BAYESIAN NONPARAMETRIC MIXTURE MODELING

by

Guoliang Cao

Institute of Statistics and Decision Sciences
Duke University

Date: _____
Approved:

_____
Mike West, Advisor

_____
Donald Burdick

_____
Michael Lavine

_____
Peter Müller

_____
Dennis A. Turner

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Institute of Statistics and Decision Sciences
in the Graduate School of
Duke University

1993

ABSTRACT

(Statistics)

BAYESIAN NONPARAMETRIC MIXTURE MODELING

by

Guoliang Cao

Institute of Statistics and Decision Sciences
Duke University

Date: _____
Approved:

_____
Mike West, Advisor

_____
Donald Burdick

_____
Michael Lavine

_____
Peter Müller

_____
Dennis A. Turner

An abstract of a dissertation submitted in partial
fulfillment of the requirements for the degree
of Doctor of Philosophy in the Institute of Statistics and Decision Sciences
in the Graduate School of
Duke University

1993

Abstract

This dissertation explores a Bayesian nonparametric approach to mixture modeling and the use of the Gibbs sampling scheme to approximate posterior estimates. The predictive distribution is modeled as a mixture of normal distributions by using a Dirichlet process prior for the unknown means and variances. The definition and some properties of mixtures of Dirichlet processes are reviewed. Analytically evaluating the predictive distribution is very tedious and difficult in this case. An approximation based on Monte Carlo integration is proposed. The Gibbs sampling method based on unknown means and variances is used and convergence based on the configuration space is proved. Different models are compared based on simulated data.

Order statistics are used to solve the problem of identifiability. Recurrence relations are derived to calculate the distribution of the order statistics in the case of independent non-identically distributed random variables.

The practicality of the nonparametric Bayesian analysis for mixture modeling is showed. Various methodological and computational aspects are developed. Mixture analysis for grouped data is discussed. Some comparisons between Bayesian methods and classical methods are described. A Bayesian analysis of mixtures of mixtures is introduced and illustrated in the context of

neurological response analysis.

Acknowledgements

I would like to express my loving gratitude toward my wife Jiang Qian for her understanding and unwavering support. She was always there in the bad and the good times making everything that much nicer. I would like to thank my parents, Simei Chen and Fengming Cao: first for being my parents, and for the environment they created for me as a child; and second for their encouragement during my study. Without such support this effort would hardly have been likely.

I would like to express my sincere appreciation to Professor Mike West. Apart from being a highly competent advisor, he was always available for discussions and often had a good idea bubbling out of his mind. Certainly his patience and understanding far exceed reasonable expectations. His inspiration and guidance are present in every aspect of this dissertation.

I would like to thank the faculty, the staff, and the graduate students of both the Institute of Statistics and Decision Sciences and the Department of Neurosurgery, Duke University Medical Center for providing a good environment for the development of research, in particular Dr. Dennis Turner, Professors Donald Burdick, Michael Lavine, Peter Müller and Donald Berry. From these communities I have been instructed, inspired, and befriended.

Finally, many thanks to all my friends and relatives for showing goodwill towards the success of my work even with little or no knowledge of what it was about.

Contents

Abstract i

Acknowledgements iii

List of Figures vii

List of Tables viii

1 Introduction 1
- 1.1 Applications of Mixture Distributions . . . 2
- 1.2 Classical Analysis of Mixture Distributions . . . 3
- 1.3 Bayesian Analysis of Mixture Distributions . . . 7
- 1.4 Outline of the Dissertation . . . 10

2 Mixtures of Dirichlet Processes 12
- 2.1 Dirichlet Processes . . . 12
- 2.2 Properties of Samples from Dirichlet Processes . . . 15
- 2.3 Normal Mixtures of Dirichlet Processes . . . 18
- 2.4 Computation of Mixtures of Dirichlet Processes . . . 20
  - 2.4.1 Gibbs Sampling . . . 20
  - 2.4.2 Convergence of the Gibbs Sampling . . . 22

3 Development of Mixtures of Dirichlet Processes 26
- 3.1 Gibbs Sampling Based on Configurations . . . 26
  - 3.1.1 Algorithms . . . 28
  - 3.1.2 An Example . . . 34
- 3.2 Gibbs Sampling Based on a Non-conjugate Base Prior . . . 39
- 3.3 Convergence Issues . . . 42
- 3.4 Proofs of Theorems . . . 50

4 Order Statistics **53**
4.1 Introduction 53
4.2 Recurrence Relations 54
4.3 Examples 59

5 Neurophysiological Case Study **64**
5.1 Background 64
5.2 Experiments and Data Sources 66
5.3 Mixture Modeling of Signal Data 68
5.4 Discussion 79

6 Bayesian Density Estimation for Grouped Data **85**
6.1 Introduction 85
6.2 Grouped Data Models 86
6.3 Normal Mixture Modeling 87
6.4 Classical Methods vs. Bayesian Methods 89
6.5 Discussion 92
6.6 Appendix 94
6.6.1 Proof of Equation (6.8) 94
6.6.2 Proof of Equation (6.10) 95

7 Bayesian Analysis of Mixtures of Mixtures **96**
7.1 Introduction 96
7.2 Mixtures of Mixtures Modeling 97
7.3 Computational Methods 98
7.4 Simulations 102
7.5 Application to Neurophysiological Data 106
7.6 Discussion 111

8 Conclusions and Further Work **112**

Appendix **115**

1 C Program — An Example **116**

Bibliography **140**

Biography **147**

List of Figures

3.1 Predictive Density Functions for Simulation Comparisons 44
3.2 Convergence of Gibbs sampler (I) 47
3.3 Convergence of Gibbs sampler (II) 48
3.4 Convergence of Gibbs sampler (III) 49

4.1 Cumulative Distribution Functions 62
4.2 Probability Density Functions 63

5.1 Neural Noise and Signal Histograms 73
5.2 Summaries of Posterior and Predictive Distributions 74
5.3 Summaries of Posterior Distributions for Hyperparameters . . . 75
5.4 Predictive Mixture Deconvolution 76
5.5 Posteriors for Ordered Component Means 77
5.6 Comparisons for Predictive Density Functions 80
5.7 Comparisons for Posterior Distributions of k 81
5.8 Comparisons for Predictive Mixture Deconvolutions 82
5.9 Comparisons for Posterior Distributions of Hyperparameters . . 83

6.1 Simulation Analysis for Grouped Data 90
6.2 Classical Methods vs. Bayesian Methods for Grouped Data . . . 93

7.1 Simulation Analysis 105
7.2 Histograms of Mixtures of Mixtures 109
7.3 Analysis of Mixtures of Mixtures 110

List of Tables

3.1 Performance of Different Algorithms — Monte Carlo Size of 500 36
3.2 Performance of Different Algorithms — Monte Carlo Size of 5000 36
3.3 Probability Distributions of k for Simulation Comparison 43

5.1 Signal Data 72
5.2 Quantiles for $\mu^*_{(j)}$ 78
5.3 Quantiles for $\delta_j = \mu^*_{(j)} - \mu^*_{(j-1)}$ 78

6.1 Binned Data 88

7.1 Posterior Distributions of the Number of Components 104
7.2 Distributions of k_0 and k_1 108

Chapter 1

Introduction

Suppose we have observed data $y_1, \ldots, y_n$, such that given $(\mu_1, v_1), \ldots, (\mu_n, v_n)$, the y_i's are independent, and the y_i has a normal distribution with mean μ_i and variance v_i. Let $\pi_i = (\mu_i, v_i)$. By using a Dirichlet process prior for π_i's, we can derive a Bayesian nonparametric mixture model.

Before we discuss the Dirichlet process prior and the nonparametric Bayes' analysis, let us first consider some typical ways to deal with a mixture model. The most common way to consider a mixture model is as follows:

Suppose a random variable y comes from any one of k components, each of them having a different probability density function. If y comes from i-th component with probability w_i and probability density function $f(y|\pi_i)$, then the probability density function (pdf) of an observation y can be represented in the mixture form

$$f(y|\phi) = \sum_{i=1}^{k} w_i f(y|\pi_i), \tag{1.1}$$

where $\phi = (k, w_1, \ldots, w_k, \pi_1, \ldots, \pi_k)$, k is the number of components, the w_i's are the mixture weights which must be non-negative and sum to one

and the π_i's denote the vector of all unknown parameters associated with i-th component. For example, in the normal case, we can think of π_i as the unknown mean μ_i and the unknown variance v_i.

In some applications it is desired to categorize or classify each observation into one of the k populations. Under a Bayesian approach this can be done by computing a classification probability, the probability that an observation comes from a certain population. Alternatively one may be interested in the elements of ϕ, which include both the mixing weights $w_1, \ldots, w_k$, and the density parameters $\pi_1, \ldots, \pi_k$. The functional form of each distribution is usually assumed to be known. Often the $f(\cdot|\pi_i)$'s will have the same functional form, differing only in the parameter π_i. In some cases the number of components k may not be known, and is estimated from the data, as in cluster analysis. The emphasis of cluster analysis is mainly on dividing observations into clusters and determining the number of distinct clusters rather than on estimation of mixing weights and population density parameters.

1.1 Applications of Mixture Distributions

Mixture modeling arises in many fields, including medicine, biology and economics. In most cases, the mixing weights and density parameters are unknown. Bhattacharya (1967) uses the distribution of fish lengths in determining which age groups are present. A collection of fish may include several different age groups, but determining a fish's age is difficult. Each age group has a different distribution of lengths, so that length provides some information about age. Since the ages are unknown, the distribution of a fish's length y is a mixture of the form (1.1), where k is the number of age groups, w_i is the

proportion of fish of the i-th age, and $f(y|\pi_i)$ is the distribution of length in the i-th group. Presumably the main goal here is to estimate the proportions of the different ages.

Another important application is in the neurophysiological field, as is illustrated in this dissertation. The trial-to-trial amplitude fluctuation of postsynaptic potentials at neural junctions contain information which can allow different events underlying synaptic transmission to be separated (Kullmann, 1989). Briefly, quanta of neurotransmitter are released from presynaptic release sites. Each quantum produces a postsynaptic membrane potential change. The quantal potentials arising from different sites then generally sum linearly. Assuming that the background noise in the cell is approximately normal, the observed excitatory postsynaptic potential peak amplitude distribution is treated as a mixture of several incompletely sampled normal distributions. The problem becomes one of resolving the parameters of the mixture.

1.2 Classical Analysis of Mixture Distributions

Classical estimation techniques are described at length in Titterington, Smith and Makov (1985), Everitt and Hand (1981) and McLachlan and Basford (1987). Graphical techniques, which focus on properties such as modality of a histogram of the raw data, are among the earliest approaches. However, the number of modes of a histogram may depend upon the number of bins and may not equal the number of components in the mixture for any reasonable choice of bins. Therefore, histogram approaches to estimating k may be misleading. Recently, much attention has been given to likelihood-based approaches, us-

ing maximum likelihood where appropriate. This approach would appear in general to be superior to other classical methods of fitting mixture models. But some difficulties still remain. For example, there is no analytical solution for the maximum likelihood estimation, so that iterative techniques such as the expectation-maximization (EM) algorithm are required. In some cases the likelihood is unbounded on the boundary of the parameter space. For example, in a mixture of univariate normal distributions, $w_1 N(y|\mu_1, v_1) + w_2 N(y|\mu_2, v_2)$, the likelihood goes to infinity if one of the μ_i's is taken to be one of the observations and v_i is set to zero. Clearly this is not a desirable solution. However, this difficulty can be avoided by finding a stationary maximum, i.e., a maximum where the first derivative is zero, in the interior of the parameter space. See also Kiefer (1978). Redner and Walker (1984) and Peters and Walker (1978) give, for the class of identifiable mixtures, the regularity conditions which must hold in order to have a sequence of roots of the likelihood equation with the properties of consistency, efficiency and asymptotic normality.

