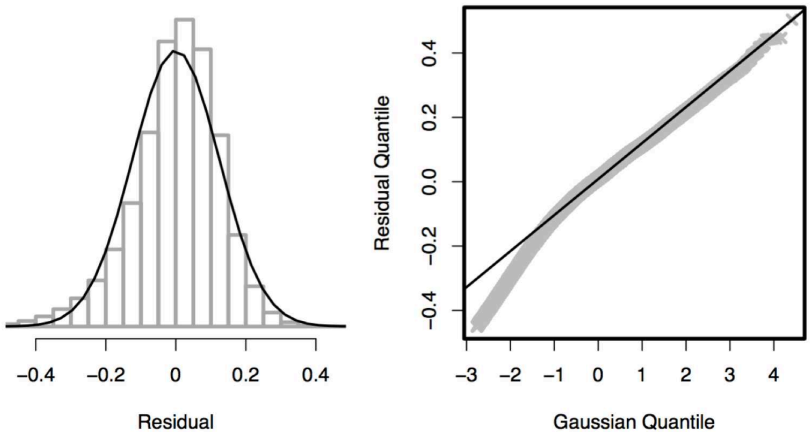

Figure 2.23 Left panel, the histogram of the residuals of fits after the grouping imposed by the Gibbs sampling, Figure 2.22. Right panel, the Q-Q plot for the same residuals as the left panel.

CHAPTER 3 ELECTRONIC NOSE: DATA VALIDATION AND ODOR CONCENTRATION PREDICTION

The term "odor" specifies the action when one or more chemicals approach the receptors in the olfactory nerve and stimulate them. Odor modulates various aspects of human life such as sexual attraction, mood, dietary preferences, and fear. The human sense of smell does not, however, respond to all harmful air pollutants. Moreover, sensitivity of humans to many air pollutants varies — one can be accustomed to a toxic smell. In the last decade, great attention has been paid to the subject of air quality, because the air directly influences the environmental and human health. A crucial element in the assessment of indoor and outdoor air quality is auditing odorants.

There are various odor measurement techniques such as dilution-to-threshold, olfactometers, and referencing techniques (McGinley and Inc, 2002). The performance of these approaches depends on human evaluation. Due to the high variability of an individual's sensitivity, such methods mostly lack accuracy. In 1982, the first gas multi-sensor array was invented as primary artificial olfaction (Persaud and Dodd, 1982). The term electronic nose (e-nose) was introduced in 1994 (Gardner and Bartlett, 1994). E-nose is an artificial olfactory system which consists of an array of gas sensors. The e-nose is designed to classify odors of its surrounding environment (Boeker, 2014). The gas sensor array receives chemical information about gaseous mixtures as the input, and converts them into measurable signals.

In this chapter, we discuss some challenging problems in the domain of electronic nose technology and a methodology to tackle them. First, a description of the problem is given and then a technique for handling the problem is presented. The applicability of the method is being examined on a set of simulated, and real data.

3.1 Problem Statement

The inherent features of gas sensors cause unnecessary complications to the process of odor recognition. Some of these features are listed below.

- Gas sensor's performance is affected by different elements, which can make the sensor unstable and less sensitive to odors. One of the most serious deterioration in sensors is a phenomenon called *drift*. Drift is a temporal change in sensor's response while all other external conditions are kept constant. The majority of manufactured sensor arrays are subject to drift, and several methods have been introduced to overcome this problem (Carlo and Falasconi, 2012; Artursson *et al.*, 2000; Padilla *et al.*, 2010a; Zuppa *et al.*, 2007).

- Cross-sensitivity of gas sensors is inevitable in sensor array structure. The cross-sensitivity is the interaction among chemicals that leads to a different signal from the component in a mixture compared to the single component.

- The behaviour of a sensor is directly influenced by the surrounding chemical and physical conditions. For instance, the response of a sensor may depend on the temperature of the gas under examination. Therefore, thermal conditions around the sensing elements must be under control.

The multivariate response of gas sensor arrays undergoes different pre-processing procedures, prior to the implementation of pattern recognition methods. Amine *et al.* (1999); Yan *et al.* (2015); Shao *et al.* (2015); Pardo *et al.* (2000); Wilson *et al.* (2000) have discussed various systematic feature extraction methods for gas sensor data by minimizing the redundancy in the data. They suggest the use of principal component analysis (PCA) in identifying the outliers for transformed measurements from sensors.
Our contributions and their importance in e-nose technology are listed below.

1. Sensors of the e-nose may report incorrect values or some of the sensors may stop functioning for a short period of time. These anomalies need to be diagnosed and reported in real time, using a computationally efficient algorithm. There has been extensive studies on identification of faulty sensors in sensor arrays including Fonollosa *et al.* (2012, 2013); Padilla *et al.* (2010b). Here, we focus on quality of e-nose measurements rather than identifying the individual faulty sensors, which is a more general approach.

   Our first contribution is to assemble various statistical methods to be used as an algorithm for anomaly detection. This algorithm takes the statistical properties of sensors' measurements into account to assure a reliable and a statistically robust anomaly detection tailored for e-nose data.

2. Often, the sensor's output is used to quantify odor concentration. Transferring the sensor's output to olfactometry laboratory is cumbersome. Only small portions of data are considered for further analyses of its concentration in olfactometry. The portion of data which is tagged by their coorresponding odor concentration is called *labaled training data*. The pattern recognition methods employ the labeled training data in order to predict the odor concentration for future sensor values. Numerous methods have been developed for modeling the gas sensor array data, including Gutierrez-Osuna (2002); Hyvarinen (1999); Kermiti and Tomic (2003); Bermak *et al.* (2006); Qin (1997).

   Our second contribution is employing a more flexible supervised learning model in terms of robustness and sparsity for predicting the odor concentration.

In short, the main focus of this chapter is on two subjects. First, the data validation for the sensors' measurements. Second, training a supervised model on data in order to predict the odor concentration for a batch of online measurements.

3.2 Data Description

The type and number of gas sensors on an e-nose depend on the application the e-nose is designed for. The sensors detect the change in electrical resistance when they are in contact with volatile compounds. Sensors react to almost all gases in the air, but each sensor is intended to be more sensitive to a specific type of gas. Better understanding of an e-nose data is necessary for designing an effective data validation algorithm. For this reason, the existence of various common statistical assumptions should be verified.

The data under the study include 11 distinct attributes, each representing one sensor value of the e-nose. As some of the sensors measure nearly the same gases, they happen to be highly positively correlated, see Figure 3.1, and Figure 3.2 (left panel). From now on, p refers to the number of attributes. Suppose that $\mathbf{x}_{p\times 1}^{\top}$ is a random vector of p attributes, in which $\mathbf{a}^{\top}$ denotes the transpose of the vector $\mathbf{a}$. Furthermore, assume that the n independent realization of $\mathbf{x}_{p\times 1}^{\top}$ are stored in the rows of the data matrix $\mathbf{X}_{n\times p}$ which is a common notation in linear regression. One crucial assumption to be verified is the Gaussianity of the data as many classical statistical methods rely on Gaussian distribution. Validity of this assumption for sensor values can be tested using various methods such as analyzing the distribution of individual sensor values, scatter plot of the linear projection of data using principal components, estimating the multivariate kurtosis and skewness, and also multivariate Mardia test, see Figure 3.2.

The aim of this research is to develop a methodology for a wide range of e-noses. For this purpose, we also discuss the inherent dependence structure of gas sensors and the sparse estimation of dependence. Sparse methods are specifically for modeling high-dimensional data. They provide better interpretability and lower the cost of modeling by selecting a subset of features. It is of interest to explore the relationship between the sensors of e-nose for the following reasons.

1. To understand the sensitivity of each sensor to different types of gases. Consequently, one would be able to assign the gas sensors to distinct groups in terms of their measurements, i.e. sensors in the same group measure similar gases.

2. To replace a non-active sensor with its active counterpart. During the sampling process, one or more sensors may stop functioning for an unknown period of time. Having known the existing structure among the sensors, one could swap some of the sensors for the

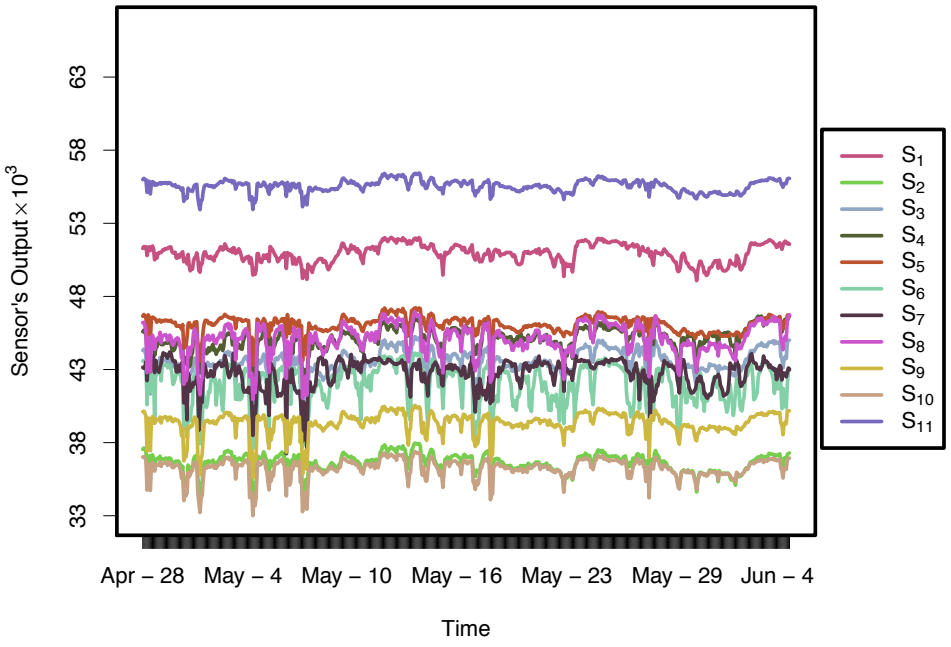

Figure 3.1 Overlay time series plot of 11 sensors.

others within the same group with negligible effect on the analysis of the collected data from the sensors. This, in turn, means excluding the redundant sensors from the study and decreasing the dimension of data.

The covariance matrix of a random vector $\mathbf{x}_{p\times 1}$ is $\mathbf{\Sigma} = [\sigma_{ij}]_{i,j=1,2,\ldots,p}$ where σ_{ij} measures the degree to which two attributes are linearly associated. It is well-known that the inverse of covariance matrix, commonly known as the precision matrix, coincides with the partial correlation between the attributes.

Formally, suppose that the random vector $\mathbf{x}_{p\times 1} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$. The dependence structure of attributes is more comprehensible through the *Gaussian graphical model* (Murphy, 2012, Chapter 26) where an *edge* between two attributes in a graph reveals the conditional dependence between these two attributes given all other existing attributes in the graph.