Identifiability is another difficult issue for mixture modeling. In order to be able to estimate all the parameters from the data, it is necessary that they should be identifiable. Identifiability is usually assumed both for parameter estimation and for asymptotic theory. In general, a parametric family of distribution $\mathfrak{S}$ is said to be identifiable if different parameter values determine distinct distributions in the family. This can be interpreted as in the case where $f(x|\phi)$ defines a class of finite mixtures (1.1). A class of finite mixture $\mathfrak{S}$ is said to be identifiable for ϕ if for two such mixtures,

$$f(y|\phi) = \sum_{i=1}^{k} w_i f(y|\pi_i) \tag{1.2}$$

and

$$f(y|\phi') = \sum_{i=1}^{k'} w_i' f(y|\pi_i'), \tag{1.3}$$

$f(y|\phi) = f(y|\phi')$ if and only if $k = k'$ and we can permute the component labels so that

$$w_i = w_i' \quad \text{and} \quad \pi_i = \pi_i' \ (i = 1, \ldots, k). \tag{1.4}$$

Yakowitz and Spragins (1968) give necessary and sufficient conditions for identifiability. Titterington, Smith and Makov (1985) point out that most finite mixtures of continuous distributions are identifiable. They also give a lucid account of the concept of identifiability for mixtures, including several examples and a survey of the literature on this topic. One difficulty with mixtures is that if $f(y|\pi_i)$ and $f(y|\pi_j)$ belong to the same parametric family, then $f(y|\phi)$ will have the same value when the component labels i and j are exchanged in ϕ. That is, although this class of mixtures may be identifiable, ϕ is not. Indeed, if all the $f(y|\pi_i)$ belong to the same parametric family, then $f(y|\phi)$ is invariant under the $k!$ permutations of the component labels in ϕ. In this case, if there is one local maximum, then the likelihood will have at least $k!$ local maxima of the same value. However, the difficulty due to the interchanging of component labels can be easily overcome by the imposition of an appropriate constraint on ϕ. For example, if $f(y|\pi_i)$ is a normal density function $N(y|\mu_i, v_i)$ and $\pi_i = (\mu_i,\ v_i)$ with known v_i, for $i = 1, \ldots, k$, we can impose constraints such as $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_k$.

The EM algorithm is convenient for the iterative computational solutions of the likelihood equation and is described in detail in Dempster, Laird and Rubin (1977). The EM algorithm is applied to the mixture model when the

data can be viewed as incomplete. By treating the classifications of the observations as missing data, we can easily program it iteratively in two steps, E (for expectation) and M (for maximization). The E step is effected simply by replacing the 'missing data' by its expectation conditional on data. One nice feature of the EM algorithm is that the solution to the M step often exists in closed form, as will be demonstrated for mixtures of normals in Chapter 6. If the classifications are known, the maximization of the likelihood is usually straightforward, which makes the M step of the EM algorithm fairly simple. Unfortunately, the EM algorithm often converges quite slowly and does not yield directly the observed information matrix which may be needed for the estimation of the standard errors of some parameters. However, Louis (1982) devises a procedure for extracting the observed information matrix when using the EM algorithm, as well as developing a method for speeding up its convergence. In addition, the EM algorithm (Dempster, Laird and Rubin, 1977) is a robust method for obtaining maximum likelihood estimates of the parameters of an incompletely sampled distribution.

As commented before, convergence with the EM algorithm is very slow, and the situation will be exacerbated by a poor choice of starting values. Indeed, in some cases where the likelihood is unbounded on the edge of the parameter space, the sequence of estimates generated by the EM algorithm will diverge if the initial values are chosen too close to the boundary. Another issue with mixture models is that the likelihood equation will usually have multiple roots, so the EM algorithm should be applied to a wide choice of starting values in a search for all local maxima.

Other possible alternatives to the EM algorithm include the Newton-Raphson method and the method of scoring. Both of them are more complicated, since

they require matrix inversion at each iteration. Convergence of the Newton-Raphson method can be quite rapid, in particular, when the log likelihood is well approximated by a quadratic function. As a Newton iterative scheme requires the inversion of the matrix of second derivatives of the log likelihood, it automatically provides an estimate of the covariance matrix of the parameters, although this inversion may not be computationally convenient at each step if the dimension of the parameter space is high. Redner and Walker (1984) discuss the performance of the EM algorithm for finite mixture models. The EM algorithm tends to perform somewhat better than the other techniques when the components in the mixture are not well separated. Because of the slow convergence of the EM algorithm, we can work instead with a hybrid method, starting out with the EM algorithm and finishing with another technique such as the Newton-Raphson method.

1.3 Bayesian Analysis of Mixture Distributions

Two kinds of Bayesian analyses of mixtures, parametric and nonparametric, have been developed. Parametric Bayesian analysis incorporates prior information about interesting parameters in the inference. Sources of prior information include beliefs about parameters before any data are observed. A noninformative prior, which contains no information about the parameters, is widely used. In some cases, a training sample can also be thought of as part of the prior information obtained before analyzing the unclassified data. Once the prior information and data are available, the posterior distribution can be obtained

by Bayes' theorem.

$$f(\phi|Y) \propto f(\phi)\ f(Y|\phi), \tag{1.5}$$

where $f(\phi)$ is the prior density function of ϕ, and $f(Y|\phi) = L(\phi|Y)$ is the likelihood function of ϕ. In practice, the proportionality constant can be ignored. However, due to the form of the likelihood, the computation of posterior distributions is still very difficult. Given k populations, if we take the form of equation (1.1), the likelihood after n unclassified observations $Y = (y_1, \ldots, y_n)$ has the form

$$L(\phi|Y) = \prod_{i=1}^{n} \sum_{j=1}^{k} w_j f(y_i|\pi_j), \tag{1.6}$$

which is a mixture of k^n terms. This leads to computational difficulties even for a moderate sample size, since the number of terms grows exponentially in n. For example, if $k = 2$ and $n = 100$, then the number of terms reaches 1.268×10^{30}. The number of terms in the posterior will increase exponentially in n unless one chooses a point mass prior. Moreover, because the likelihood is a product of sums rather than a single term, in general it is impossible to factorize the likelihood in terms of the unknown parameters; therefore, no sufficient statistics of fixed dimension exist. Techniques used in maximum likelihood estimation can be applied to find the posterior modes, but not other posterior quantities, such as posterior means, marginal densities and predictive distributions. The computation of those quantities involves numerical integration in several dimensions. The storage requirements for the posterior increase very rapidly with the sample size.

Several different approximations to the posterior have been proposed to solve such mixture models. Most of the techniques work with the sequential form of calculating the posterior. After each observation, a new approximation

to the posterior is calculated by collapsing the mixture obtained by adding the new observation. The number of terms in the approximation thus remains fixed instead of growing with the sample size. This avoids many computational problems associated with an exact analysis. This collapsing, however, means that information is lost at each step. A non-sequential method, which makes no simplifications until all the data have been collected, is potentially more accurate than a sequential approximation. Of course, the non-sequential method is more complicated to work with, since making fewer simplifications along the way entails greater complexity in collapsing at the last step. Various numerical techniques, used for complicated single-component distributions, can be applied in the mixture setting as well. These methods, which do not take into account the mixture structure, are used when integrations such as the calculation of the posterior mean or marginal density are analytically intractable. Numerical approximations provide one such way of approximating posterior quantities. Naylor and Smith (1982) propose an iterative numerical integration procedure using Gaussian quadrature, which simplifies calculations involving integration of the posterior. Such numerical integration procedures , however, can be computationally intensive. Tierney, Kass, and Kadane's (1986) modification of the Laplace method, a non-sequential technique, is found to provide an accurate yet easily computed approximation in the context of finite mixtures to posterior expectations of functions of the unknown parameters, such as posterior moments and marginal densities. Crawford (1988) explores exact and approximate Bayesian analyses of finite mixture distributions. Under some conditions, finite mixtures of several types of densities are shown to lead to asymptotically normal posteriors.

Recently, an iterative resampling approach, called Gibbs sampling, has been

proposed to solve the mixture model. Lavine and West (1992) use the Gibbs sampling approach to calculate the traditional normal mixture models. Such a method is applied to the mixture model by introducing the classification variables which identify data points with specific components. Conditional on classification variables, y_i has a simpler density form. This enables us to analyze the mixture model easily.

On the other hand, Bayesian nonparametric mixture modeling has been developed recently. Suppose data y_i $(i = 1, \ldots, n)$, are distributed as $f(y_i|\pi_i)$, independently over i. The mixture structure simply imposes the constraint that, for some positive integer k, there exist k distinct numbers $\pi^* = (\pi_1^*, \ldots, \pi_k^*)$ such that, for each $i = 1, \ldots, n$, $\pi_i = \pi_j^*$ for some $j = 1, \ldots, k$. One way of generating such a mixture is to use a Dirichlet process for the prior distribution of $\pi = (\pi_1, \ldots, \pi_n)$. Full technical details are reviewed in Chapter 2. An important result of the Dirichlet process is that the probability of $\pi_i = \pi_j$ for $i \neq j$ is positive. Such Bayesian analyses have been of limited use due to computational difficulties. However, these difficulties can now be overcome using Gibbs sampling techniques, as described in Escobar (1991), Escobar and West (1991) and later in this dissertation. This permits the extension of standard Bayesian "parametric" analyses to natural Bayesian "nonparametric" analyses, by linking the Gibbs sampling computations for Dirichlet process priors to those for the standard models.

1.4 Outline of the Dissertation

In Chapter 2, properties of mixtures of Dirichlet processes are reviewed. Standard Gibbs sampling techniques are presented. Convergence issues of such

sampling schemes are discussed. In Chapter 3, the Gibbs sampling techniques are applied to the configuration space rather than the parameter space. Different models are compared based on simulated data. The convergence of the Gibbs sampler is proved. In Chapter 4, recurrence relationships among the distribution functions of order statistics of independent, but not identically distributed random quantities are derived in order to solve the problem of identifiability. In Chapter 5, a case study in the neurophysiological context is demonstrated. Predictive density estimates and posterior density functions of some hyperparameters are given. In Chapter 6, mixture analysis for grouped data is discussed. Some comparisons between Bayesian methods and classical methods are exhibited there. In Chapter 7, a Bayesian analysis of mixtures of mixtures is introduced. This is illustrated in the context of neurological response analysis where the issue of non-normality of the noise terms occurs. In Chapter 8, areas of future research related to Bayesian nonparametric analysis of mixtures are outlined.

Chapter 2

Mixtures of Dirichlet Processes

2.1 Dirichlet Processes

Before we define Dirichlet processes, let us first define the class of Dirichlet distributions. The vector $(y_1, \ldots, y_n)$ has a Dirichlet distribution with parameter $(\alpha_1, \ldots, \alpha_n)$, if its density function is

$$f(y_1, \ldots, y_n | \alpha_1, \ldots, \alpha_n) = \frac{\Gamma(\alpha_1 + \ldots + \alpha_n)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_n)} \prod_{i=1}^{n} y_i^{\alpha_i - 1} \quad \text{for } \mathrm{y_i} \geq 0 \ \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{y_i} = 1, \tag{2.1}$$

where $\alpha_i > 0$ for $i = 1, \ldots, n$. The class of Dirichlet distributions is known to Bayesians as the conjugate prior for the parameters of a multinomial distribution. The main properties of the class of Dirichlet distributions are

1. *If* $(y_1, \ldots, y_n)$ *has a Dirichlet distribution with parameter* $(\alpha_1, \ldots, \alpha_n)$ *and* $r_1, \ldots, r_n$ *are integers such that* $0 < r_1 < \ldots < r_l = n$, *then* $(\sum_{i=1}^{r_1} y_i, \sum_{i=r_1+1}^{r_2} y_i \ldots, \sum_{i=r_{l-1}+1}^{r_l} y_i)$ *has a Dirichlet distribution with parameter* $(\sum_{i=1}^{r_1} \alpha_i, \sum_{i=r_1+1}^{r_2} \alpha_i, \ldots, \sum_{i=r_{l-1}+1}^{r_l} \alpha_i)$.

2. *If* $(y_1, \ldots, y_n)$ *has a Dirichlet distribution with parameter* $(\alpha_1, \ldots, \alpha_n)$*, then*

$$\begin{aligned} Ey_i &= \alpha_i/\alpha, \\ Ey_i^2 &= \alpha_i(\alpha_i+1)/(\alpha(\alpha+1)), \\ Ey_iy_j &= \alpha_i\alpha_j/(\alpha(\alpha+1)) \quad \textit{if } i \neq j, \end{aligned}$$

where $\alpha = \sum_1^n \alpha_i$.