In order to investigate the inherent dependence between the sensor values, the partial correlation must be estimated. Formally, suppose that the random vector $\mathbf{x}_{p\times 1} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$, and therefore $\mathbf{\Delta} = \mathbf{\Sigma}^{-1}$ is the desired parameter to be estimated. To study the relationship between these p attributes, one can use the *Gaussian graphical model* which is a graph-based representation of a non-causal structure of attributes. There is a one-to-one correspondence

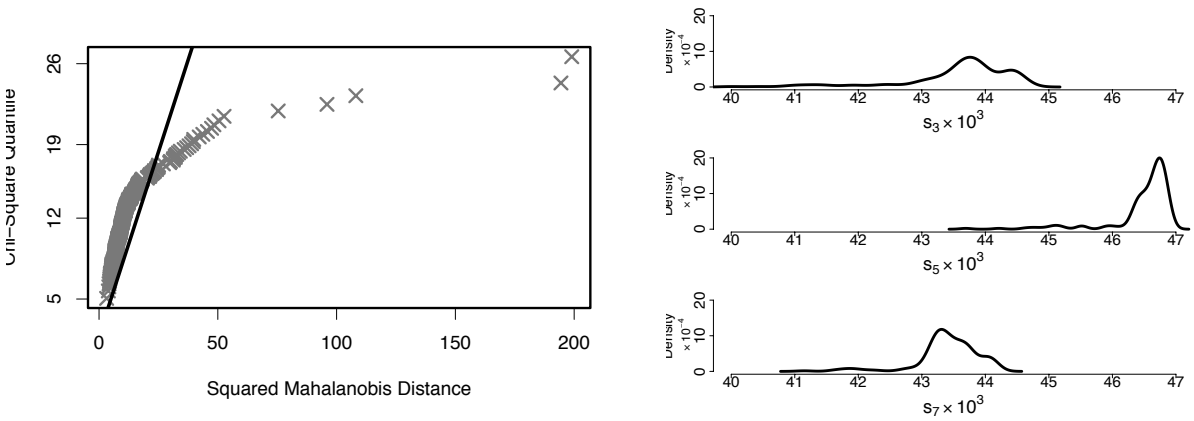

Figure 3.2 Left panel, the Q-Q plot of squared Mahalanobis distance supposed to follow chi-squared distribution for Gaussian data. Right panel, the non-parametric marginal density estimation for some randomly chosen sensor values. Both graphs confirm the non-Gaussianity of the data.

between the elements of the precision matrix, $\mathbf{\Delta}$, and the edges in the Gaussian graphical models. Thus non-zero elements of $\mathbf{\Delta}$ imply conditional dependence and the sparse estimation of $\mathbf{\Delta}$ reveals block dependence structure of attributes. The sparse estimation of $\mathbf{\Delta}$ sets some of the off-diagonal $\mathbf{\Delta}$ entries exactly to zero. The graphical lasso (Friedman *et al.*, 2008) sparsely estimates graphs using the Gaussian log-likelihood with a *lasso penalty* (Tibshirani, 1996). Various techniques were suggested for estimating $\mathbf{\Delta}$ sparsely, such as Meinshausen and Buhlmann (2006); Yuan and Lin (2007); Banerjee *et al.* (2008). Assuming that the attributes are centered, the log-likelihood for n realizations of a random vector $\mathbf{x}_{p\times 1}$, $\mathbf{x}_{p\times 1} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ is

$$\ell(\mathbf{\Delta}) = -\frac{np}{2}\log(2\pi) + \frac{n}{2}\log|\mathbf{\Delta}| - \frac{1}{2}\mathrm{tr}(\mathbf{X}^\top\mathbf{X}\mathbf{\Delta}),$$

where $|.|$ and tr() are the determinant and the trace operators, respectively. The graphical lasso estimates the covariance matrix $\mathbf{\Sigma}$ under the assumption that its inverse, $\mathbf{\Delta}$, is sparse. The graphical lasso minimizes

$$-\log|\mathbf{\Delta}| + \mathrm{tr}(\mathbf{S}\mathbf{\Delta}) + \lambda||\mathbf{\Delta}||_1 \tag{3.1}$$

over the positive semi-definite matrix $\mathbf{\Delta} \geq \mathbf{0}$, where $\mathbf{S} = \frac{1}{n}\{\mathbf{X}^\top\mathbf{X}\}$ is the sample covariance, $||\mathbf{\Delta}||_1$ is the sum of the absolute entries of $\mathbf{\Delta}$ and λ is a regularization parameter. The larger the λ is, the more sparse the estimated precision matrix $\mathbf{\Delta}$ will be. Minimization problem

(3.1) is a semi-definite programming problem— a convex optimization of a linear objective function over positive semi-definite matrices. Using the sub-gradient method, one may solve the optimization problem (3.1)

$$-\boldsymbol{\Delta}^{-1} + \mathbf{S} + \lambda\boldsymbol{\Gamma} = \mathbf{0}, \tag{3.2}$$

where $\boldsymbol{\Gamma} = [\gamma_{ij}]_{i,j=1,2,\ldots,p}$ is the sign of each element of $\boldsymbol{\Delta}$ such that $\gamma_{ij} = \text{sign}(\delta_{ij})$ if $\delta_{ij} \neq 0$ or $\gamma_{ij} \in [-1, 1]$ if $\delta_{ij} = 0$. The graphical lasso employs the block-coordinate technique for solving (3.2). First, matrices $\boldsymbol{\Delta}$ and $\boldsymbol{\Gamma}$ are partitioned as

$$\boldsymbol{\Delta} = \begin{bmatrix} \boldsymbol{\Delta}_{11} & \boldsymbol{\delta}_{12} \\ \boldsymbol{\delta}_{21} & \delta_{22} \end{bmatrix}, \quad \boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{\gamma}_{12} \\ \boldsymbol{\gamma}_{21} & \gamma_{22} \end{bmatrix}, \tag{3.3}$$

where $\boldsymbol{\Delta}_{11}$ is a matrix of dimensions $(p-1) \times (p-1)$, $\boldsymbol{\delta}_{12} = \boldsymbol{\delta}_{21}^{\top}$ is a vector of dimension $(p-1)$ and δ_{22} is a scalar. The matrix $\boldsymbol{\Gamma}$ has the same partitioning structure as $\boldsymbol{\Delta}$. Assuming $\mathbf{W}$ to be an estimate for $\boldsymbol{\Sigma}$, $\mathbf{W} = \boldsymbol{\Delta}^{-1}$, the entries of $\mathbf{W}$ can be calculated using the rule of inverse for a partitioned matrix. After some simplifications, the entries of $\mathbf{W}$ are

$$\mathbf{W}_{11} = \left(\boldsymbol{\Delta}_{11} - \frac{\boldsymbol{\delta}_{12}\boldsymbol{\delta}_{21}}{\delta_{22}}\right)^{-1},$$
$$\mathbf{w}_{12} = \mathbf{w}_{21}^{\top} = -\frac{\boldsymbol{\Delta}_{11}^{-1}\boldsymbol{\delta}_{12}}{\delta_{22} - \boldsymbol{\delta}_{21}\boldsymbol{\Delta}_{11}^{-1}\boldsymbol{\delta}_{12}},$$
$$w_{22} = \frac{1}{\delta_{22} - \boldsymbol{\delta}_{21}\boldsymbol{\Delta}_{11}^{-1}\boldsymbol{\delta}_{12}},$$

and

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{21} & w_{22} \end{bmatrix}.$$

Taking the first $p-1$ elements of pth column of equation (3.2), we may write

$$-\mathbf{w}_{12} + \mathbf{s}_{12} + \lambda\boldsymbol{\gamma}_{12} = \mathbf{0}. \tag{3.4}$$

Substituting $\mathbf{w}_{12}$ into equation (3.4), we have

$$\mathbf{W}_{11}\frac{\boldsymbol{\delta}_{12}}{\delta_{22}} + \mathbf{s}_{12} + \lambda\boldsymbol{\gamma}_{12} = \mathbf{0}. \tag{3.5}$$

The above equation is equivalent to the following L_1 regularized problem,

$$\min_{\boldsymbol{\beta}}\{\frac{1}{2}\boldsymbol{\beta}^\top \mathbf{W}_{11}\boldsymbol{\beta} + \boldsymbol{\beta}^\top \mathbf{s}_{12} + \lambda||\boldsymbol{\beta}||_1\}, \tag{3.6}$$

where $\boldsymbol{\beta} = \frac{\boldsymbol{\delta}_{12}}{\delta_{22}}$ and $\delta_{22} > 0$. This optimization problem corresponds to a lasso regression (Tibshirani, 1996) of pth attribute on the remaining ones where the matrix $\mathbf{S}_{11}$, the sub-matrix of dimensions $(p-1)\times(p-1)$ in the partitioned sample covariance matrix, is replaced by its current estimate $\mathbf{W}_{11}$. The solution to the above problem can be found through the element-wise coordinate descent method. Mazumder and Hastie (2012) suggested a new approach to overcome the occasional convergence issues with the graphical lasso. They proved that the graphical lasso solves the convex dual problem of equation (3.1). In Figure 3.3 (right panel), the undirected graph depicts the estimation of $\boldsymbol{\Delta}$ with $\lambda = 0.75$ by connecting two attributes which are conditionally correlated given all the other attributes. This value of λ is chosen deliberately in order to provide a more clear image of the underlying relation between the sensor values.

For instance, the sensors 9, 10 and 11 are conditionally correlated with each other. This also agrees with the heatmap of the correlation matrix in Figure 3.3 (left panel). The conditional correlation among some of the sensors implies that these sensors are measuring similar gases. This dependence must be taken into account when robust modeling of the e-nose data is of interest.

3.3 Data Validation

To verify the validity of the measurements automatically, it is necessary to have some reference samples for the purpose of comparison. Our first task is to allocate each sample to a meaningful measurement zone, say green, yellow or red to distinguish reliable from unreliable measurements. The reference samples are collected while the e-nose is at its best performance, and the conditions are fully under control. For the dataset under the study, there are two distinct reference sets. *Unlabeled training data* consists of a subset of data in a period of sampling, approved by an expert to be reliable data after installation of the e-nose. *Labeled training data* is manually gathered samples from the field and brought to the olfactometry to quantify its odor concentration. The labeled training data are used for odor concentration prediction, $\mathbf{y}$, by a supervised learning method while unlabeled training data are used for data validation, see Figure 3.4.