3. *If the prior distribution of* $(y_1, \ldots, y_n)$ *is a Dirichlet distribution with parameter* $(\alpha_1, \ldots, \alpha_n)$ *and if*

$$P(X = j | y_1, \ldots, y_n) = y_j \quad \textit{for } j = 1, \ldots, n,$$

then the posterior distribution of $(y_1, \ldots, y_n)$ *given* $X = j$ *is a Dirichlet distribution with parameter* $(\alpha_1', \ldots, \alpha_n')$ *where*

$$\alpha_i' = \begin{cases} \alpha_i & \textit{if } i \neq j \\ \alpha_i + 1 & \textit{if } i = j. \end{cases}$$

Now we can move on Dirichlet processes. Let $\mathfrak{X}$ be a set and $\mathfrak{A}$ a σ-field of subsets of $\mathfrak{X}$, and G be a finite nonnull measure on $(\mathfrak{X}, \mathfrak{A})$. Then a stochastic process P indexed by elements A of $\mathfrak{A}$, is said to be a Dirichlet process on $(\mathfrak{X}, \mathfrak{A})$ with parameter G if for any measurable partition $(A_1, \ldots, A_k)$ of $\mathfrak{X}$, the random vector $(P(A_1), \ldots, P(A_k))$ has a Dirichlet distribution with parameter $(G(A_1), \ldots, G(A_k))$. The process P may be considered as a random probability measure on $(\mathfrak{X}, \mathfrak{A})$. Ferguson (1973) gives the following basic result:

Theorem 2.1 *(Ferguson 1973) If* P *is a Dirichlet process with parameter* G*, and if, given* P*,* $\pi_1, \ldots, \pi_n$ *is a sample from* P*, then the posterior distribution of* P *given* $\pi_1, \ldots, \pi_n$ *is a Dirichlet process with parameter* $G + \sum_{j=1}^n \delta_{\pi_j}(P)$*, where* $\delta_{\pi_j}(P)$ *represents the distribution giving mass one to the point* π_j.

Throughout this dissertation, we shall use $\alpha = G(\mathfrak{X})$ to represent the total mass of G, and $G_0 = G/\alpha$ to be the prior guess at P. We denote $D(\alpha, G_0)$ as a Dirichlet process with parameters α and G_0. The latter phrase stems from the fact that from the definition, $P(A)$ has a Beta distribution, $Be(G(A), \alpha - G(A))$, so that $E(P(A)) = G(A)/\alpha = G_0(A)$. In particular, the posterior guess at P given a sample from P is, according to Theorem 2.1, $G_n = w_n G_0 + (1 - w_n)\widehat{F}$, where $\widehat{F} = \frac{1}{n}\sum_{j=1}^{n} \delta_{\pi_j}(P)$ is the empirical cumulative distribution function and $w_n = \alpha/(\alpha + n)$. As a consequence, suppose that it is required to estimate the mean $\mu = \int x dP(x)$ of an unknown distribution P on the real line based on a sample $\pi_1, \ldots, \pi_n$, with a prior $P \in D(\alpha, G_0)$. Under the squared error loss function and assuming G_0 has finite first moment μ_0, then μ is finite almost surely and

$$E(\mu|\pi_1, \ldots, \pi_n) = w_n \mu_0 + (1 - w_n)\overline{\pi}_n,$$

$\overline{\pi}_n$ is the sample mean.

Blackwell (1973) shows if $P \in D(\alpha, G_0)$, then P is almost surely discrete and does not have a probability density function.

Other alternative definitions of the Dirichlet process and a number of simple applications such as estimating the distribution function or the median , mean and variance, are described by Ferguson (1973). Antoniak (1974) discusses a number of Bayesian statistical problems with Dirichlet process priors. The problems include empirical Bayes, bio-assay, regression, discrimination, and classification problems. A number of other similar applications have appeared since that time. We will not discuss them here. The interested reader can refer to the papers by Ferguson (1973), Antoniak (1974), Yamato (1975), Berry and Christensen (1979), West and Cao (1992), West and Turner (1992) and Turner

and West (1993).

2.2 Properties of Samples from Dirichlet Processes

Let P be a Dirichlet process on a standard Borel space $(\mathfrak{X}, \mathfrak{A})$ with parameter α and G_0. We assume G_0 is nonatomic and α and G_0 are fixed. If π_1 and π_2 were a sample from G_0 in the usual independent identically distributed sense, we would expect that the probability of $\pi_1 = \pi_2$ is zero. However, for such a Dirichlet process P , the probability of $\pi_1 = \pi_2$ is equal to $1/(\alpha + 1)$ This is because the conditional distribution of P given π_1 is a Dirichlet process with parameter $\alpha + 1$ and $G_0 + \delta_{\pi_1}(P)$, which is already atomic with an atom of measure 1 at π_1. If $\pi_1, \pi_2, \ldots, \pi_n$ is a sample of size n from P, Ferguson (1973) and Antoniak (1974) discover the following results:

$$\begin{aligned}
\pi_1 &\sim G_0, \\
(\pi_2|\pi_1) &\sim \frac{\alpha}{\alpha+1} G_0 + \frac{1}{\alpha+1}\delta_{\pi_1}(\pi_2), \\
&\vdots \\
(\pi_n|\pi_1, \ldots, \pi_{n-1}) &\sim \frac{\alpha}{\alpha+n-1} G_0 + \frac{1}{\alpha+n-1}\sum_{j=1}^{n-1}\delta_{\pi_j}(\pi_n),
\end{aligned}$$

where $\delta_{\pi_j}(\pi_n)$ denotes a unit point mass at $\pi_n = \pi_j$. The joint distribution of $\pi = (\pi_1, \ldots, \pi_n)$ is

$$f(\pi) = \prod_{i=1}^{n} \frac{\alpha G_0 + \sum_{j=1}^{i-1}\delta_{\pi_j}(\pi_i)}{\alpha + i - 1} \tag{2.2}$$

and the full conditional distribution of $(\pi_i|\pi_1,\ldots,\pi_{i-1},\pi_{i+1},\ldots,\pi_n)$ for $(i = 1,\ldots,n)$ is:

$$f(\pi_i|\pi_1,\ldots,\pi_{i-1},\pi_{i+1},\ldots,\pi_n) = \frac{\alpha}{\alpha+n-1}G_0 + \frac{1}{\alpha+n-1}\sum_{j\neq i}\delta_{\pi_j}(\pi_i). \quad (2.3)$$

Given $\pi = (\pi_1,\ldots,\pi_n)$, a sample of size n from P, the next sample π_{n+1} has the distribution

$$(\pi_{n+1}|\pi) \sim \frac{\alpha}{\alpha+n}G_0(\pi_{n+1}) + \frac{1}{\alpha+n}\sum_{i=1}^{n}\delta_{\pi_i}(\pi_{n+1}). \quad (2.4)$$

Thus, given π, the next case π_{n+1} represents a new, distinct value with probability $\alpha/(\alpha+n)$, and is otherwise drawn uniformally from among the first n values $\pi_1,\ldots,\pi_n$. The new value can be drawn from the distribution G_0.

For the relationship between a generalized Polya urn scheme and the Dirichlet process prior, see Blackwell and MacQueen (1973) and Ferguson (1973). Polya trees, an interesting generalization of the Dirichlet process, are discussed by Ferguson (1974), Mauldin, Sudderth and Williams (1991) and Lavine (1991), under a different name.

Let K denote the number of distinct elements of π. Then the distribution of K can be found (Antoniak, 1974) as follows:

$$\Pr(K = k) \propto a_k\alpha^k \qquad k = 1,\ldots,n, \quad (2.5)$$

subject to $\sum_{k=1}^{n}\Pr(K = k) = 1$, where the a_k's are the absolute values of Stirling numbers of the first kind (Abramowitz and Stegun, 1965). In particular, the expectation of K is about $\alpha\ log((\alpha+n)/\alpha)$. The mean of K is an increasing function of α. Recently, West (1992) discovered the asymptotic

distribution of K. For $K = o(log(n))$, $K - 1$ has a Possion distribution with mean $\alpha(\gamma + log(n))$, where γ is the Euler constant.

Suppose $\pi_1^*, \ldots, \pi_k^*$ is a set of distinct elements of π. Then given k, the π_i^*'s are independent draws from G_0. The joint distribution of $\pi_1^*, \ldots, \pi_k^*$ is just one single term, $\prod_{i=1}^{k} G_0(\pi_i^*)$. The following definition of $k - configuration$ provides some basic ingredients (West, 1990).

For each integer k, $(1 \leq k \leq n)$, let $c = (c_1, c_2, \ldots, c_n)$ be any integer $n-$vector whose elements take values between 1 and k and such that each of the values between 1 and k appears at least once. Define $C_k(c)$ as the configuration of the n elements $\{\pi_i\}$ into exactly k distinct values, $\pi_1^*, \ldots, \pi_k^*$, with $\pi_i = \pi_j^*$ where $c_i = j$, $(i = 1, \ldots, n)$. Then $C_k(c)$ is called a $k - configuration$ of the $\{\pi_i\}$. Finally, let n_j be the number of the $\{\pi_i\}$ equal to π_j^*; that is, $n_j = \#\{c_i = j; i = 1, \ldots, n\}$, for $j = 1, \ldots, k$.

Note that for $k = 1$, $C_1(1, \ldots, 1)$ implies $\pi_1 = \ldots = \pi_n = \pi_1^*$, with $n_1 = n$. At $k = n$, the $C_n(1, \ldots, n)$ implies n distinct values, $\pi_i = \pi_i^*$, and $n_j = 1$ for $j = 1, \ldots, n$. For $1 < k < n$, $C_k(c)$ implies exactly k distinct values $\{\pi_1^*, \ldots, \pi_k^*\}$ among $\{\pi_1, \ldots, \pi_n\}$. The probabilities $\Pr(C_k(c))$ can be found using extensions of results in Antoniak (1974).

$$\begin{aligned} c_1 &= 1, \\ c_2|c_1 &\sim \frac{\alpha\delta_2(c_2) + \delta_{c_1}(c_2)}{\alpha + 1}, \\ &\vdots \\ c_n|c_1, \ldots, c_{n-1} &\sim \frac{\alpha\delta_k(c_n) + \sum_{j=1}^{n-1} \delta_{c_j}(c_n)}{\alpha + n - 1}. \end{aligned}$$

Also, if we define $c^{(i)} = (c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n)$ for $i = 1, \ldots, n$, then the

conditional distribution of $(c_i|c^{(i)})$ is:

$$\Pr(c_i|c^{(i)}) = \frac{\alpha}{\alpha+n-1}\delta_{new\ \ value}(c_i) + \frac{1}{\alpha+n-1}\sum_{j\neq i}\delta_{c_j}(c_i). \qquad (2.6)$$

2.3 Normal Mixtures of Dirichlet Processes

Suppose that observed data y_i's given π_i's are independent and have normal distributions with means μ_i's and variances v_i's, where $\pi_i = (\mu_i, v_i)$. π is a sample of size n from P. To proceed, we need to specify the prior mean $G_0(\cdot)$ of P. Ferguson (1983) discusses Bayesian density estimation by assuming that $G_0(\cdot)$ is the usual normal/inverse-gamma conjugate prior for the two-parameter normal distribution; this has $v_j^{-1} \sim G(v_j|s_0/2, V_0/2)$, a gamma prior with shape parameter $s_0/2$ and scale parameter $V_0/2$, and given v_i, $\mu_i \sim N(\mu_i|m, \tau v_i)$, a normal distribution with mean m and variance τv_j. Ferguson (1983) samples π_i's from the prior distribution of π to obtain the Monte Carlo estimate of the density function. Escobar (1990) uses the Gibbs sampler to analyze the model with known v_i's. Escobar and West (1991) use the Gibbs sampler to compute the predictive density function where $G_0(\cdot)$ is the gamma-normal conjugate prior. For the moment, we assume, s_0, V_0, m and τ are fixed and $Y = (y_1, \ldots, y_n)$. Then, with respect to predicting y_{n+1}, it is clear that $f(y_{n+1}|\pi, Y) \equiv f(y_{n+1}|\pi)$, which may be evaluated as $\int f(y_{n+1}|\pi_{n+1})f(\pi_{n+1}|\pi)d\pi_{n+1}$. This implies

$$(y_{n+1}|\pi) \sim \alpha a_n T_{s_0}(y_{n+1}|m, M_0) + a_n \sum_{i=1}^{n} N(y_{n+1}|\mu_i, v_i), \qquad (2.7)$$

where $T_{s_0}(y_{n+1}|m, M_0)$ is a student-t distribution with s_0 degrees of freedom, mode m and scale factor $\sqrt{M_0}$, $M_0 = (1+\tau)V_0/s_0$ and $a_n = 1/(\alpha+n)$. If there are k distinct values $\pi^* = (\pi_1^*, \ldots, \pi_k^*)$ among $\pi_1, \ldots, \pi_n$ with $\pi_j^* = (\mu_j^*, v_j^*)$, we

can reduce the above model to

$$(y_{n+1}|\pi) \sim \alpha a_n T_{s_0}(y_{n+1}|m, M_0) + a_n \sum_{i=1}^{k} n_j N(y_{n+1}|\mu_j^*, v_j^*). \tag{2.8}$$

As discussed with Professor Mike West and Peter Müller, $G_0(\cdot)$ can be taken as a non-conjugate prior, namely, $v_j^{-1} \sim G(v_j|s_0/2, V_0/2)$ and $\mu_j \sim N(\mu_j|m, \tau)$ where μ_j is independent of v_j. A detailed discussion about this model will be given in the next chapter. After a minor modification of equation (2.7), the predictive density function based on the non-conjugate base prior is

$$(y_{n+1}|\pi) \sim \alpha a_n \int N(y_{n+1}|m, z+\tau) IG(z|s_0/2, V_0/2) dz + a_n \sum_{i=1}^{n} N(y_{n+1}|\mu_i, v_i), \tag{2.9}$$

where $IG(z|s_0/2, V_0/2)$ is an inverse-gamma distribution with shape parameter $s_0/2$ and scale parameter $V_0/2$. Also we can reduce equation (2.8) to

$$(y_{n+1}|\pi) \sim \alpha a_n \int N(y_{n+1}|m, z+\tau) IG(z|s_0/2, V_0/2) dz + a_n \sum_{i=1}^{k} n_j N(y_{n+1}|\mu_j^*, v_j^*). \tag{2.10}$$

The Bayesian prediction, or density estimation problem is solved by

$$f(y_{n+1}|Y) = \int f(y_{n+1}|\pi) f(\pi|Y) d\pi. \tag{2.11}$$

In principle, posterior distributions of k and π^* can be obtained from $f(\pi|Y)$. But it is very difficult to evaluate $f(k|Y)$, $f(\pi_i^*|Y)$ and equation (2.11) for even rather small sample sizes n due to the inherent complexity of the posterior $f(\pi|Y)$. Fortunately, it is possible to use a Monte Carlo approximation in order to compute some interesting posterior distributions.

2.4 Computation of Mixtures of Dirichlet Processes

2.4.1 Gibbs Sampling

Before we discuss a Gibbs sampling technique, let us review some typical ways to deal with computation of mixtures of Dirichlet processes. Antoniak (1974) considers some applications of mixtures of Dirichlet processes for a small sample size. Berry and Christensen (1979) try a parametric approximation for binomial models. Kuo (1986) and Lo (1984) have independently developed similar Monte Carlo integration algorithms to estimate posterior quantities. They draw samples from the original Dirichlet process prior which may often sample values far from the data vector. For example, a straightforward Monte Carlo method suggests that we take

$$\begin{aligned}
\pi_1 &\sim G_0(\cdot), \\
\pi_2 &\sim \frac{\alpha G_0(\cdot) + \delta_{\pi_1}(\pi_2)}{\alpha + 1}, \\
&\vdots \\
\pi_n &\sim \frac{\alpha G_0(\cdot) + \sum_{j=1}^{n-1} \delta_{\pi_j}(\pi_n)}{\alpha + n - 1}.
\end{aligned}$$

This method does not work well, since the likelihood, being a function of π, is peaked at a very small region of the π space depending on data Y. Kuo (1986) proposes a modified Monte Carlo method based on configurations: in some cases it works well, but it is not recommended when the posterior distribution is far away from the prior distribution. As Escobar (1991) points out, performing a Monte Carlo integration by sampling without conditioning on the data results in summing up values in which just a tiny fraction of all the sampled

values has all the weight. This is because the integration is characterized by large areas with minuscule weight and a small area near the data vector which contains almost all the weight. Therefore, sampling near the data vector by using important sampling techniques is extremely important. It is not easy to draw the posterior samples directly because of the complexity of the Dirichlet process. The Gibbs technique is introduced by Geman and Geman (1984) to solve the problems in image processing. Gelfand and Smith (1990) use Gibbs sampling to calculate posterior distributions. Escobar (1990), and Escobar and West (1991) develop Gibbs sampling to analyze mixtures of Dirichlet processes.

Let us define, for each i, $\pi^{(i)} = \{\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_n\}$. For the normal /inverse-gamma conjugate model defined in the last section, Escobar and West (1991) show that the posterior conditional distribution of $(\pi_i|\pi^{(i)}, Y)$ is the mixture

$$(\pi_i|\pi^{(i)}, Y) \sim q_0 G_i(\pi_i) + \sum_{j \neq i}^{n} q_j \delta_{\pi_j}(\pi_i), \tag{2.12}$$

where:

(a) $G_i(\pi_i)$ is the bivariate normal/inverse gamma distribution whose components are $v_i^{-1} \sim G(v_i|(1+s_0)/2, S_i/2)$ with $S_i = V_0 + (y_i - m_0)^2/(1+\tau)$, and $(\mu_i|v_i) \sim N(\mu_i|x_i, Xv_i)$ where $X = \tau/(1+\tau)$ and $x_i = (m + \tau y_i)/(1+\tau)$; and

(b) the weights q_j are defined as

$$\begin{aligned} q_0 &\propto \alpha T_{s_0}(y_i|m, M_0), \\ q_j &\propto N(y_i|\mu_j, v_j), \end{aligned}$$

subject to $q_0 + \ldots + q_{i-1} + q_{i+1} + \ldots + q_n = 1$. The Gibbs sampling can be implemented as follows:

(a) Choose a starting value of π*, say,* $\mu_i = y_i$*, and* $v_i = V_0/s_0$ *for* $i = 1, \ldots, n$*.*

(b) Sequentially sample elements of π *by drawing from the distribution of* $(\pi_1|\pi^{(1)}, Y)$*, then* $(\pi_2|\pi^{(2)}, Y)$*, and so on up to* $(\pi_n|\pi^{(n)}, Y)$*, with the relevant elements of the most recently sampled* $\pi^{(i)}$ *values inserted in the conditioning vectors at each step.*

(c) Return to (b), and proceed iteratively until convergence.

2.4.2 Convergence of the Gibbs Sampling

Escobar and West (1991) discuss theoretical aspects of convergence for above scheme. The idea is to show the Markov chains generated by above algorithm are Harris chains and then apply the ergodic theorem for Harris chains. Doss (1991) gives a following definition of Harris chain.

Definition of Harris Chain (Doss, 1991). *Let* $\{X_k\}$ *be a Markov chain on the state space* $\{\chi, B_\chi\}$ *with transition probability* $P_x(D)$*. That is, the probability of* r*-th step of* X_k *in* D *given* $r-1$*-th step of* X_k *at* x*. Let* $P_x^k(D)$ *be the* k *step transition probability defined by* $P_x^1(D) = P_x(D)$ *and* $P_x^k(D) = \int P_z(D)P_x^{k-1}(dz)$ *for* $k = 2, \ldots$*. For any set* $D \in B_\chi$*, let* $G_x(D) = \sum_{k=1}^{\infty} P_x^k(D)$*. The Markov chain is said to be a Harris chain if there exist sets* A*,* $B \in B_\chi$*, a probability distribution* ρ *on* (χ, B_χ) *concentrated on* B*, and an* $\epsilon > 0$ *such that*

$$G_x(A) > 0 \quad for\ all\ x \in \chi$$

and

$$P_x(D) > \epsilon\rho(D) \quad for\ all\ x \in A, and\ for\ all\ D \subset B.$$

Using the above definition of a Harris chain, Escobar and West (1991) prove the following theorem:

Theorem 2.2 *(Escobar and West, 1991) Let* $\Pr[\pi(r) \in D|\pi(0), Y]$ *be the probability that the r-th iteration of above algorithm produces values* $\pi(r)$ *that lie in a set* D *if above algorithm is started at* $\pi(0)$ *and let* B *be the obvious Borel field, then*

$$lim_{r\to\infty} sup_{D\in B} |\Pr[\pi(r) \in D|\pi(0), Y] - \Pr[\pi \in D|Y]| = 0.$$

The sampling process is computationally straightforward. The result of the process is an approximate draw from $f(\pi|Y)$ in the model as specified.

We can incorporate prior information of hyperparameters s_0, V_0, τ and m into above model. The posterior sampling of those hyperparameters and π can be obtained from the Gibbs sampling technique, just being a minor modification of the above scheme (Escobar and West, 1991). For example, suppose τ and m are unknown and the independent priors of τ and m have the form $m \sim N(m|a, A)$ and $\tau^{-1} \sim G(\tau|t_0/2, R_0/2)$ for some specified hyperparameters a, A, t_0 and R_0. It follows that

(i) given τ and π, m is conditionally independent of Y and normally distributed with mean $E(m|\tau, \pi) = (1-z)a + z\overline{v}\sum(v_j^*)^{-1}\mu_j^*$ and variance $z\tau\overline{v}$ where $z = A/(A+\tau\overline{v}), \overline{v}^{-1} = \sum(v_j^*)^{-1}$ and all sums are over $j = 1, \ldots, k$; also

(ii) given m and π, τ is conditionally independent of Y and has the inverse gamma posterior with shape parameter $(t_0 + k)/2$ and scale parameter $(R_0 + K)/2$ where $K = \sum_{j=1}^{k}(\mu_j^* - m)^2/v_j^*$.

Therefore we can modify above algorithm as follows:

(a) Given τ, m, generate π using above sampling scheme.

(b) Given π, sample m and τ using the relevant distributions as just described.

(c) Repeat (a) and (b) until convergence.

This algorithm is very similar to the first algorithm, the proof of the convergence of the Gibbs sampling is contained in Escobar and West (1991).

Theorem 2.3 *(Escobar and West, 1991)*
Let $\Pr[(\pi(r), \tau(r), m(r)) \in D|\pi(0), \tau(0), m(0), Y]$ *be the probability that, starting with initial values* $(\pi(0), \tau(0), m(0))$*, the extended algorithm produces values* $(\pi(r), \tau(r), m(r))$ *contained in a set D. and let B be the associated Borel field. Then*

$$lim_{r\to\infty} sup_{D\in B}|S(r, D)| = 0,$$

where

$$S(r, D) = \Pr[(\pi(r), \tau(r), m(r)) \in D|\pi(0), \tau(0), m(0), Y] - \Pr[(\pi, \tau, m) \in D|Y].$$

The above sampling scheme is based on the parameter space π, τ and m. MacEachern (1992) describes a new Gibbs sampler algorithm that is implemented on a collapsed state space for the case that v_i is known. Such an implementation of the Gibbs sampler helps the rate of convergence. Specially, for fixed v_i $(i = 1, \ldots, n), \tau$ and m, an iteration of the Gibbs sampler algorithm based on the parameter space involves the generation of $(\mu_1|\mu_2, \ldots, \mu_n, Y)$, then $(\mu_2|\mu_1, \mu_3, \ldots, \mu_n, Y)$, and so on until $(\mu_n|\mu_1, \ldots, \mu_{n-1}, Y)$ is generated. Instead of using the parameter space, R^n, for the state space, and generation of

$(\mu_i|\mu_1,\ldots,\mu_{i-1},\mu_{i+1},\ldots,\mu_n,Y)$ to produce the transition matrix, we coarsen the state space to create a finite state space with evaluable transition probabilities. Suppose the coarsened state space C consist of all partitions of the set of indices $\{1,\ldots,n\}$. An element of the set C, say c, consists of $k \leq n$ distinct groups, $g_1,\ldots,g_k$ of the indices $\{1,\ldots,n\}$. Group i contains n_i of the indices, and $\sum_{i=1}^{k} n_i = n$. A generation step of the Gibbs sampler will take the chain from one element of C to another. Since the state space consists of partitions of μ, when the sampler comes to μ_i, we generate a group membership for μ_i. Formally, we define a reduced state space $C^{(i)}$ to consist of all partitions of the set of indices $\{1,\ldots,n\}-\{i\}$. An individual state of $C^{(i)}$ is $c^{(i)}$. At any point in time, the current status of the chain is a state $c \in C$. This state corresponds to a single state $c^{(i)} \in C^{(i)}$. The generation step is a generation of $c|c^{(i)}$. A more formal statement of the algorithm is

Algorithm (MacEachern 1992)

(1) Choose an initial state c.