Assume that an e-nose is installed in a field and the measuring process starts after the e-nose installation. After a period of time which is defined by an expert, say one week, the data collected in this period constitute the unlabeled training data. *Online data* are the current

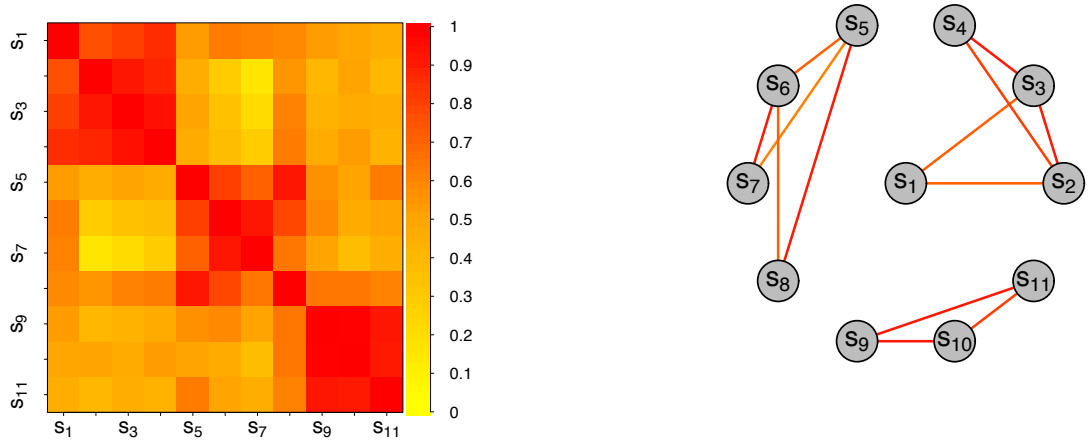

Figure 3.3 Left panel, heatmap of the correlation matrix of the sensor values (s_1–s_{11}). Right panel, the undirected graph of partial correlation estimated using the graphical lasso. The undirected graph of the right panel approves the block diagonal structure of the heatmap of the left panel.

measurements to be validated and predicted. Online data are compared with the unlabeled training data and the data history. If the online data diverge greatly from the overall pattern of data previously seen, they are marked as outliers and are allocated to the red zone. This zone represents a dramatic change in the pattern of samples and is referred to as "risky" samples. If the online data are not outliers and are located within the data convex polytope of the unlabeled training data, they are assigned to the green zone. Convex polytope is a robust version of data confidence region. If data are multivariate Gaussian, then their convex polytope converges to an ellipsoid. This zone represents the "safe" samples. If the online data are not outliers, but outside of the convex polytope of unlabeled training data, they are assigned to the yellow zone. This zone displays potentially "critical" samples. Producing many samples belonging to the yellow and the red zones is an indication of a major flaw in the system.

For the red zone, it is required to find the outliers of online data as they are being sampled by the e-nose. Common outlier detection methods rely on the assumption of elliptical contoured distributions. These common methods must be avoided as this assumption is violated in data exploration. Here, outliers are flagged by means of adjusted outlyingness (AO) criterion (Brys *et al.*, 2006).

For the green and the yellow zones, the online data are projected onto a lower dimension

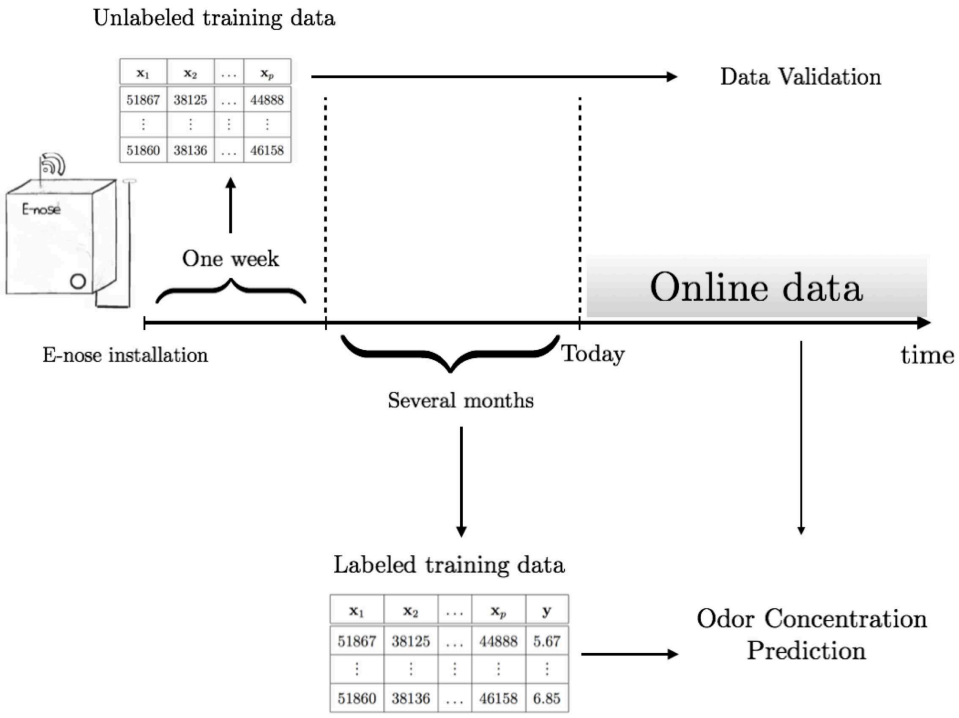

Figure 3.4 Data validation and odor concentration prediction for e-nose data.

subspace. Dimension reduction methods such as principal component analysis (PCA) can serve this purpose (Jolliffe, 2002). PCA exploits empirical covariance matrix, $\hat{\mathbf{\Sigma}}$, which is sensitive to outliers (Prendergast, 2008). Since the data contain many outliers, robust covariance estimation must be applied to avoid misleading results. Therefore, robust principal component analysis (Hubert *et al.*, 2005) is employed for dimension reduction purpose. This robust PCA computes the covariance matrix through projection pursuit (Li and Chen, 1985) and minimum covariance determinant (Croux and Haesbroeck, 2000) methods. The robust PCA procedure can be summarized as follows:

1. The matrix of data is pre-processed such that the data spread in the subspace of at most $\min(n-1, p)$.

2. In the spanned subspace, the most obvious outliers are diagnosed and removed from data. The covariance matrix is calculated for the remaining data, $\hat{\mathbf{\Sigma}}_0$.

3. The estimated covariance matrix $\hat{\mathbf{\Sigma}}_0$ is used to decide about the number of principal components to be retained in the analysis, say k_0 ($k_0 < p$).

4. The data are projected onto the subspace spanned by the first k_0 eigenvectors of $\hat{\mathbf{\Sigma}}_0$.

5. The covariance matrix of the projected points is estimated robustly using minimum covariance determinant method and its k leading eigenvalues are computed. The corresponding eigenvectors are the robust principal components.

The specification of the green and yellow zones requires the computation of the polytope of unlabeled training data. This polytope is built using the convex hull of the robust principal component *scores* (Mirshahi *et al.*, 2016, 2017b). A short description of each zone is provided in Table 3.1. Before determining the color tag for each new data, the samples are checked for missing values and are imputed in case needed by multivariate imputation methods such as Josse *et al.* (2011).

Table 3.1 Description of each zone in validity assessment procedure.

Zone	**Description**
Red	Observations that are outliers in terms of AO measure.
Green	Observations that are non-outliers in terms of AO measure. Moreover, they fall into the polytope of the unlabeled data.
Yellow	Observations that are non-outliers in terms of AO measure. Moreover, they do not fall into the polytope of neither the unlabeled nor the labeled data.

Suppose that $\mathbf{X}_{N\times 11}$ represents the matrix of sensor values for N samples, $\mathbf{y}_N$ the vector of corresponding odor concentration values and $\mathbf{x}_l^\top$ is the lth row of $\mathbf{X}_{N\times 11}$, $l = 1, 2, \ldots, N$. Furthermore, suppose that n_1 refers to the number of samples in the unlabeled data and n_2 refers to the number of samples in the labeled dataset. Two different scenarios occur based on the availability of the labeled dataset. If the labeled dataset is accessible, then Scenario 1 happens. Otherwise, we only deal with Scenario 2. Scenario 1 is a general case which is explained more in details. The data undergo a pre-processing stage, including imputation and outlier detection, before any further analysis. Having done the pre-processing stage, data are stored as the unlabeled data, $\mathbf{X}_{n_1\times 11}$, and the labeled data, $\mathbf{X}_{n_2\times 11}$. The first k, e.g. $k = 2, 3$, robust principal components of $\mathbf{X}_{n_1\times 11}$ are calculated and the corresponding loading matrix is denoted by $\mathbf{L}_1$. The pseudo code of two algorithms for Scenario 1 is provided below. In Scenario 2, there is no model for odor concentration prediction in the Main Algorithm. Figure 3.5 shows the data validation during the sampling process for 700 sensors' measurements.

Sub-Algorithm (Scenario 1)

1: **if** the point $\mathbf{x}_l^\top, l = 1, 2, \ldots, N$ is identified as an outlier by AO measure **then**
2: $\quad \mathbf{x}_l^\top$ is in red zone,
3: **else if** $\mathbf{x}_l^\top \mathbf{L}_1 \in \text{ConvexHull}^{(1)}$ AND $\mathbf{x}_l^\top \mathbf{L}_1 \notin \text{ConvexHull}^{(2)}$ **then**
4: $\quad \mathbf{x}_l^\top$ is in green zone,
5: **else**
6: $\quad \mathbf{x}_l^\top$ is in yellow zone.
7: **end if**

Main Algorithm (Scenario 1)

Require: $\mathbf{X}_{n_1 \times 11}$, $\mathbf{X}_{n_2 \times 11}$, and the loading matrix $\mathbf{L}_1$ using robust PCA over the unlabeled data, $\mathbf{X}_{n_1 \times 11}$.

1: $\text{ConvexHull}^{(1)} \leftarrow$ the convex hull of the projected values of the unlabeled data, $\mathbf{X}_{n_1 \times 11}\mathbf{L}_1$.
2: Train a supervised learning model on the labeled data, $\mathbf{X}_{n_2 \times 11}$, and its odor concentration vector, $\mathbf{y}_{n_2}$.
3: $\text{ConvexHull}^{(2)} \leftarrow$ the convex hull of the projected values of the labeled data, $\mathbf{X}_{n_2 \times 11}\mathbf{L}_1$.
4: Do **Sub-Algorithm** for new data $\mathbf{x}^*$.
5: Predict the odor concentration for new data $\mathbf{x}^*$ using the trained supervised learning model.

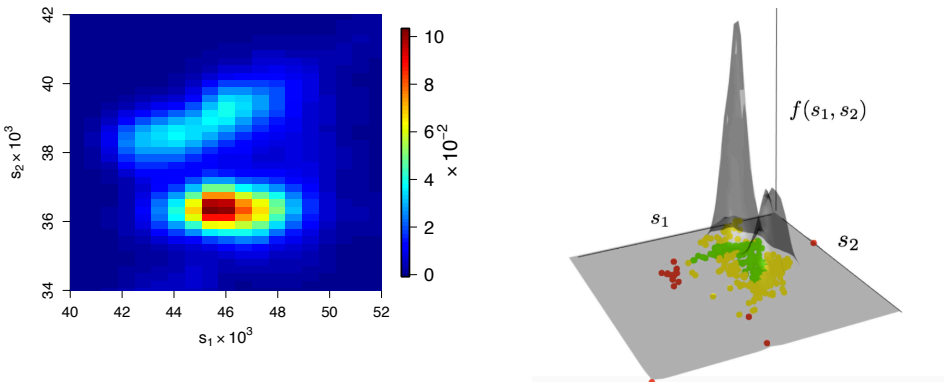

Figure 3.5 Data validation for about 700 samples using two sensors. Left panel, the plot illustrates the contour map of estimated density function for two sensors. Right panel, the density function of the samples demonstrated in $3D$ with zones identified for each of the samples in the sensor 1 (s_1) versus sensor 2 (s_2) plane. Higher densities are assigned to "safe" zones compared to "critical" and "risky" zones.