(2) Repeat many times:

for $i = 1,\ldots,n$, generate $(c|c^{(i)},Y)$.

MacEachern (1992) proves that above algorithm converges to the posterior distribution of $c|Y$ at a uniform rate. Once we obtain the posterior samples of C, it is very straightforward to make other statistical inferences. We will develop such a technique for unknown v_i's in the next chapter.

Chapter 3

Development of Mixtures of Dirichlet Processes

Although in many situations the full conditional distributions can be easily obtained, it takes a long time to get satisfactory posterior samples. For the most part, the Escobar and West (EW) algorithm described in the last chapter is fairly simple to implement in terms of computations. Due to the correlation of Gibbs samples from the Dirichlet process, the speed of convergence is slow. In order to improve the speed of calculation and accuracy, we present methods that update with configurations. We shall describe in detail several developments of mixtures of Dirichlet processes where the 'new' Gibbs sampling method is illustrated.

3.1 Gibbs Sampling Based on Configurations

Assume that the data $y_1, \ldots, y_n$ are conditionally independent and normally distributed, $(y_i|\pi_i) \sim N(y_i|\mu_i, v_i)$, $\pi_i = (\mu_i, v_i)$, where $\pi_1, \ldots, \pi_n$ are parameters which we need to estimate. For each i, π_i is defined on the parameter space $R \times R^+$. Similar to Escobar and West (1991), the prior distribution for

$\pi = (\pi_1, ..., \pi_n)$ is a Dirichlet process whose properties are described in the previous chapter.

This section provides an algorithm, similar to that of MacEachern (1992), that is based on coarsened Markov chains. In the last chapter, we described the Gibbs sampler algorithm of Escobar and West (1991) by defining the parameter space as the state space of a Markov chain. The following algorithm is based on the Markov chain defined on the configurations. Such algorithm is shown to converge at least as quickly as the one based on the parameter space (MacEachern, 1992).

Let us use the notation defined in section 2.2. Suppose there are k distinct values $\pi^* = (\pi_1^*, \ldots, \pi_k^*)$ among $\pi_1, \ldots, \pi_n$ where $\pi_j^* = (\mu_j^*, v_j^*)$. According to the properties of the Dirichlet process, π_j^* is a random sample from $G_0(\cdot)$ for $1 \leq j \leq k$. Conditional on k and $c_i = j$, for $i = 1, \ldots, n$ we have $(y_i | c_i = j, k, \pi^*) \sim N(y_i | \mu_j^*, v_j^*)$. Therefore, once we know π, we can obtain π^*, k and c, where $c = (c_1, \ldots, c_n)$, and vice versa. Given data Y, the posterior distribution of π can be obtained by the posterior distribution of $(\pi^*,\ k,\ c)$. The posterior distribution of π can be replaced by the posterior distribution of $(\pi^*,\ k,\ c)$. Let us consider the following conditional relationship:

$$f(\pi^*, k, c|Y) = f(c, k|Y) f(\pi^*|c, k, Y). \tag{3.1}$$

The new algorithm to generate posterior samples of π can be implemented by drawing a posterior sample (c, k) from $f(c, k|Y)$ and then drawing a sample π^* from a conditional posterior distribution $f(\pi^*|c, k, Y)$.

Now we want to find the posterior distribution of c, k and the conditional distribution of $(\pi^*|c, k, Y)$. We have

$$f(c, k|Y) \quad \propto \quad f(c, k) f(Y|c, k)$$

$$\propto \quad f(c,k)\int f(Y|c,k,\pi_1^*,\ldots,\pi_k^*)f(\pi_1^*,\ldots,\pi_k^*|c,k)d\pi_1^*\ldots d\pi_k^*.$$

There is no closed form for the posterior distribution of (c,k). Note that the posterior distribution of k can be completely determined by the $f(c,k|Y)$ which is independent of π^*. The conditional posterior distribution of $(\pi^*|c,k,Y)$ is

$$f(\pi^*|c,k,Y) \quad \propto \quad \prod_{j=1}^{k}\{G_0(\pi_j^*)\prod_{\{i:\ c_i=j\}}N(y_i|\mu_j^*,v_j^*)\}.$$

3.1.1 Algorithms

The main difficulty in applications, so far, is evaluating $f(c,k|Y)$. It gets even more difficult to draw the configuration from the posterior distribution as n increases, since there are $n!$ possible values for c, for instance when $n = 10$, $n!$ =3628800. However, the full conditional distribution of $(c_i|c^{(i)},k,Y)$ is easily obtained. Conditional on k, let $c^* = (c_1^*,\ldots,c_k^*)$ denote the distinct elements of $c = (c_1,\ldots,c_n)$.

For $j = 1,\ldots,k$, let n_j = number of i such that $\{c_i = c_j^*\}$,
$\overline{y_j} = \sum_{\{i:\ c_i=c_j^*\}} y_i/n_j$, $SS_j = \sum_{\{i:\ c_i=c_j^*\}}(y_i-\overline{y_j})^2$.

If we delete the i-th data point y_i, there are $n-1$ data points left. For $i = 1,\ldots,n$, let $k^{(i)}$ denote the number of different values of $\pi^{(i)}$, where $\pi^{(i)} = (\pi_1,\ldots,\pi_{i-1},\pi_{i+1},\ldots,\pi_n)$. Also denote $n_j^{(i)}$ = number of $\{l : l \neq i, c_l = c_j^*\}$, $\overline{y_j}^{(i)} = \sum_{\{l:\ l\neq i,c_l=c_j^*\}} y_l/n_j^{(i)}$, $SS_j^{(i)} = \sum_{\{l:\ l\neq i,c_l=c_j^*\}}(y_l-\overline{y_j}^{(i)})^2$, for $j = 1,\ldots,k^{(i)}$. $\pi^{*(i)} = (\pi_1^{*(i)},\ldots,\pi_{k^{(i)}}^{*(i)})$ gives the distinct elements of $\pi^{(i)}$.

The conditional distribution of $(\pi_i|\pi^{(i)},Y)$ is given by equation (2.12), which

can also be written as:

$$(\pi_i|\pi^{*(i)}, c, Y) \propto \alpha T_{s_0}(y_i|m, M_0) f(\pi_i|y_i) + \sum_{j=1}^{k^{(i)}} n_j^{(i)} N(y_i|\mu_j^{*(i)}, v_j^{*(i)}) \delta_{\pi_j^{*(i)}}(\pi_i), \tag{3.2}$$

where $M_0 = (1+\tau)V_0/s_0$, $(\mu_j^{*(i)}, v_j^{*(i)}) = \pi_j^{*(i)}$ and $f(\pi_i|y_i)$ is the bivariate normal/inverse gamma distribution whose components are $v_i|y_i \sim IG((1+s_0)/2, \{V_0+(y_i-m)^2/(1+\tau)\}/2)$, and $(\mu_i|v_i) \sim N(\{\tau y_i+m\}/\{1+\tau\}, \tau v_i/\{1+\tau\})$. By integrating out π^*, following results are obtained.

$$(c_i|c^{*(i)}, k, Y) \sim q_0 \delta_{c_{new}}(c_i) + \sum_{j=1}^{k^{(i)}} q_j \delta_{c_j^{*(i)}}(c_i), \tag{3.3}$$

where the weights q_j are defined as

$$\begin{aligned} q_0 &\propto \alpha T_{s_0}(y_i|m, (1+\tau)V_0/s_0), \\ q_j &\propto n_j^{(i)} T_{s_0+n_j^{(i)}}(y_i|m_j^{(i)}, M_j^{(i)}), \quad j = 1, \ldots, k^{(i)}, \end{aligned}$$

subject to $q_0 + q_1 + \ldots q_{k^{(i)}} = 1$, with

$$\begin{aligned} m_j^{(i)} &= \frac{\tau n_j^{(i)} \overline{y_j^{(i)}} + m}{1+\tau n_j^{(i)}}, \\ M_j^{(i)} &= \left(\frac{1+\tau(n_j^{(i)}+1)}{1+\tau n_j^{(i)}}\right) \left(\frac{V_0 + SS_j^{(i)} + n_j^{(i)}(\overline{y_j^{(i)}} - m)^2/(1+\tau n_j^{(i)})}{s_0 + n_j^{(i)}}\right), \end{aligned}$$

for $j = 1, \ldots, k_{(i)}$.

Once we have the posterior samples of (c, k) from the Gibbs sampler, it is very straightforward to obtain the conditional distribution of $(\pi^*|c, k, Y)$. For $j = 1, \ldots, k$,

$$(\mu_j^*|v_j^*, c, k, Y) \sim N(\mu_j^*| \frac{\tau n_j \overline{y_j} + m}{1 + \tau n_j}, \frac{\tau v_j^*}{1 + \tau n_j}), \tag{3.4}$$

$$(v_j^*|c, k, Y) \sim IG(v_j^*| \frac{s_0 + n_j}{2}, \frac{V_0 + SS_j + (\overline{y_j} - m)^2/(\tau + 1/n_j)}{2}). \tag{3.5}$$

This permits an iterative procedure that requires only the initial values of c, k, π^*. Let us assume τ, m are known. At each sampling stage, given old values of π, we may sample c by using equation (3.3), and an updated k. Substitution of these in the above equations gives a new sample of π^*. The key step is drawing samples for c, which is described below.

Algorithm 3.1a

Step 1: Choose an initial state c. For example, let $c = (1, 2, \ldots, n)$.

Step 2: Generate c by recursive application of (3.3).

Step 3: Generate π^* from the distribution of $(\pi^*|c, k)$ (3.4 and 3.5).

Step 4: Repeat steps 2 and 3 many times.

The Gibbs sampler is applied only on step 2 of this algorithm. Step 3 is used to make inferences about π. Therefore we need to consider the convergence of the Gibbs sampler. We have the following theorem:

Theorem 3.1 *Algorithm 3.1a converges to the posterior distribution of* $c|Y$ *at a uniform rate.*

Proof. The Markov chain obtained from equation (3.3) is an aperiodic, irreducible chain on a finite state space. The result follows from standard results for Markov chains. See MacEachern (1992) and Tierney (1991).

Also, we can use the ergodic theorem for Harris chains to prove the following theorem:

Theorem 3.2 *Let* $\Pr[(c(r),k(r)) \in D|c(0),k(0),Y]$ *be the probability that the* r*-th iteration of algorithm 3.1a produces values* $(c(r),k(r))$ *that lie in a set* D*, if the algorithm is started at* $(c(0),k(0))$*, and let* B *be a* σ*-field, then*

$$lim_{r\to\infty} sup_{D\in B} |\Pr[(c(r),k(r)) \in D|c(0),k(0),Y] - \Pr[(c,k) \in D|Y]| = 0.$$

The proof is given in Section 3.4.