3.4 Computational Complexity

Here, we discuss the computational complexity of our proposed algorithm (Main Algorithm). We start with a brief introduction to computational complexity.

The computational complexity of an algorithm is studied asymptotically by the big O-notation (Arora and Barak, 2009). The big O-notation explains how quickly the run-time of an algorithm grows relative to its input. For instance, sum of n values requires $(n-1)$ operations. Consequently, the mean requires n operations reserving one for the division of the sum by n. As they are both bounded by a linear function, they have computational complexity of order $\mathcal{O}(n)$. In other words, the performance of the sum and mean grow linearly and in direct proportion to the size of the input. Note that not all algorithms are computationally linear. Computational complexity of covariance matrix, for instance, is $\mathcal{O}(np^2)$ where n is the sample size and p is the number of attributes. Since each covariance calls for sum of the pairwise cross-products each of complexity $\mathcal{O}(n)$. In total, there are $\frac{p(p-1)}{2}$ off-diagonal cross products and p square sums for the diagonal entries of the covariance matrix. This yields $n\{p(p-1)+p\}$ operations. For a fixed number of attributes p, the computation is of order $\mathcal{O}(n)$. Likewise, for a fixed number of observations the computation is of order $\mathcal{O}(p^2)$. Another nontrivial example for non-linear algorithm is PCA or the robust PCA. Computation of robust principal components involves various operations discussed in Section 3.3. Computational complexity of robust PCA is described below. Computation of robust PCA comprises the following steps:

1. Reducing the data space to an affine subspace spanned by the n observations using singular value decomposition of $(\mathbf{X}-\mathbf{1}_n\hat{\boldsymbol{\mu}})^\top(\mathbf{X}-\mathbf{1}_n\hat{\boldsymbol{\mu}})$, where $\mathbf{1}_n$ is the column vector of n dimension with all entries equal to 1. This step is of order $\mathcal{O}(p^3)$, see Golub and Loan (1996) and Holmes *et al.* (2007).

2. Finding the least outlying points using the Stahel-Donoho affine-invariant outlyingness (Stahel, 1981; Donoho, 1982). Adjusting this outlyingness measure by the minimum covariance determinant location and scale estimators is of order $\mathcal{O}(pn\log n)$, see Hubert and Van der Veeken (2008) and Hubert *et al.* (2005). Then the covariance matrix of the non-outliers data, $\hat{\boldsymbol{\Sigma}}_0$, is calculated which is computationally less expensive.

3. Performing the principal component analysis on $\hat{\boldsymbol{\Sigma}}_0$ and choosing the number of projection components (say $k_0 < p$) to be retained. Computing the $\hat{\boldsymbol{\Sigma}}_0$ needs np^2 operations. Thus its complexity is $\mathcal{O}(np^2)$. The spectral decomposition of the covariance matrix is achieved by applying matrix-diagonalization method, such as singular value decomposition or Cholesky decomposition. This results in $\mathcal{O}(p^3)$ computational complexity.

Determining the k_0 largest eigenvalues and their corresponding eigenvectors has time complexity of $\mathcal{O}(k_0 p^2)$ (Du and Fowler, 2008). As a result, the time complexity of this step is $\mathcal{O}(np^2)$.

4. Projecting the data onto the subspace spanned by the first k_0 eigenvectors, i.e $(\mathbf{X} - \mathbf{1}_n\hat{\boldsymbol{\mu}})\mathbf{P}_{p\times k_0}$, where $\mathbf{P}_{p\times k_0}$ is the matrix of eigenvectors corresponding to the first k_0 eigenvalues. This step has $\mathcal{O}(npk_0)$ time complexity.

5. Computing the covariance matrix of the projected points using the method of fast minimum covariance determinant has the computational complexity which is sub-linear in n, for fixed p. This is $\mathcal{O}(n)$ (Rousseeuw and Driessen, 1999). The calculation of the spectral decomposition of the final covariance matrix has at maximum $\mathcal{O}(nk_0)$ time complexity.

Considering the worst case complexity in the above steps, we conclude that the computational complexity of robust PCA is $\mathcal{O}(\max\{pn\log n, np^2\})$, and hence equal to $\mathcal{O}(p^2 n\log n)$.

To ascertain the complexity of the Main Algorithm, one needs to analyze each step separately. The measurement validation in e-nose requires the calculation of certain steps of the Main Algorithm including Step Require, Step 1, Step 3, and Step 4. All these tasks excluding Step 4 of the Main Algorithm (Sub-Algorithm) must be run only once. Step 4 duplicates upon the arrival of the new observations.

First, we start by evaluating the complexity of Step Require, Step 1, and Step 3 that should be run once. Afterwards Step 4 is analyzed in a similar fashion. Note that for the e-nose data, the number of samples is generally much greater than the number of sensors p. In addition, as the number of sensors p is fixed in an e-nose equipment, the computational complexity is reported as the function of number of samples only.

The Main Algorithm starts with the robust PCA over the unlabeled data. As a result, Step Require has $\mathcal{O}(\{n_1\log n_1\})$ complexity assuming p to be fixed. Step 1 requires $\mathcal{O}(n_1 k_0)$ computing time for computing $\mathbf{X}_{n_1\times 11}\mathbf{L}_1$, where k_0 stands for the the number of eigenvectors retained in the loading matrix $\mathbf{L}_1$. Computing the convex hull of these projected values for $k_0 \leq 3$ is of order $\mathcal{O}(n_1\log n_1)$. For $k_0 > 3$, the computational complexity of hull increases exponentially with k_0, see Ottmann *et al.* (1995) and Chan (1996), or one may use approximations such as Cutler and Breiman (1994). Similarly, the same complexity is valid for Step 3. Performing some pre-processing steps on the labeled or unlabeled datasets including outlier detection using AO measure has $\mathcal{O}(n_1\log n_1)$ complexity (Hubert and Van der Veeken, 2008) assuming that $n_1 > n_2$ which is common in practise. As a result, Step Require, Step 1, and Step 3 performed only once, take $\mathcal{O}(n_1\log n_1)$ run-time.

Now, we analyze Step 4 in terms of its computational complexity. Step 4 mainly does the following three tasks.

i) Accumulating the new observations with the past history by stacking the matrix of observations from time 1 to time $t-1$ row-wise with vector of observations at time t, $\mathbf{X}_{1:t}^{\top} = [\mathbf{X}_{1:t-1}^{\top} \mid \mathbf{x}_t]$, where $n_1 < t \leq N$, and identifying outliers using AO measure. This has computational complexity of $\mathcal{O}(t \log t)$.

ii) Projecting the observations onto the space of unlabeled data, $\mathbf{x}_l^{\top} \mathbf{L}_1$. This is a simple matrix product and has the computational complexity of $\mathcal{O}(k_0 p)$.

iii) Verifying whether the projection of data, $\mathbf{x}_l^{\top} \mathbf{L}_1$, locates within the convex hull of either unlabeled data or labeled data, which is equivalent to solving a linear optimization with linear constraints (Kan and Telgen, 1981; Dobkin and Reiss, 1980). The algorithm used for this purpose has computational complexity which varies quadratically with respect to the number of vertices of the convex hull, and has $\mathcal{O}(n_1^2 k_0)$ complexity in the worst case. The R code used for solving this linear program resembles the MATLAB code [1] and is available upon request.

Thus, the computational complexity of Step 4 is $\mathcal{O}(t \log t)$ as in practice the convex hull of unlabeled data is computed, in Step 1, and kept fixed prior to this step.

The mean CPU time in seconds for Step Require, Step 1, and Step 3 that need to be run once and Step 4 which duplicates for each new sample, are reported in Figure 3.6.

[1] http://www.mathworks.com/matlabcentral/fileexchange/10226-inhull

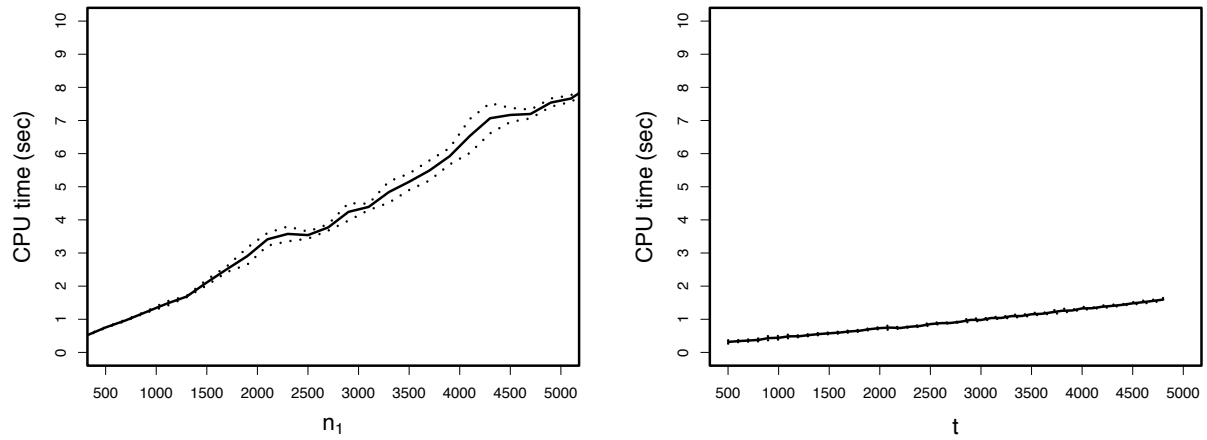

Figure 3.6 The solid line shows the mean CPU time in seconds as a function of input being run on Intel Core i5 1.3 GHz . The dashed lines depict the lower and the upper bound of the 95% confidence interval for the mean CPU time. Left panel, the run-time corresponding to Step Require, Step 1, and Step 3 as the function of the number of samples in unlabeled data, n_1. Right panel, the run-time associated with Step 4 as a function of the total number of samples up to the moment, t. In each iteration, 100 new observations are sampled.

Figure 3.6 confirms that the run-times for the ensemble of the steps Require, 1, and 3 and the Step 4 agree with the computational complexity evaluated theoretically earlier. This implies that measurement validation can be achieved with $\mathcal{O}(t \log t)$ time complexity employing our proposed method.

3.5 Odor Concentration Prediction

The ultimate goal of this section is to suggest a suitable model for predicting odor concentration (Mirshahi *et al.*, 2017a). The data validation serves as a method for analyzing the quality of obtained predictions.

During odor testing, the most common variable of interest is the odor concentration which is evaluated by the olfactometer. The odor concentration of a gaseous sample of odorants is determined by presenting the sample to a panel of selected and screened humans. In order to determine the dilution factor at the 50% detection threshold, the concentration of sample is varied by diluting with neutral gas. At that dilution factor the odor concentration is 1 ou_E/m^3 (European odor unit per cubic meter). The odor concentration of the examined sample is then expressed as a multiple of 1 ou_E/m^3 at standard conditions for olfactometry. Only small proportions of the samples are selected for the examination of their concentrations (labeled data). Consequently, small proportions of data are available for the modeling stage. Here, sparse partial robust M-regression (SPRM) (Hoffman *et al.*, 2015) is used for modeling

the data. SPRM is a new method of modeling which combines sparseness and robustness with the classical partial least square regression. This regression is claimed to be robust with respect to both response and leverage outliers. Although sparse methods are mostly designed for high-dimensional data, they can be advantageous if applied to low-dimensional data as well (Filzmoser *et al.*, 2012).