As mentioned before, the Bayesian prediction, or density estimation problem is solved by summarizing the unconditional predictive density

$$\begin{aligned} f(y_{n+1}|Y) &= \int f(y_{n+1}|\pi)\ f(\pi|Y)d\pi \\ &= \alpha a_n T_{s_0}(y_{n+1}|m_0,M_0) \\ &\qquad + \int a_n \sum_{j=1}^{k} n_j N(y_{n+1}|\mu_j^*,v_j^*)\ f(\pi^*,c,k|Y)d\pi^*\ dc\ dk \\ &= \alpha a_n T_{s_0}(y_{n+1}|m_0,M_0) \\ &\qquad + \int a_n \sum_{j=1}^{k} n_j T_{s_0+n_j}(y_{n+1}|m_j,M_j)\ f(c,k|Y)dc\ dk, \end{aligned}$$

where $a_n = 1/(\alpha+n)$ and m_j, M_j are defined as

$$\begin{aligned} m_0 &= m, \\ M_0 &= (1+\tau)V_0/s_0, \\ m_j &= \frac{\tau n_j \overline{y_j} + m}{1+\tau n_j}, \qquad j = 1,\ldots,k, \\ M_j &= \left(\frac{1+\tau(n_j+1)}{1+\tau n_j}\right)\left(\frac{V_0 + SS_j + n_j(\overline{y_j}-m)^2/(1+\tau n_j)}{s_0+n_j}\right), \qquad j = 1,\ldots,k. \end{aligned}$$

So far we have emphasized the algorithm based on configurations. It is important, however, to bear in mind that there are several methods available to evaluate the posterior distributions. If v^* or π^* can not be integrated out of (3.2), then c, k can be sampled based on the updated v^* or π^*. For example, c can be sampled based on v^* as follows:

$$(c_i|v_i^{*(i)}, c^{(i)}, Y) \propto \alpha T_{s_0}(y_i|m, M_0)\delta_{new}(c_i) + \sum_{j=1}^{k^{(i)}} n_j^{(i)} N(y_i|m_j^{(i)}, z_j^{(i)})\delta_{c_j^*}(c_i) \quad (3.6)$$

with $z_j^{(i)} = v_j^{*(i)} + (\tau v_j^{*(i)})/(1+\tau n_j^{(i)})$ or c can be sampled based on π^* as follows:

$$(c_i|\pi^{*(i)}, c^{(i)}, Y) \propto \alpha T_{s_0}(y_i|m, M_0)\delta_{new}(c_i) + \sum_{j=1}^{k^{(i)}} n_j^{(i)} N(y_i|\mu_j^{*(i)}, v_j^{*(i)})\delta_{c_j^*}(c_i). \quad (3.7)$$

Based on (3.6), we can obtain the following algorithm which is similar to Algorithm 3.1a.

Algorithm 3.1b

Step 1: Choose an initial state c. For example, let $c = (1, \ldots, n)$.

Step 2: Generate c through recursive application of (3.6).

Step 3: Generate π^* from the conditional distribution $(\pi^*|c, k, Y)$ by using the update c and k from step 2.

Step 4: Repeat steps 2 and 3 many times.

If we can not integrate π^* out of (3.2), The following algorithm can be obtained from (3.7).

Algorithm 3.1c

Using all steps of Algorithm 3.1b, except step 2, which is changed to generating c using (3.7).

Algorithms 3.1a, 3.1b and 3.1c are modified MacEachern's algorithms. They are of interest since all of them are based on configurations. According to equations (3.4) and (3.5), the values of μ_j^* and v_j^* can be changed by every Gibbs iteration, while these values may not be changed by using the EW algorithm, see also equation (2.12). If π^* changes slowly, then the Gibbs sampling converges slowly. Algorithm 3.1a can be used only there are close forms for $q_0, q_1, \ldots, q_{k^{(i)}}$ in equation (3.3). In some cases, there are no close forms for q_i's. For example, the non-conjugate prior model was described in Section 2.3. The q_i is an integration of v_i^*. Instead of evaluating the numerical integration for q_i, Algorithm 3.1b can be used very easily.

Similar to Algorithm 3.1a, we have the following theorems for algorithm 3.1b and 3.1c:

Theorem 3.3 *Let* $\Pr[(c(r), k(r), v(r)) \in D | c(0), k(0), v(0), Y]$ *be the probability that the r-th iteration of algorithm 3.1b produces values* $(c(r), k(r), v(r))$ *that lie in a set D, if the algorithm is started at* $(c(0), k(0), v(0))$*, and let B be a* σ*-field, then*

$$lim_{r\to\infty} sup_{D\in B} |S(r, D)| = 0,$$

where

$$S(r, D) = \Pr[(c(r), k(r), v(r)) \in D | (c(0), k(0), v(0)), Y] - \Pr[(c, k, v) \in D | Y].$$

Theorem 3.4 *Let* $\Pr[(c(r), k(r), \pi(r)) \in D | c(0), k(0), \pi(0), Y]$ *be the probability that the r-th iteration of algorithm 3.1c produces values* $(c(r), k(r), \pi(r))$ *that lie in a set D, if the algorithm is started at* $(c(0), k(0), \pi(0))$*, and let B be a* σ*-field, then*

$$lim_{r\to\infty} sup_{D\in B} |S(r, D)| = 0,$$

where

$$S(r, D) = \Pr[(c(r), k(r), \pi(r)) \in D | c(0), k(0), \pi(0), Y] - \Pr[(c, k, \pi) \in D | Y].$$

The proofs are given in Section 3.4.

3.1.2 An Example

In order to compare these algorithms, we use a very simple example where we can calculate the exact posterior distribution for π, and compare the results from above algorithms with the exact values. Suppose $n = 2$ and data are given by $y_1 = -5.0,\ y_2 = 5.0$, with parameters $\alpha = 1,\ m = 1.0,\ \tau = 10,\ s_0 = 2,\ V_0 = 10$. It is trivial to get the following results.

$$\begin{aligned}
f(\pi_1, \pi_2 | y_1, y_2) &= A\ G_0(\pi_1)\ (\alpha G_0(\pi_2) + \delta_{\pi_1}(\pi_2))\ N(y_1|\mu_1, v_1)\ N(y_2|\mu_2, v_2), \\
f(\pi_i | y_1, y_2) &= A\ (\alpha\ f(y_1)f(y_2)f(\pi_i|y_i) + f(y_1, y_2)f(\pi_i|y_1, y_2))\ ,\ i = 1, 2, \\
A^{-1} &= \alpha f(y_1)f(y_2) + f(y_1, y_2), \\
f(y_1, y_2) &= \frac{1}{2\sqrt{\pi}} \frac{B}{B^*}\ T_{s_0+1}(\overline{y}|m, (\tau + 0.5)V_0^*/(s_0 + 1)),
\end{aligned}$$

where $f(y_i) = T_{s_0}(y_i|m, (1+\tau)V_0/s_0)$ for $i = 1, 2,\ \overline{y} = (y_1+y_2)/2$ and $V_0^* = V_0 + (y_1-y_2)^2/2$. We define $B = (V_0/2)^{s_0/2}/\Gamma(s_0/2)$ and $B^* = (V_0^*/2)^{(s_0+1)/2}/\Gamma((s_0+1)/2)$. So that, $\frac{1}{2\sqrt{\pi}}\frac{B}{B^*} = 1/(24\sqrt{30})$; also

(a)$f(\pi_i|y_i)$ is the bivariate normal/inverse gamma distribution whose components are $v_i \sim IG(v_i|(1+s_0)/2,\ V_i/2)$ with $V_i = V_0+(y_i-m)^2/(1+\tau)$, and $(\mu_i|v_i) \sim N(\mu_i|x_i, Xv_i)$ with $X = \tau/(1+\tau)$ and $x_i = (m+\tau y_i)/(1+\tau)$; and

(b)$f(\pi_i|y_1, y_2)$ is the bivariate normal/inverse gamma distribution whose components are $v_i \sim IG(v_i|(2+s_0)/2,\ (V_0^* + (\overline{y}-m)^2/(\tau+0.5))/2)$, and $(\mu_i|v_i) \sim N(x^*, X^*v_i)$ with $X^* = \tau/(1+2\tau)$ and $x^* = (m+2\tau\overline{y})/(1+2\tau)$. Here, the exact mean of μ_i can be obtained as follows:

$$E(\mu_i|y_1, y_2) \quad = \quad A\,[\alpha f(y_1)f(y_2)x_i + f(y_1, y_2)x^*], \quad i = 1, 2.$$

Table (3.1) shows a comparison of the four different algorithms. The results are based on burn-in cycles of 2000 iteration. The Monte Carlo sample size used here was 500. In Table (3.1), columns two and three give posterior probabilities for $k = 2$ and 3 respectively. The posterior means of μ_1 and μ_2 are given in columns four and five respectively.

At this stage, we can see that the posterior distribution of k obtained from Algorithm (3.1a) is very close to the exact value of $p(k|Y)$. This is because $p(k|Y)$ can be obtained from $p(c, k|Y)$, independent of π^*. In this case, we prefer Algorithm (3.1a). In fact, all those algorithms can give us satisfactory results if the Monte Carlo sample size is large enough. If we increase the Monte Carlo sample size, for example, to 5000, then algorithms (3.1a), (3.1b) and (3.1c) do not improve the results a lot, while algorithm EW gives us better results.

Table 3.1: Performance of Different Algorithms — Monte Carlo Size of 500

	$p(k=1\|Y)$	$p(k=2\|Y)$	$\mathsf{E}(\mu_1\|Y)$	$\mathsf{E}(\mu_2\|Y)$
Exact	0.13692	0.86308	-3.83810	4.00807
Algorithm (3.1a)	0.12400	0.87600	-3.80477	3.90262
Algorithm (3.1b)	0.11800	0.88200	-4.06042	4.04896
Algorithm (3.1c)	0.11000	0.89000	-3.98071	4.17450
Algorithm EW	0.08400	0.91600	-4.11829	4.07017

Table 3.2: Performance of Different Algorithms — Monte Carlo Size of 5000

	$p(k=1\|Y)$	$p(k=2\|Y)$	$\mathsf{E}(\mu_1\|Y)$	$\mathsf{E}(\mu_2\|Y)$
Exact	0.13692	0.86308	-3.83810	4.00807
Algorithm (3.1a)	0.12500	0.87500	-3.83152	4.11672
Algorithm (3.1b)	0.11900	0.88100	-4.02846	4.09393
Algorithm (3.1c)	0.12300	0.87700	-3.83472	4.09366
Algorithm EW	0.12800	0.87200	-3.84651	4.05617

In most cases, we do not know τ and m. The conditionally conjugate structure built into the model easily allows for an extension of the sampling based analysis to include learning about the prior parameters m and/or τ. Suppose m has a reference prior, and $\tau \sim IG(\tau|t_0/2, R_0/2)$ independently, for some specified hyper-parameters t_0 and R_0. It follows that

$$\begin{aligned}(\tau|c,k,\pi^*) &\sim IG(\tau|(t_0+k-1)/2,(R_0+S_\tau)/2),\\ (m|\tau,c,k,\pi^*) &\sim N\left(m|\sum_{j=1}^{k}\mu_j^* v_j^{*-1}/\sum_{j=1}^{k} v_j^{*-1}, \tau/\sum_{j=1}^{k} v_j^{*-1}\right),\end{aligned}$$

where $S_\tau = \sum_{i=1}^{k}\left((\sum_{j=1}^{k}\mu_j^* v_j^{*-1}/\sum_{j=1}^{k} v_j^{*-1} - \mu_i^*)^2/v_i^*\right)$.

Incorporating m and/or τ into the iterative resampling scheme provides samples from the complete joint posterior of $(c, k, \pi^*, m, \tau|Y)$. Furthermore, we can also consider α as an unknown parameter and sample α from the Gibbs sampling scheme. Suppose the prior distribution of α is a gamma distribution with parameters a_0 and b_0, denoted by $G(\alpha|a_0, b_0)$, West (1992b) shows that $f(\alpha|k, \pi, Y) \equiv f(\alpha|k)$. He also shows that we can sample a dummy variable z from $Beta(z|\alpha+1, n)$, then sample α from a mixture of two gamma densities as follows:

$$(\alpha|z, k) \sim w_z G(\alpha|a_0+k, b_0-log(z))+(1-w_z)G(\alpha|a_0+k-1, b_0-log(z)) \quad (3.8)$$

with weights w_z defined by

$$\frac{w_z}{1-w_z} = \frac{a_0+k-1}{n\ (b_0-log(z))}. \quad (3.9)$$

Thus, the Gibbs sampling algorithm generates, from some initial setup for $c, k, \pi^*, m, \tau, \alpha$, a sequence of samples of $(c, k, \pi^*, m, \tau, \alpha)$. Each completed iteration consists of the following four steps:

(c, k) step: Generate c, k from one of the conditional distributions (3.3), (3.6) or (3.7).

π^* step: Generate π^* from the conditional distribution $(\pi^*|c, k, m, \tau, \alpha)$.

(m, τ) step: Generate m, τ from the conditional distribution of $(m, \tau|c, k, \pi^*)$.

α step: Generate a dummy variable z from $Beta(\alpha+1, n)$, then generate α from the conditional distribution (3.8).

Very often the π^* step and (m, τ) step are explicit. They are easily implemented. The convergence of the Gibs sampler is given by the following theorem.

Theorem 3.5 *Let*

$\Pr[(c(r), k(r), \pi(r), \tau(r), m(r), \alpha(r)) \in D | c(0), k(0), \pi(0), \tau(0), m(0), \alpha(0), Y]$ *be the probability that the r-th iteration of the above algorithm produces values* $(c(r), k(r), \pi(r), \tau(r), m(r), \alpha(r))$ *that lie in a set D, if the algorithm is started at* $(c(0), k(0), \pi(0), \tau(0), m(0), \alpha(0)$*, and let B be a* σ*-field, then*

$$lim_{r \to \infty} sup_{D \in B} |S(r, D)| = 0,$$

where

$$\begin{aligned} S(r, D) &= \Pr[\phi(r) \in D | \phi(0), Y] - \Pr[\phi \in D | Y], \\ \phi(r) &= (c(r), k(r), \pi(r), \tau(r), m(r), \alpha(r)), \\ \phi(0) &= (c(0), k(0), \pi(0), \tau(0), m(0), \alpha(0)), \\ \phi &= (c, k, \pi, \tau, m, \alpha). \end{aligned}$$

The proof is similar to the one for Theorem 3.4 and omitted here.

Also, an exceptionally simple general proof, based on Bayes' theorem, shows that $E(m | \tau, c, k, \pi^*, Y) = \sum_{j=1}^{k} w'_j \mu^*_j$ where w'_j is proportional to $1/v^*_j$, constrained to $\sum_{j=1}^{k} w'_j = 1$. These 'weights' are related to the variances of v^*_j. As discussed in West, Müller and Escobar (1993), if one of the data points is far away from the rest, and that point comes from a distinct component with a small variance, then the conditional posterior for m will be peaked around the data point which is not reasonable. Such a problem can be avoided, if we change the conjugate prior of $G_0(\cdot)$ to a non-conjugate prior, Assuming the non-conjugate prior is $\mu_i \sim N(\mu_i | m, \tau)$, $v_i \sim IG(v_i | s_0/2, V_0/2)$, and μ_i is independent of v_i, the expectation of m will be independent of v or v^*. In the next section, we will discuss the non-conjugate prior model.

3.2 Gibbs Sampling Based on a Non-conjugate Base Prior

As pointed out in the last section, the conjugate baseline prior is not reasonable if some of the data points are well separated from the rest. A non-conjugate prior can be developed by using the model from the end of the previous section. Algorithms in section 3.1 and EW algorithm can be easily derived here. For example, the posterior conditional distribution of $(\pi_i|\pi^{(i)}, Y)$ can be obtained as follows:

$$(\pi_i|\pi^{(i)}, Y) \sim q_0' G_i(\pi_i) + \sum_{j\neq i}^{n} q_j' \delta_{\pi_j}(\pi_i), \tag{3.10}$$

where:

(i) $G_i(\pi_i)$ is the posterior density function of π given the data point y_i, that is, $G_i(\pi) \propto G_0(\pi)N(y_i|\mu_i, v_i)$. There is no closed form expression for the posterior distribution of π given the y_i. The density function of v_i is proportional to $N(y_i|m, \tau + v_i)\ IG(v_i|s_0/2, V_0/2)$. Given v_i, μ_i has a normal distribution with mean $(\tau y_i + m v_i)/(\tau + v_i)$ and variance $\tau v_i/(\tau + v_i)$; and

(ii) the weights q_j' are defined as

$$\begin{aligned} q_0' &\propto \alpha \int N(y_i|m, \tau + z) IG(z|s_0/2, V_0/2) dz, \\ q_j' &\propto N(y_i|\mu_j, v_j), \end{aligned}$$

subject to $q_0' + \ldots + q_{i-1}' + q_{i+1}' + \ldots + q_n' = 1$. The EW algorithm can be implemented as before, see Section 2.4.1. The following theorem states the convergence of above algorithm for known τ, m, α.

Theorem 3.6 *Let* $\Pr[\pi(r) \in D|\pi(0), Y]$ *be the probability that the r-th iteration of the modified EW algorithm produces values* $\pi(r)$ *that lie in a set* D *if the algorithm is started at* $\pi(0)$ *and let* B *be the obvious Borel field, then*

$$lim_{r\to\infty} sup_{D\in B} |\Pr[\pi(r) \in D|\pi(0), Y] - \Pr[\pi \in D|Y]| = 0.$$

The proof is contained in Section 3.4.

In order to implement other three algorithms, we have to obtain the equations which are similar to (3.3), (3.6) and (3.7). Since there is no closed form expression for $(\pi_i|\pi^{(i)}, Y)$, the algorithm 3.1a is hard to implement. However, we have the following equations:

$$(c_i|c^{*(i)}, k, v^{*(i)}, Y) \sim q_0''\delta_{new}(c_i) + \sum_{j=1}^{k^{(i)}} q_j''\delta_{c_j^*}(c_i), \tag{3.11}$$

where the weights q_j'' are defined as

$$\begin{aligned} q_0'' &\propto \alpha \int N(y_i|m, \tau + z) IG(z|s_0/2, V_0/2) dz, \\ q_j'' &\propto n_j^{(i)} N(y_i|m_j^{(i)''}, M_j^{(i)''}), \quad j = 1, \ldots, k^{(i)}, \end{aligned}$$

subject to $q_0'' + q_1'' + \ldots + q_{k^{(i)}}'' = 1$, with

$$\begin{aligned} m_j^{(i)''} &= \frac{\tau n_j^{(i)} \overline{y_j^{(i)}} + m v_j^{*(i)}}{v_j^{*(i)} + n_j^{(i)} \tau}, \\ M_j^{(i)''} &= \frac{\tau v_j^{(i)}}{\tau n_j^{(i)} + v_j^{*(i)}}, \end{aligned}$$

for $j = 1, \ldots, k_{(i)}$. The integration in calculating q_0'' can be easily avoided by incorporating a dummy variable z' in the Gibbs scheme. That is, we can sample z' from the inverse gamma distribution $IG(z'|s_0/2, V_0/2)$, then substitute

z' into the normal distribution with mean m and variance $\tau + z'$, evaluate the density function at y_i. According to (3.10), we can implement the algorithm which is similar to Algorithm 3.1b. It is also very easily to have Algorithm 3.1c based on the non-conjugate prior. Again, the steps to obtain samples from posterior distributions of the hyperparameters τ, m and α can be added easily to the Gibbs sampler. Suppose m has a reference prior and $\tau \sim IG(\tau|t_0/2, R_0/2)$ independently, then we have following conditional distributions:

$$\begin{aligned} \tau &\sim IG(\tau|(t_0+k-1)/2, (R_0+S)/2), \\ (m|\tau) &\sim N(m|\overline{\mu^*}, \tau/k), \end{aligned}$$

where $S = \sum_{j=1}^{k}(\mu_j^* - \overline{\mu^*})^2$ and $\overline{\mu^*} = \sum_{j=1}^{k}\mu_j^*/k$ We can use the equation (3.8) to sample α. We can prove the convergence of the modified EW algorithm which includes sample τ, m and α.

Theorem 3.7 *Let* $\Pr[(\pi(r), \tau(r), m(r), \alpha(r)) \in D|\pi(0), \tau(0), m(0), \alpha(0), Y]$ *be the probability that, starting with initial values* $(\pi(0), \tau(0), m(0), \alpha(0))$*, the modified EW algorithm produces values* $(\pi(r), \tau(r), m(r), \alpha(r))$ *contained in a set D. and let B be the associated Borel field. Then*

$$lim_{r\to\infty} sup_{D\in B}|S(r, D)| = 0,$$

where

$$\begin{aligned} S(r, D) &= \Pr[(\pi(r), \tau(r), m(r), \alpha(r)) \in D|\pi(0), \tau(0), m(0), \alpha(0), Y] \\ &\quad - \Pr[(\pi, \tau, m, \alpha) \in D|Y]. \end{aligned}$$

The proof is given in Section 3.4. For other algorithms, the convergence of the Gibbs sampler is similar to Theorems 3.3, and 3.4.

The Gibbs sampling method was tested on a simulated data set of 30 points from a mixture of normal distributions $0.5N(Y|0,1) + 0.5N(Y|4,1)$. The histogram of simulated data is displayed in Figure (3.1). The prior distributions were $v \sim IG(v|10/2, 10/2)$, $\tau \sim IG(\tau|2/2, 20/2)$, $\alpha \sim G(\alpha|4,8)$ and the reference prior for m. The initial value for the μ_i was taken to be y_i. All simulations were run for 10,000 iterations, after a burn-in of 2,000 cycles.

In this analysis, since the prior mean of α is small and the prior mean of τ is large, the posterior mode of k will concentrate on a smaller number. Table (3.3) presents estimates of the posterior distribution of the number of clusters under the conjugate and non-conjugate prior distributions. The posterior mode of k based on conjugate prior is 3, while the posterior mode of k from non-conjugate prior is 2. Our explanation is that the prior variance of μ_i is τ for the nonconjugate prior while the prior variance of μ_i for the conjugate prior is $\tau V_0/(s_0 - 2)$ which is always larger than τ for any $\tau \sim IG(\tau|1, 10)$. If the variance of μ_i is smaller, then it is more likely for μ_i to take values around m, and to show less components .

Figure 3.1(b) displays the predictive density functions using conjugate and non-conjugate priors along with the mixture used to simulate the data. In this case, the non-conjugate prior gives a better performance.