In a linear regression setting, the relationship between the attributes and the response variable is formulated as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is the vector of measurement errors. The estimate of regression coefficients, $\boldsymbol{\beta}$, is computed through the ordinary least squares $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$. However, there are often situations where the matrix $\mathbf{X}^\top\mathbf{X}$ is not invertible: 1) the attributes are highly correlated and 2) the number of attributes, p, is larger than the number of samples, n. Partial least squares (PLS) regression (Wold, 1966) is used as an alternative to the ordinary least squares regression while $\mathbf{X}^\top\mathbf{X}$ is ill-conditioned. The PLS method projects the data onto a number of latent components and then models the components by one dimensional linear regression, see Manne (1987); Hoskuldsson (2005). Chun and Keles (2010) combined the feature selection with dimension reduction techniques which led to sparse partial least squares regression. This sparse PLS regression produces sparse linear combinations of original attributes based on the least angle regression of Efron *et al.* (2004).

The classical least squares method suits Gaussian errors. In the case of heavy-tailed errors, Cauchy distribution or ε-contaminated Gaussian distributions, the M-estimators tend to provide more promising results (Huber, 1981). Serneels *et al.* (2005) introduced *partial robust M-regression* by embedding the M-estimators in the PLS.

The sparse partial robust M-regression (SPRM) has the characteristics of both partial robust M-regression and sparse PLS in its inner nature. Here, we briefly explain the SPRM regression procedure. The latent linear components, say $\mathbf{T}$, in PLS are defined as linear combinations of the original attributes, $\mathbf{T} = \mathbf{XA}$. The columns of $\mathbf{A}$, the direction vectors $\mathbf{a}_h$, maximizes

$$f\mathbf{a}_h = \underset{\mathbf{a}}{\operatorname{argmax}}\ \mathrm{Cov}^2(\mathbf{Xa}, \mathbf{y}) \text{ for } h = 1, \ldots, h_{\max}$$
$$\text{s.t. } ||\mathbf{a}_h||_2 = 1 \text{ and } \mathbf{a}_h^\top\mathbf{X}^\top\mathbf{X}\mathbf{a}_i = 0, \tag{3.7}$$

for $1 \leq i < h$, where $||.||_2$ is the L_2-norm ($||\mathbf{x}||_2 = \sqrt{\sum_{i=1}^{p} x_i^2}$). The $h_{\max}$ refers to the maximum number of components we prefer to keep in the study. It is assumed that all the attributes and the corresponding response variable $\mathbf{y}$ are centered, such that $\hat{\mathrm{Cov}}^2(\mathbf{Xa}, \mathbf{y}) = \frac{1}{(n-1)^2}\mathbf{a}^\top\mathbf{X}^\top\mathbf{y}\mathbf{y}^\top\mathbf{Xa}$. On the other hand, $\mathbf{y}$ can be decomposed as $\mathbf{y} = \mathbf{Tv} + \boldsymbol{\varepsilon}$. This equation can be rewritten as $\mathbf{y} = \mathbf{XAv} + \boldsymbol{\varepsilon}$, where $\mathbf{Av}$ is the vector of coefficients, $\boldsymbol{\beta}$, that relates

$\mathbf{y}$ to the original attributes in $\mathbf{X}$. Once the matrix $\mathbf{A}$ is found and $\mathbf{v}$ is estimated through the ordinary least squares method, the estimates of the coefficients are obtained by $\hat{\boldsymbol{\beta}} = \mathbf{A}\hat{\mathbf{v}}$. To make the PLS results robust in the presence of outliers, some weights, $\omega_i \in [0, 1]$; $i = 1, 2, \ldots, n$, are assigned to each row of $\mathbf{X}$ and $\mathbf{y}$. Outliers are given a weight smaller than one. Suppose that $\mathbf{t}_i$ is the ith column of the matrix $\mathbf{T}$ and $e_i = y_i - \mathbf{t}_i^{\top}\hat{\mathbf{v}}$ is the residual of the latent variable regression model. The weight, ω_i, is;

$$\omega_i^2 = \omega_R(\frac{e_i}{\hat{\sigma}})\omega_T\Big(\frac{||\mathbf{t}_i - \text{median}_j(\mathbf{t}_j)||_2}{\text{median}_i||\mathbf{t}_i - \text{median}_j(\mathbf{t}_j)||_2}\Big),$$

where $\hat{\sigma}$ is the median absolute deviation of the residuals, ω_R and ω_T are the Hampel weighting function with quantiles of standard Gaussian and chi-squared distribution (Hampel *et al.*, 2011). In order to obtain a robust PLS, equation (3.7) should be rewritten in terms of $\tilde{\mathbf{X}} = \boldsymbol{\Omega}\mathbf{X}$ and $\tilde{\mathbf{y}} = \boldsymbol{\Omega}\mathbf{y}$, where $\boldsymbol{\Omega}$ is a diagonal matrix with diagonal elements of ω_i, $i = 1, 2, \ldots, n$. A fully robust version of PLS requires estimating $\mathbf{v}$ robustly using *M-estimators*. Moreover, if an L_1 penalty is imposed while computing direction vectors, $\mathbf{a}_h$, the product is a sparse version of the PLS. Zou *et al.* (2006) suggest penalization on a surrogate direction vector, say $\mathbf{c}$, yields sufficiently sparse estimates. Therefore, using Zou *et al.* (2006) suggestion, (3.7) can be transformed to

$$\begin{aligned}\min_{\mathbf{a},\mathbf{c}} &- \kappa\mathbf{a}^{\top}\tilde{\mathbf{X}}^{\top}\tilde{\mathbf{y}}\tilde{\mathbf{y}}^{\top}\tilde{\mathbf{X}}\mathbf{a}, +(1-\kappa)(\mathbf{c}-\mathbf{a})^{\top}\tilde{\mathbf{X}}^{\top}\tilde{\mathbf{y}}\tilde{\mathbf{y}}^{\top}\tilde{\mathbf{X}}(\mathbf{c}-\mathbf{a}) + \lambda_1||\mathbf{c}||_1 \\ &\text{s.t. } ||\mathbf{a}_h||_2 = 1 \text{ and } \mathbf{a}_h^{\top}\tilde{\mathbf{X}}^{\top}\tilde{\mathbf{X}}\mathbf{a}_i = 0. \end{aligned} \tag{3.8}$$

The desired direction vector is given by $\mathbf{a}_h = \frac{\hat{\mathbf{c}}}{||\hat{\mathbf{c}}||_2}$, where $\hat{\mathbf{c}}$ is the estimate of the surrogate vector acquired from (3.8). For more details on SPRM, see Chun and Keles (2010) and Hoffman *et al.* (2015).

3.6 Simulation

We first examine the effectiveness of our algorithm on a simulated dataset. The data are simulated using the same setup appeared in Hoffman *et al.* (2015) such that the data resemble e-nose measurements. Consider the linear model $\mathbf{y} = \mathbf{XAv} + \boldsymbol{\varepsilon}$, the details of which are explained earlier in Section 3.5. Let $\mathbf{X}_{500\times 11}$ be a data matrix generated according to the multivariate Gaussian distribution with 30% contamination and a random covariance matrix, such the final data are highly correlated over some of the attributes. The matrix of direction vectors, $\mathbf{A}$, is generated such that only the first 4 attributes are predictive of the response

variable $\mathbf{y}$, that is:

$$\mathbf{A}_{11\times h_{\max}} = \begin{bmatrix} \mathbf{A}_{4\times 4} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

The non-zero part of $\mathbf{A}$ is the eigenvectors of $\mathbf{X}_{n\times 4}^{\top}\mathbf{X}_{n\times 4}$, which is the design matrix of the first 4 attributes, each measured over $n = 500$ samples. The components of $\mathbf{v}$ are generated from the uniform distribution on the interval $(0.5, 1.5)$. The error terms, ε_i's, $i = 1, 2, \ldots, 500$, are simulated from standard Gaussian distribution. In order to add some additional outlier effects in our study, 10% of the error terms are generated from $\mathcal{N}(3, 0.3)$ instead of $\mathcal{N}(0, 1)$, giving a contaminated Gaussian mixture overall, see Figure 3.7.

We review the zone assignment step to describe the procedure more in detail. For this purpose, we need to define the reference sets initially. For an easy understanding and a better visualization, only the first two attributes are used for the computation of the zone assignments. The unlabeled data are a random sub-sample of size $n_1 = 200$ from $\mathbf{X}_{500\times 2}$ data matrix. The labeled dataset corresponds to another random sub-sample of size $n_2 = 50$ from $\mathbf{X}_{500\times 2}$ data matrix and it contains only 2/3 of contaminated data. The result of data validation on the simulated dataset is visualized in Figure 3.7.

For the odor concentration prediction, the two models of PLS and SPRM are tried. Primarily, the optimum values of the parameters for each model is computed using 5-fold cross-validation proposed in the literature (Hastie *et al.*, 2001, Chapter 7). As an example, for SPRM model, computation is run over a grid of different values of components ($h_{\max}$) and the shrinkage parameter (λ_1). The 15% trimmed means squared error of prediction (MSE_{pred}) is used for the final selection of the parameters in the 5-fold cross-validation procedure. Once the parameters are determined, models are compared in terms of their prediction error in 200 rounds of computations. The obtained results are summarized in Table 3.2. It shows that the optimum number of components are set to 4 for both models, while SPRM suggests a shrinkage parameter $\lambda_1 = 0.71$. In terms of prediction power, the two models compete closely with each other. The main advantage of SPRM is its feature selection ability while

Table 3.2 Specification of the parameters for the PLS and SPRM models in 200 repetitions.

Model	$h_{\max}$	λ_1	MSE_{pred} (s.d.)	Average number of zero β's (s.d.)
PLS	4	.	2.01 (0.18)	0 (0)
SPRM	4	0.71	2.38 (0.45)	2.6 (1.45)

modeling, and this counts as a great asset in high-dimensional data— perhaps for e-nose equipment with more gas sensors. The SPRM model produces a more parsimonious and easy to interpret direction vectors compared with the ordinary PLS. In addition, SPRM models

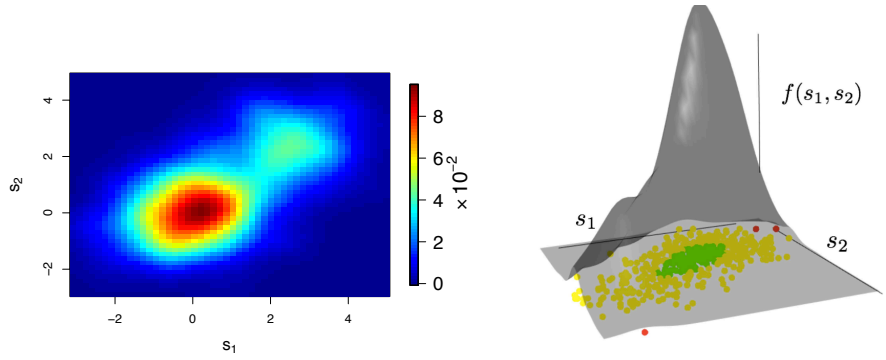

Figure 3.7 Data validation for about 500 samples based on two attributes generated from bivariate Gaussian distribution with 30% contamination. Left panel, the plot illustrates the contour map of the estimated density function for two attributes. Right panel, the density function of the samples demonstrated in $3D$ with zones identified for each of the samples in the attribute s_1 versus attribute s_2 plane.

estimate the coefficients which are robust with respect to various types of outliers.

3.7 Application

The algorithms of Section 3.3 and Section 3.5 are implemented using over 8 months of data collected by the e-nose equipment. The first 3 robust principle components of the data are used for the zone assignment stage. These components correspond to the 3 largest eigenvalues of the covariance matrix. The odor concentration of the sample, $\hat{y}$, is evaluated using PLS and SPRM models. The zone color associated with each set of sensors' measurements and their corresponding odor concentration are plotted in Figure 3.8.

The data contain no measurements on the odor concentration of samples, but rather there is some prior information on its habitual behavior at the specific field that the e-nose was installed. Given the prior knowledge, it is expected that odor maintains high levels of concentration for the month of April until the end of July. The odor concentration is anticipated to drop to small values for the month of August and then to increase steadily over the next months. It is also known that odor concentration should not be over 1000 ou_E/m^3 for the industrial site where the data were collected. Using 10-fold cross-validation (another common choice in the literature (Hastie *et al.*, 2001, Chapter 7)), the optimum number of components for the two models is $h_{\max} = 2$. For the SPRM model, $\lambda_1 = 0.1$ and the first hidden component is only a function of 10 sensor values; the s_2 was eliminated by this choice of λ_1. The predictions based on the SPRM and PLS models closely follow each other, see Figure 3.8. However, the predictions for the SPRM model do not have as high

peaks as the PLS model. The predicted values of both models failed to increase to higher levels for the months of September to November.

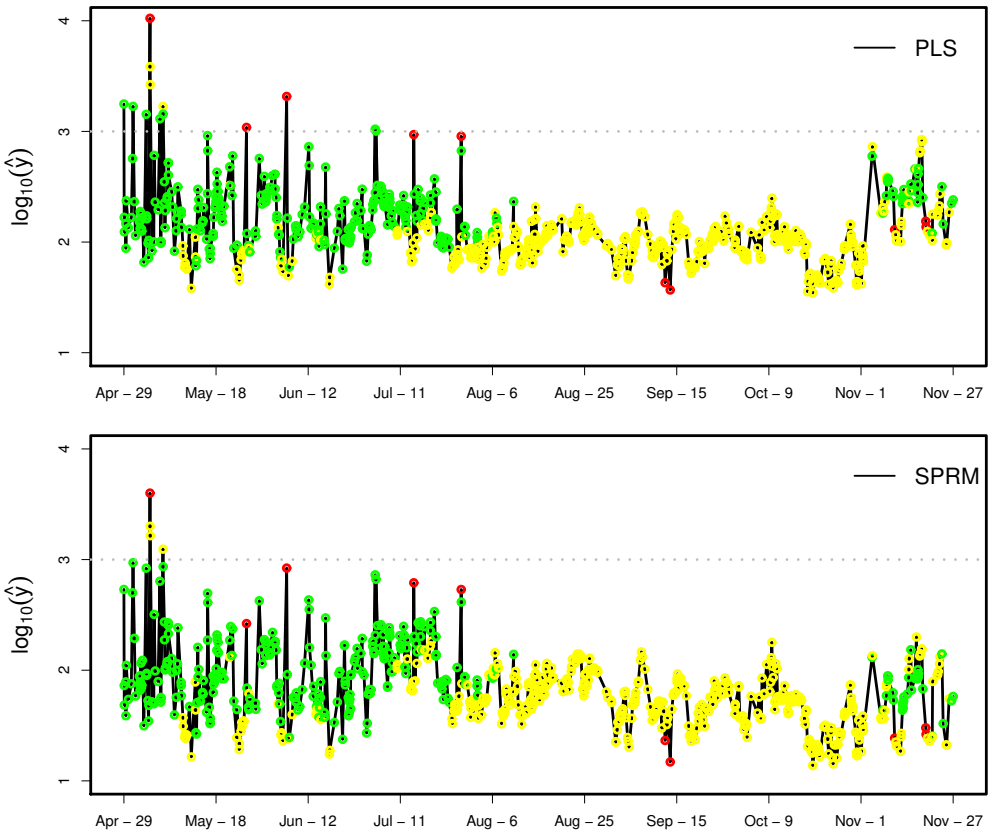

Figure 3.8 A random sample of size $n = 800$ over time and their predicted odor concentrations according to the SPRM and the PLS models. The coloured circles show the associated zone color to each of the samples. The number of hidden components used in the study is $h_{\max} = 2$ and $\lambda_1 = 0.1$.

The zones' definition is helpful in interpreting the results. As an example, the green zone reveals the fact that the sampling points are very close to the samples that have already been observed in unlabeled dataset. The observations in unlabeled dataset were entirely under control, therefore, the green zone justifies the credibility of samples. Consequently, the prediction obtained over these samples is expected to be more accurate. On the contrary, the prediction values for the points in the yellow zone are less accurate. The potential outliers are reported in the red zone. Our described methodology reveals that the predicted values of such data can be misleading; producing a noticeable percentage of samples belonging to the yellow and the red zones. Such findings indicate a possible failure of the e-nose equipment, and hence the need for an on-site visit by a technician.

CHAPTER 4 CONCLUSION

This research has explored two different problems: shape clustering, and odor prediction. In this Chapter, we briefly discuss each problem and our proposed methodologies as well as the directions for the future work.

4.1 Shape Clustering

Geometrical properties of shapes have been extensively studied using various image analysis techniques and statistical measures. In Chapter 1, we suggested a new approach for surface modeling of shapes such as biological cell shapes. We discussed the role of statistical modeling using different basis functions. To adapt basis functions to surface modeling, we considered some constrains for estimating parameters in the modeling phase. In this approach, we regard the surface of each shape as a continuous function rather than discrete landmarks.

The investigation of physical structure of cells, as simple closed shapes, is highly informative in biology, specifically for cancer diagnosis. The result, in this preliminary work, proves that the suggested methodology is quite applicable and can produce promising results.

Having modeled the surface of shapes, the final goal for us is to distinguish between shapes through some clustering methods. In this thesis, we proposed a new information criterion for model-based clustering of linear models and called it CLUSBIC.

The model-based clustering approach can be divided into two main streams: 1) mixture models (Fraley and Raftery, 2002; Yeung *et al.*, 2001; Fraley and Raftery, 2007), 2) product partition models (Hartigan, 1990; Booth *et al.*, 2008; Barry and Hartigan, 1992; Casella *et al.*, 2014). Our new technique, CLUSBIC, falls into the latter model-based approach, with the distinction of considering product partition models for linear models.

In this thesis, we considered the Gaussian conjugate priors, see Section 2.1, to favor computational simplicity. We assumed that the variance of error, σ^2 is constant over different shapes. The assumption of different variances may lead to junk clusters in hierarchical clustering (Smith *et al.*, 2008). In Section 2.5, we proved the consistency of CLUSBIC in clustering. Note that in our settings, the increase in number of observations N does not necessarily imply the increase in the number of clusters $\mathcal{C}(\mathbf{d})$, contrary to the classical clustering problem. Therefore, the consistency of CLUSBIC remains valid.

In Section 2.6, we assumed a uniform distribution on $\mathcal{C}(\mathbf{d})$, and a multinomial-Dirichlet on the number of shapes in each cluster. This Drichelet prior gives more weight to a grouping vector $\mathbf{d}$ with fewer number of clusters. In other words, the Drichlet prior promotes parsimonious models. It would be interesting to explore other priors such as Ewens-Pitman prior

or hierarchichal uniform prior proposed by Casella *et al.* (2014).

Dendrograms are used mainly in this thesis for visualization and for exploring the posterior mode. Besides, we employed random Gibbs sampling as a stochastic search algorithm for computing the posterior probabilities. As the number of possible grouping gets exponentially large with relatively small number of shapes, it would be interesting to develop an efficient stochastic search algorithm that does not require the exploration of whole space.

In Section 2.9.1, we discussed the importance of hyper-parameters in the final number of clusters produced by the model. We propose exploring other possible methods for estimating these parameters. One may consider a proper prior on these parameters and verify the effect of the prior on the final clustering result.

4.2 Odor Prediction

The ability to recognize chemicals in the environment is a basic and essential need for the living organisms. All species are provided with a chemical awareness system. Living species employ their chemical senses to approach safe conditions. In many ways, olfaction is probably one of the most important senses and critical for survival in a wide range of living species.

Electronic nose (e-nose) devices have received continuous attention in the field of sensor technology. The applications of e-nose include industrial production, processing, manufacturing, mainly in quality control, grading, processing controls, and gas leak detection.

The measurement quality of the e-nose depends on its sensors' performance. Due to the high variability of gases in the air and the sensitivity of the sensor values, e-nose measurements can fluctuate very often and fail to maintain a certain level of precision. An automatic procedure that detects the samples credibility in an online fashion has been a technical shortage for a long time and was addressed in this work. The smart olfaction provides an automated process for assessing the validity of samples and predicting the odor concentration accurately during the sampling procedure and eliminates the need for unnecessary personnel.

The majority of manufactured sensor arrays suffer from drift and cross-sensitivity, which render the sensor values unstable and less sensitive to odors. The behavior of a sensor is directly influenced by the surrounding chemical and physical conditions.

Our first contribution was to assemble various statistical methods to be used as an algorithm for anomaly detection. This algorithm takes the statistical properties of sensors' measurements into account to assure a reliable and a statistically robust anomaly detection tailored for e-nose data.

Many studies have been devoted to faulty sensor detection, such as Fonollosa *et al.* (2012, 2013); Padilla *et al.* (2010b). In this thesis, we focused on quality of e-nose measurements

rather than identifying the individual faulty sensors. Although it would be interesting to consider the diagnosis of faulty sensors from different perspectives. One may consider the data associated to each sensor as a continuous function and apply the same methodology proposed in Chapter 2 of this thesis or any other model-base clustering method.

Another interesting problem in this domain would be to compare the sensor measurements between two or various e-noses installed in the same field. This comparison helps us in evaluating the performance of e-noses. To this end, one may treat the sensors' measurement over a period of time as multivariate time series data. Each e-nose is represented by a multivariate time series model. Consequently, one can assign e-noses to different clusters depending on their sensors' measurements using clustering techniques for time series data, see Singhal and Seborg (2005); Liao (2005); Keogh *et al.* (2001).

BIBLIOGRAPHY

AMEMIYA, T. (1985). *Advanced econometrics*. Harvard university press.

AMINE, H., BAZZO, S. and LABRECHE, S. (1999). Intensity and quality discrimination using the fox4000 gas sensor array system in: Electronic noses and sensor array bases system. *Design and Applications*, W.J. Hurst (Ed.), Technomic Co., Lancaster, PA. 235–248.

ANDREWS, D. and MALLOWS, S. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society, Series B*, 36, 99–102.

ARORA, S. and BARAK, B. (2009). *Computational complexity: A Moden approach*, Cambridge University Press.

ARTURSSON, T., EKLOV, T., LUNDSTROM, I., MARTENSSON, P., SJOSTROM, M. and HOLMBERG, M. (2000). Drift correction methods for gas sensors using multivariate methods. *Journal of Chemometrics*, 14, 711–723.

ATTEIA, M. (2014). *Hilbertian kernels and spline functions*, vol. 4. Elsevier.

BANERJEE, O., GHAOUI, L. E. and D'ASPREMONT, A. (2008). Model selection through sparse maximum likelihood estimation. *Journal of Machine Learning Research*, 9, 458–516.

BARRY, D. and HARTIGAN, J. A. (1992). Product partition models for change point problems. *The Annals of Statistics*, 260–279.

BARTELS, R. H., BEATTY, J. C. and BARSKY, B. A. (1995). *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufmann.

BERG, B. A. (2005). Introduction to Markov chain Monte Carlo simulations and their statistical analysis. *Markov Chain Monte Carlo. Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap*, 7, 1–52.

BERMAK, A., BELHOUARI, S. B., SHI, M. and MARTINEZ, D. (2006). Pattern recognition techniques for odor discrimination in gas sensor array. *Encyclopedia of Sensors*, X, 1–17.

BERNARDO, J. M. and SMITH, A. F. M. (1994). *Bayesian Theory*. Wiley, New York.

BOEKER, P. (2014). On 'electronic nose' methodology. *Sensors and Actuators B: Chemical*, 204, 2–17.

BOOTH, J. G., CASELLA, G. and HOBERT, J. P. (2008). Clustering using objective functions and stochastic search. *Journal of the Royal Statistical Society, Series B*, 70, 119–139.

BORWEIN, P. and ERDÉLYI, T. (2012). *Polynomials and polynomial inequalities*, vol. 161. Springer Science and Business Media.

BOX, G. E. and TIAO, G. C. (2011). *Bayesian inference in statistical analysis*, vol. 40. John Wiley & Sons.

BRECHBÜHLER, C., GERIG, G. and KÜBLER, O. (1995). Parametrization of closed surfaces for 3-d shape description. *Computer vision and image understanding*, 61, 154–170.

BRYS, G., HUBERT, M. and ROUSSEEUW, P. J. (2006). A robustification of independent component analysis. *Chemometrics*, 19, 364–375.

CARLO, S. D. and FALASCONI, M. (2012). Drift correction methods for gas chemical sensors in artificial olfaction systems : techniques and challenges. *Advances in Chemical Sensors*, 305–326.

CASELLA, G., MORENO, E., GIRÓN, F. J. *ET AL.* (2014). Cluster analysis, model selection, and prior distributions on models. *Bayesian Analysis*, 9, 613–658.

CHAN, T. M. (1996). Output-sensitive results on convex hulls, extreme points, and related problems. *Dicrete and Computational Geometry*, 16, 369–387.

CHIB, S. (1995). Marginal likelihood from the gibbs output. *Journal of the Amercian Statistical Association*, 90, 1313–1321.

CHIB, S. and JELIAZKOV, I. (2001). Marginal likelihood from the matropolis hastings algorithm. *Journal of the Amercian Statistical Association*, 96, 270–281.

CHUN, H. and KELES, S. (2010). Sparse partial least squares regression for simulataneous dimension reduction and variable selection. *Journal of Royal Statistics Society, Series B*, 72, 3–25.

CLYDE, M. and GEORGE, E. I. (2000). Flexible empirical bayes estimation for wavelets. *Journal of Royal Statistical Society Series B*, 62, 681–698.

COOTES, T., TAYLOR, C., COOPER, D. and GRAHAM, J. (1995). Active shape models: their training and application. *Computer Vision and Image Understanding*, 61, 38–59.

CROUX, C. and HAESBROECK, G. (2000). Principal components analysis based on robust estimators of the covariance or correlation matrix: Infulence functions and efficiencies. *Biometrika*, 87, 603–618.

CUTLER, A. and BREIMAN, L. (1994). Archetypal analysis. *Technometrics*, 36, 338–347.

DAILEY, M. E., MANDERS, E., SOLL, D. R. and TERASAKI, M. (2006). *Confocal Microscopy of Living Cells*, Springer US, Boston, MA. 381–403.

DAVIE, A. and STOTHERS, A. (2013). Improved bound for complexity of matrix multiplication. *Proceedings of the Royal Society of Edinburgh*, 143A, 351–370.

DAVIS, W. W. (1978). Bayesian analysis of linear model subject to linear inequality constraints. *Journal of the American Statistical Association*, 73, 573–579.

DE BOOR, C., HÖLLIG, K. and RIEMENSCHNEIDER, S. (2013). *Box splines*, vol. 98. Springer Science & Business Media.

DIERCKX, P. (1995). *Curve and Surface Fitting with Splines*. Oxford University Press.

DOBKIN, D. P. and REISS, S. P. (1980). The complexity of linear programing. *Theoritical Computer Science*, 11, 1–18.

DONOHO, D. L. (1982). Breakdown properties of multivariate location estimators. *Ph.D. Qualifying Paper, Harvard University*.

DRYDEN, I. L. and MARDIA, K. V. (1998). *Statistical Shape Analysis*. Wiley.

DU, Q. and FOWLER, J. E. (2008). Low-complexity principal component analysis for hyperspectral image compression. *International Journal of High Performance Computing Applications*, 22, 438–448.

DUCROZ, C., OLIVE-MARIN, J.-C. and DUFOUR, A. (2011). Spherical harmonics based extraction and annotation of cell shape in 3d time-lapse microscopy sequences. *IEEE Engineering in Medicine and Biology Confernce, Boston, MA*, 6619–6622.

DUNCAN, B. S. and OLSON, A. J. (1993). Approximation and characterization of molecular surfaces. *Biopolymers*, 33, 219–229.

EFRON, B., HASTIE, T. J., JOHNSTONE, I. M. and TIBSHIRANI, R. J. (2004). Least angle regression. *Annals of Statistics*, 32, 407–499.

FILZMOSER, P., GSCHWANDTNER, M. and TODROV, V. (2012). Review of sparse methods in regression and classification with application in chemometrics. *Journal of Chemometrics*, 26, 42–51.

FONOLLOSA, J., VERGARA, A. and HUERTA, R. (2012). Sensor failure mitigaton based on multiple kernels. *Sensors, 2012 IEEE*, 1–4.

FONOLLOSA, J., VERGARA, A. and HUERTA, R. (2013). Algorithm mitigation of sensor failure: Is sensor replacement really necessary? *Sensors and Actuators B: Chemical*, 183, 211–221.

FRALEY, C. and RAFTERY, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97, 611–631.

FRALEY, C. and RAFTERY, A. E. (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of classification*, 24, 155–181.

FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9, 432–441.

GARDNER, J. and BARTLETT, P. (1994). A brief history of electronic noses. *Sensors and Actuators B: Chemical*, 18, 211–220.

GEMAN, S. and GEMAN, D. (1984). Stochastic relaxation, Gibbs distributions and bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741.

GEORGE, E. I. and FOSTER, D. P. (2000). Calibration and empirical bayes variable selection. *Biometrika*, 87, 731–747.

GOLUB, G. H. and LOAN, C. F. V. (1996). *Matrix Computations*, The John Hopkins University Press. Troisième édition.

GONÇALVES, F., PRATES, M. and LACHOS, V. (2015). Robust bayesian model selection for heavy-tailed linear regression using finite mixtures. *arXiv preprint arXiv:1509.00331*.

GREEN, P. and SILVERMAN, B. (1994). *Non-parametric regression and generalized linear models*, London: Chapman and Hall.

GRENANDER, U. and MILLER, M. (1995). Computational anatomy: an emerging discipline. *Quarterly of Applied Mathematics*, LVI, 617–694.

GUTIERREZ-OSUNA, R. (2002). Pattern analysis for machine olfaction : a review. *IEEE Sensors Journal*, 2, 189–202.

HAAR, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69, 331–371.

HAMPEL, F., RONCHETTI, E., ROUSSEEUW, P. and STAHEL, W. (2011). *Robust Statistics: the approach based on influence functions*, John Wiley & Sons.

HANSEN, M. H. and YU, B. (2001). Model selection and principle of minimum description length. *Journal of American Statistical Association*, 96, 746–774.

HARTIGAN, J. A. (1990). Partition models. *Communications in Statistics-Theory and Methods*, 19, 2745–2756.

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2001). *The Elements of Statistical Learning*, Springer series in statistics Springer, Berlin.

HEARD, N. A., HOLMES, C. C. and STEPHENS, D. A. (2006). A quantitative study of gene regulation involved in the immune response of *Anopheline* mosquitoes: An application of Bayesian hierarchical clustering of curves. *Journal of the American Statistical Association*, 101, 18–29.

HIGHAM, N. J., COX, M. G. and HAMMARLING, S. (1990). Analysis of the Cholesky decomposition of a semi-definite matrix. *Oxford University Press*, 161–185.

HOFFMAN, I., SERNEELS, S., FILZMOSER, P. and CROUX, C. (2015). Sparse partial robust m-regression. *Chemometrics and Intelligent Laboratory Systems*, 149, 50–59.

HOLMES, M. P., GRAY, A. G. and ISBELL, C. L. (2007). Fast SVD for large-scale matrices. *Workshop on Efficient Machine Learning at Neural Information Processing Systems NIPS*, 58.

HONG, H. and PRESTON, B. (2012). Bayesian averaging, prediction and nonnested model selection. *Journal of Econometrics*, 167, 358–369.

HOSKULDSSON, A. (2005). Pls regression methods. *Journal of Chemometrics*, 2, 211–288.

HUBER, P. (1981). *Robust Statistics*, Wiley, New York.

HUBERT, M., ROUSSEEUW, P. J. and BRANDEN, K. V. (2005). Robpca: A new approach to robust principal component analysis. *Thechnometrics*, 47, 64–79.

HUBERT, M. and VAN DER VEEKEN, S. (2008). Outlier detection for skewed data. *Journal of Chemometrics*, 22, 235–246.

HYVARINEN, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10, 626 – 634.

JAIN, S. and NEAL, R. (2007). Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Analysis*, 2, 445–472.

JOHNSON, G., BUCK, T., SULLIVAN, D., RHODE, G. and RF, M. (2015). Joint modelling of cell and nuclear shape variation. *Molecular Biology of Cell*, 26, 40476–4056.

JOLLIFFE, I. (2002). *Principal Component Analysis*, Springer.

JOSSE, J., PAGÈS, J. and HUSSON, F. (2011). Multiple imputation for principal component analysis. *Advances in Data Analysis and Classifications*, 5, 231–246.

KAN, A. R. and TELGEN, J. (1981). The complexity of linear programming. *Statistica Neerlandica*, 2.

KEOGH, E., CHAKRABARTI, K., PAZZANI, M. and MEHROTRA, S. (2001). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3, 263–286.

KERMITI, M. and TOMIC, O. (2003). Independent component analysis applied on gas sensor array measurement data. *IEEE Sensors Journal*, 3, 218–228.

KHAIRY, K. and HOWARD, J. (2010). Minimum-energy vesicle and cell shapes calculated using spherical harmonics parameterization. *Soft Matter*, 7, 2138–2143.

KLASSEN, E., SRIVASTAVA, A., MIO, W. and JOSHI, S. (2004). Analysis of planar shape using geodesic paths on shape spaces. *IEEE Pattern Analysis and Machiner Intelligence*, 26, 372–383.

KRIM, H. and YEZZI, A., éditeurs (2006). *Statistics and Analysis of Shapes*. Birkhaüser.

LEE, A. M., BERNEY-LANG, M. A., LIAO, S., KANSO, E. and KUHN, P. (2012). A low-dimensional deformation model for cancer cells in flow. *Physics Fluids*, 24.

LEHMUSSOLA, A., SELINUMMI, J., RUUSUVUORI, P., NIEMISTO, A. and YLI-HARJA, O. (2005). Simulating fluorescent microscope images of cell populations. *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 1–4.

LI, G. and CHEN, Z. (1985). Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and monte carlo. *Journal of the American Statistical Association*, 80, 759–766.

LIANG, F., PAULO, R., MOLINA, G., CLYDE, M. A. and BERGER, J. O. (2008). Mixtures of g-priors for bayesian variable selection. *Journal of the American Statsitical Association*, 103, 410–423.

LIAO, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition*, 38, 1857–1874.

MANNE, R. (1987). Analysis of two partial least squares algorithms for multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 2, 187–197.

MATHAI, A. M. and PROVOST, S. B. (1992). *Quadratic Forms in Random Variables, Theory and Applications*. Marcel Dekker, New York.

MATLAB (2013a). Image processing toolbox R2013b. *The MathWorks Inc. , Natick, Massachusetts, United States*.

MATLAB (2013b). Signal processing toolbox R2013b. *The MathWorks Inc. , Natick, Massachusetts, United States*.

MAZUMDER, R. and HASTIE, T. (2012). The graphical lasso: new insights and alternatives. *Electronic Journal of Statistics*, 6, 2125–2149.

MCGINLEY, P. C. and INC, S. (2002). Standardized odor measurement practices for air quality testing. *Air and Waste Management Association Symposium on Air Quality Measurement Methods and Technology, San Francisco, CA*.

MEINSHAUSEN, N. and BUHLMANN, P. (2006). High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34, 1436–1462.

MILOVANOVIC, G. V., MITRINOVIC, D. S. and RASSIAS, T. M. (1994). *Topics in polynomials: extremal problems, inequalities, zeros*. World Scientific.

MIRSHAHI, M., PARTOVI NIA, V. and ADJENGUE, L. (2016). Statistical measurement validation with application to electronic nose technology. *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods*. 407–414.

MIRSHAHI, M., PARTOVI NIA, V. and ADJENGUE, L. (2017a). Automatic odor prediction for electronic nose. Submitted to Journal of Applied Statistics.

MIRSHAHI, M., PARTOVI NIA, V. and ADJENGUE, L. (2017b). *Pattern Recognition Applications and Methods*, Springer, chapitre An online data validation algorithim for electronic nose. No. G-2015-81. 104–120. Lecture Notes in Computer Science.

MIRSHAHI, M., PARTOVI NIA, V. and ASGHARIAN, M. (2017c). Bayesian information criterion for three-dimensional shape clustering. Manuscript.

MURPHY, K. P. (2012). *Machine Learning, A Probabilistic Perspective*, The MIT press.

NEWEY, W. and MCFADDEN, D. (1994). Large sample estimation and hypothesis testing. *Handbook of Econometrics*, 4, 2113–2245.

NISHI, R. (1984). Aysmptotic properties of citeria for selection of variables in multiple regression. *The Annals of Statistics*, 12, 758–765.

O'HAGAN, A. (1973). Bayes estimation of a convex quadratic. *Biometrika*, 60, 565–571.

O'HAGAN, A. and FORSTER, J. J. (2004). *Kendall's advanced theory of statistics, volume 2B: Bayesian inference*, vol. 2. Arnold.

OTTMANN, T., SCHUIERER, S. and SOUNDARALAKSHMI, S. (1995). Enumerating extreme points in higher dimensions. *STACS 95: 12th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science*, 900, 562–570.

PADILLA, M., PERERA, A., MONTOLIU, I., CHAUDRY, A., PERSAUD, K. and MARCO, S. (2010a). Drift compensation of gas sensor array data by orthogonal signal correction. *Journal of Chemometrics and Intelligent Labrotory System*, 100, 28–35.

PADILLA, M., PERERA, A., MONTOLIU, I., CHAUDRY, A., PERSAUD, K. and MARCO, S. (2010b). Fault detection, identification, and reconstrcution of faulty chemical gas sensors under drif conditions, using Principal Component Analysis and Multiscale-PCA. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 1–7.

PARDO, M., NIEDERJAUFNER, G., BENUSSI, G., COMINI, G. and FAGLIA, E. (2000). Data preprocessing enhances the classification of different brands of Espresso coffee with an electronic nose. *Sensors and Actuators B*, 69, 359–365.

PENG, T. and MURPHY, R. F. (2011). Image-derived, three-dimensional generative models of cellular organization. *Cytometry Part A*, 79, 383–391.

PERSAUD, K. and DODD, G. (1982). Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose. *Nature*, 299, 352–355.

PINCUS, Z. and THERIOT, J. A. (2007). Comparison of quantitative methods for cell-shape analysis. *Journal of Microscopy*, 227, 140–156.

PRENDERGAST, L. (2008). A note on sensitivity of principal component subspaces and the efficient detection of influential observations in high dimensions. *Electronic Journal of Statistics*, 2, 454–467.

PRIESTLEY, M. (1982). *Spectral analysis and time series*. No. v. 1-2 Probability and mathematical statistics. Academic Press.

QIN, S. J. (1997). Neural networks for intelligent sensors and control—practical issues and some solutions. *Neural Systems for Control*, 213–234.

RAO, C. R. and WU, Y. (1989). A strongly consistent procedure for model selection in a regression problem. *Biometrika*, 76, 369–374.

ROUSSEEUW, P. J. and DRIESSEN, K. V. (1999). A fast algorithm for the minumum covariance determinant estimator. *Technometrics*, 41, 212–223.

SCHOENBERG, I. (1973). Cardinal spline interpolation. *Applied Mathematics*.

SCHOENBERG, I. J. (1946). Contributions to the problem of approximation of equidistant data by analytic functions. part a. on the problem of smoothing or graduation. a first class of analytic approximation formulae. *Quarterly of Applied Mathematics*, 4, 45–99.

SCHUMAKER, L. (2007). *Spline Functions: Basic Theory*. Cambridge University Press.

SCOTT, G. L. (1987). The alternative snake–and other animals. *Proceedings, 3rd Alvey Vision Conference, Cambridge*, 341–347.

SEN, P. and SINGER, J. (1994). *Large Sample Methods in Statistics: An Introduction with Applications*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis.

SERNEELS, S., CROUX, C., FILZMOSER, P. and ESPEN, P. J. V. (2005). Partial robust m-regression. *Chemometrics and Intelligent Laboratory Systems*, 79, 55–64.

SHAO, J. (1997). An asymptotic theory for linear model selection. *Statistica Sinica*, 221–264.

SHAO, X., LI, H., WANG, N. and ZHANG, Q. (2015). Comparison of different classification methods for analyzing electronic nose data to characterize sesame oils and blends. *Journal of Sensors*, 15, 26726–26742.

SHIBATA, R. (1981). An optimal selection of regression variables. *Biometrika*, 68, 45–54.

SINGHAL, A. and SEBORG, D. E. (2005). Clustering multivariate time-series data. *Journal of chemometrics*, 19, 427–438.

SMITH, J., ANDERSON, P. and LIVERANI, S. (2008). Separation measures and the geometry of Bayes factor selection for classification. *Journal of the Royal Statistical Society, Series B*, 70, 957–980.

SRIVASTAVA, A., JOSHI, S., KAZISKA, D. and WILSON, D. (2006). *Planar Shape Analysis and Its Applications in Image-Based Inferences*, Springer US, Boston, MA. 189–203.

STAHEL, W. A. (1981). Robust estimation: Infinitesimal optimality and covariance matrix estimators. *Ph.D. Thesis, ETH, Zurich*.

STEFFENSEN, I. (1950). *Interpolation*. Chelsea, New York, seconde édition.

THEIL, H. (1963). On the use of incomplete prior information in regression analysis. *Journal of the American Statistical Association*, 58, 401–414.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58, 267–288.

TOLSTOV, G. P. (2012). *Fourier Series*. Dover Publcations.

TRYON, R. C. and BAILEY, D. E. (1970). *Cluster Analysis*. McGraw-Hill Inc., US, New York.

WAHBA, G. (1990). *Spline models for observational data*, The Society for Industrial and Applied Mathematics.

WARD, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58, 236–244.

WILSON, D. M., DUNMAN, K., ROPPEL, T. and KALIM, R. (2000). Rank extraction in tin-oxide sensor arrays. *Sensors and Actuators B: Chemical*, 62, 199–210.

WOLD, H. (1966). Estimation of principal components and related models by iterative least squares. multivariate analysis. edited by: Krishnaiaah pr. 1966.

YAN, J., GUO, X., DUAN, S., JIA, P., WANG, L., PENG, C. and ZHANG, S. (2015). Electronic nose feature feature extraction methods: A review. *Journal of Sensors*, 15, 27804–27831.

YEUNG, K. Y., FRALEY, C., MURUA, A., RAFTERY, A. E. and RUZZO, W. L. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17, 977–987.

YUAN, M. and LIN, Y. (2007). Model selection and estimation in the guassian graphical model. *Biometrika*, 94, 19–35.

YUEDONG, W. (2011). *Smoothing Splines: Methods and Applications*, Taylor and Francis.

ZELLNER, A. (1983). Application of Bayesian analysis with g-prior distributions. *The Statistician*, 32, 23–34.

ZELLNER, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti. North-Holland Elsevier*, 233–243.

ZHAO, T. and MURPHY, R. F. (2007). Automated learning of generative models for subcellular location: building blocks for system biology. *Cytometry*, Part A, 978–990.

ZITOVÀ, B. and FLUSSER, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, 21, 977–1000.

ZOU, H., HASTIE, T. and TIBSHIRANI, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15, 265–286.

ZUPPA, M., DISTANTE, C., PERSAUD, K. C. and SICILIANO, P. (2007). Recovery of drifting sensor responses by means of DWT analysis. *Journal of Sensors and Actuators*, 120, 411–416.