3.3 Convergence Issues

Whether one uses the conjugate base prior model or the non-conjugate base prior model, diagnostics for monitoring convergence will be of value. McEachern (1992) proves the convergence of his algorithm which is a special case of our model. This section is intended as a practical guide to the Gibbs sam-

Table 3.3: Probability Distributions of k for Simulation Comparison

k	Conjugate prior	Non-conjugate prior
1	0.0002	0.0000
2	0.2869	0.9764
3	0.3264	0.0190
4	0.2231	0.0032
5	0.1064	0.0013
6	0.0378	0.0001
7	0.0135	0.0000
8	0.0044	0.0000
9	0.0013	0.0000

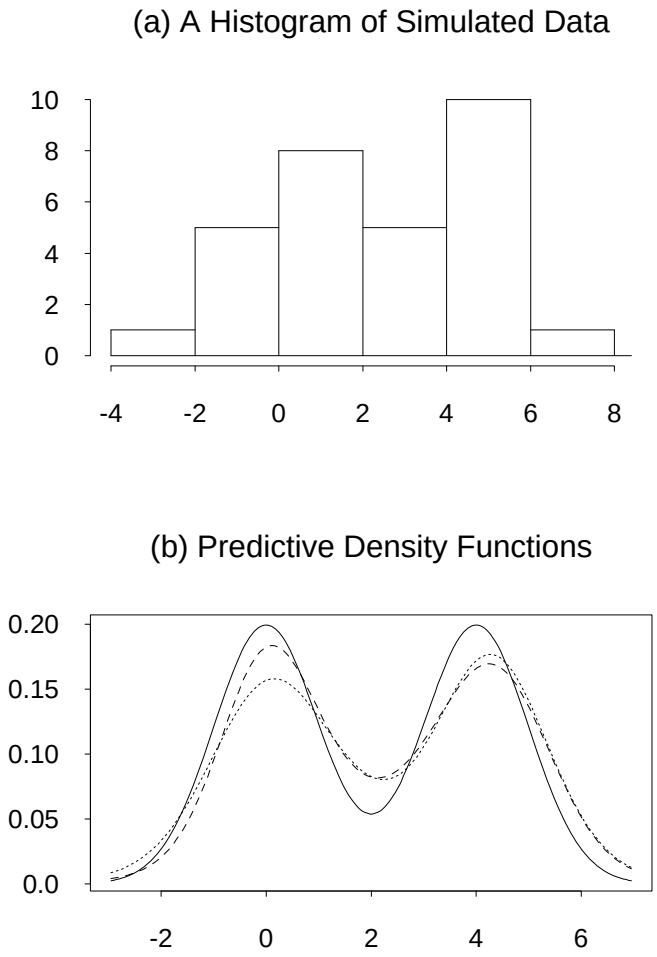


Figure 3.1: Predictive Density Functions for Simulation Comparisons

—— true density function
..... predictive density function of conjugate prior
- - - predictive density function of non-conjugate prior

pler. Though much has been written on the subject recently and the theory has been considerably clarified, some of the simplest aspects have remained controversial. The most basic issue in dispute is whether valid inference from Gibbs sampler results from averaging over one long run of the chain, as the name of the method and all the theory suggest, or whether multiple shorter runs are desirable or even necessary for valid inference. The view expounded here is that there can be no valid inference from runs that are too short and that if runs are long enough, one run suffices. Throughout this dissertation, we recommend that inference be based on a single long run, but that this be monitored using carefully chosen diagnostics, and that starting values and the exact form of the algorithm be chosen on the basis of experimentation. A bad starting value can lead to slow convergence. This can be diagnosed from one run and rectified by changing the starting value. Diagnostics should monitor all the key features of the model, such as hyperparameters τ, m, α in hierarchical models, as well as a selection of less essential features such as v^*. If only the quantities of interest are monitored, lack of convergence can be missed.

Since the starting value can have an important effect on the performance of the Gibbs sampler, there is certainly advantage to a systematic search for good starting values. For the previous analysis, we choose initial values for $\mu_i = y_i, v_i = V_0/s_0$ for $i = 1, \cdots, n$. The starting values of m, τ and α are $0, R_0/t_0$ and a_0/b_0 respectively. The initial value of K, the number of components, is always equal to n, the number of data y_i.

In Section 3.2, we describe the model by using the conjugate base prior. To illustrate the convergence of the Gibbs sampler and how the starting value affect the convergence, we use the algorithm (3.1a) and sample first 1000 iterations and exhibit them in following three figures. Instead of displaying all

features of the model, we only select six interesting parameters to show the convergence of Gibbs sampler here. We also monitor all other convergence features for other parameters which are not displayed here.

Figure 3.2 shows the convergence of Gibbs sampler by choosing starting values $\mu_i = y_i, v_i = V_0/s_0$ for $i = 1, \cdots, n$. The initial values of m and τ are 0 and R_0/t_0. We fix α for the first 500 iterations, then incorporate it into Gibbs sampler. Figure 3.2(a) displays random samples of the number of components for the first 1000 burn-in cycles. We choose the initial value of K as 30. After one Gibbs runs, Gibbs sampler gives us $K = 11$. After 20 Gibbs iterations, K randomly choose values of 3,4,5 and even 6 or 7. From this plot, it is very clearly the random samples are dependent. This is a drawback of Gibbs sampler. However we think the distribution of K is reaching the stable distribution which is the posterior distribution of K.

Figure 3.2(b), (c) (d) and (e) show us the first 1000 random samples from Gibbs sampler for μ_1, v_1, m_0 and τ. After several Gibbs runs, those parameters reach their stable distribution. Figure 3.2 (f) exhibits the convergence issue about α. We get the first random sample of α after 500 iterations.

Figure 3.3 displays all features of Gibbs sampler for those parameters discussed in Figure 3.3, but we choose different starting values for α where α is sampled after the first iteration. After several Gibbs iterations, Figure 3.3 displays the same feature as Figure 3.2.

Figure 3.4 exhibits the features of Gibbs sampler by choosing the same starting values for those parameters as in Figure 3.3 except $\mu_i = 0.0$. From above three figures, we are sure Gibbs sampler converges.

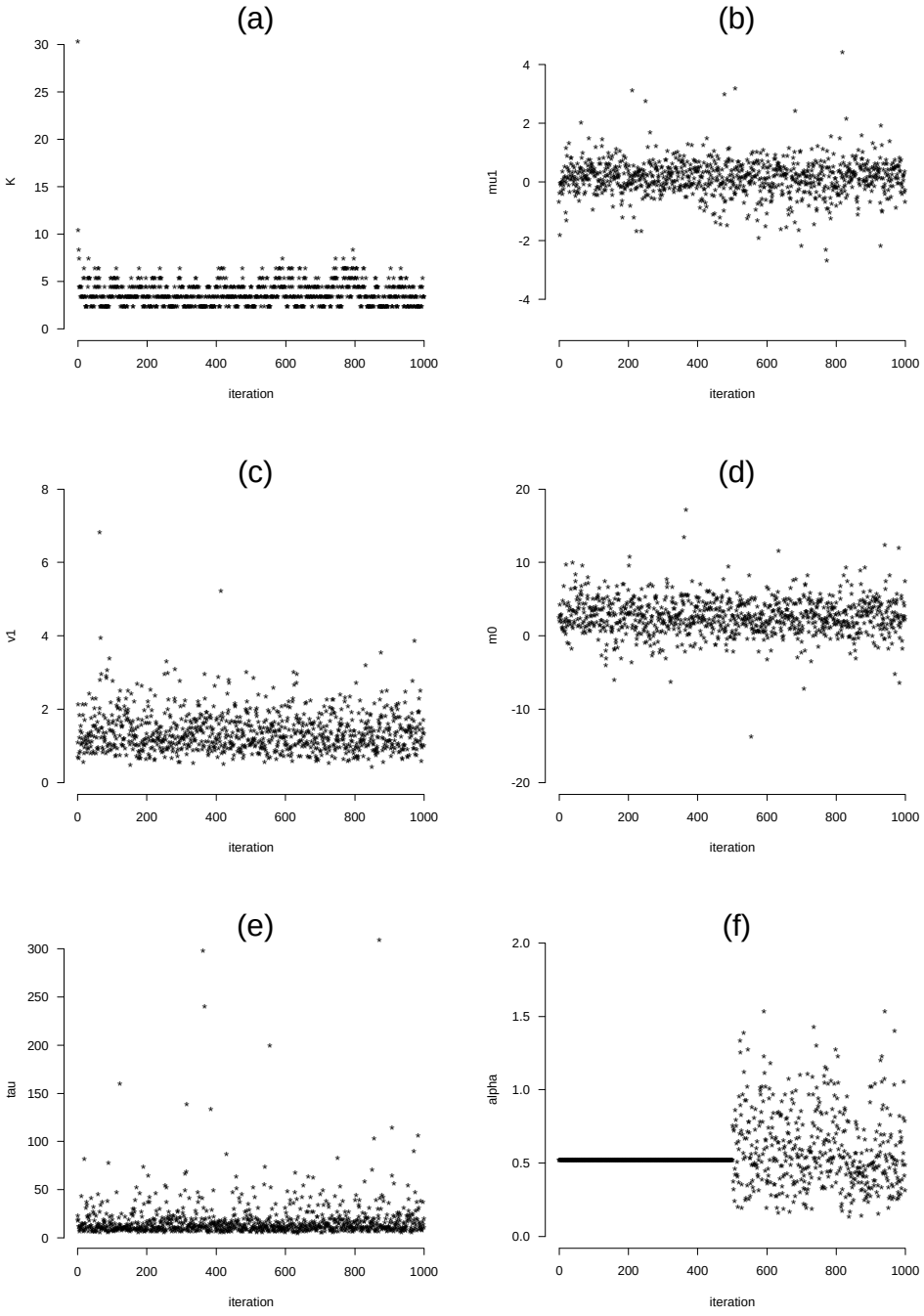


Figure 3.2: Convergence of Gibbs Sampler (I)

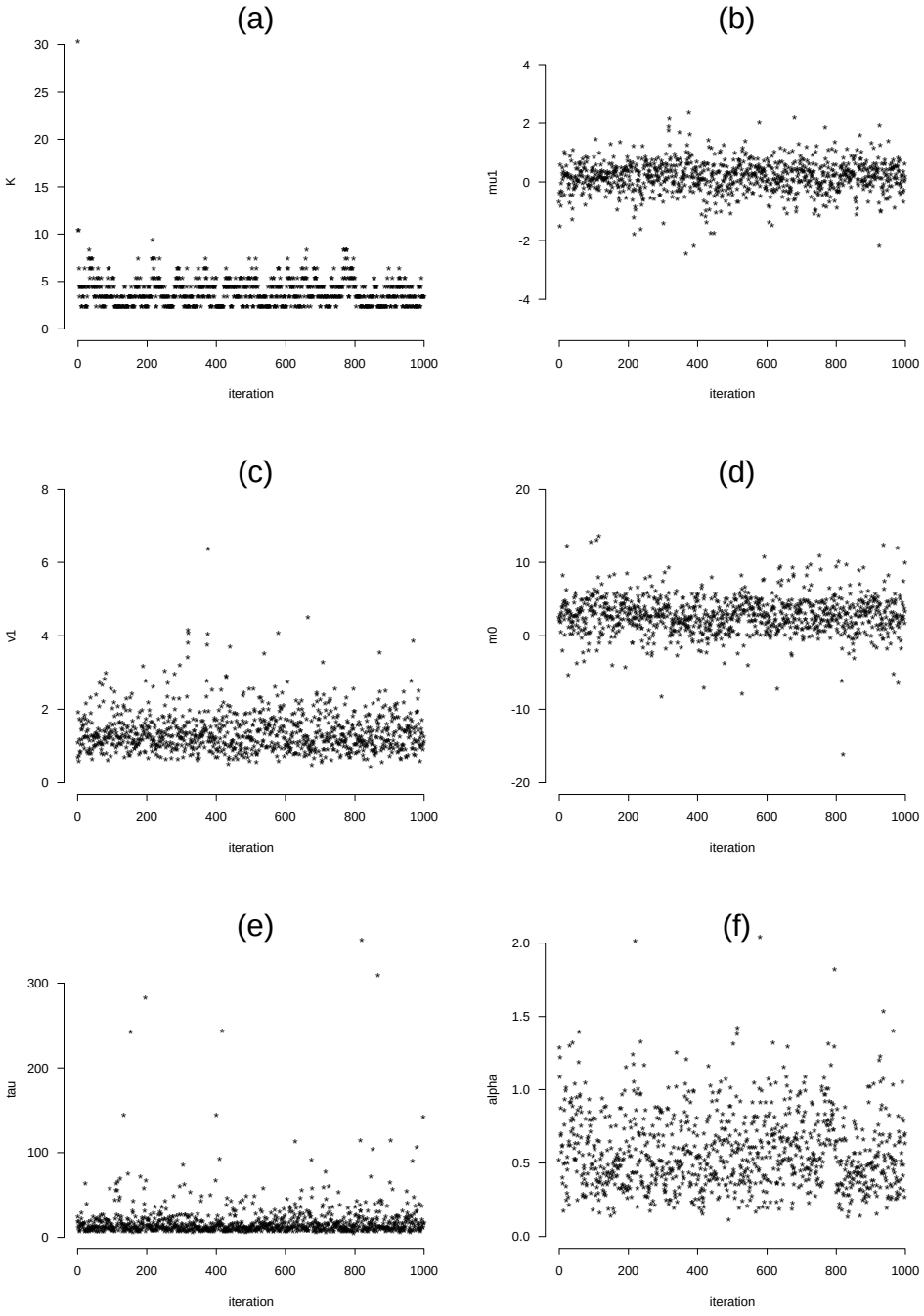


Figure 3.3: Convergence of Gibbs Sampler (II)

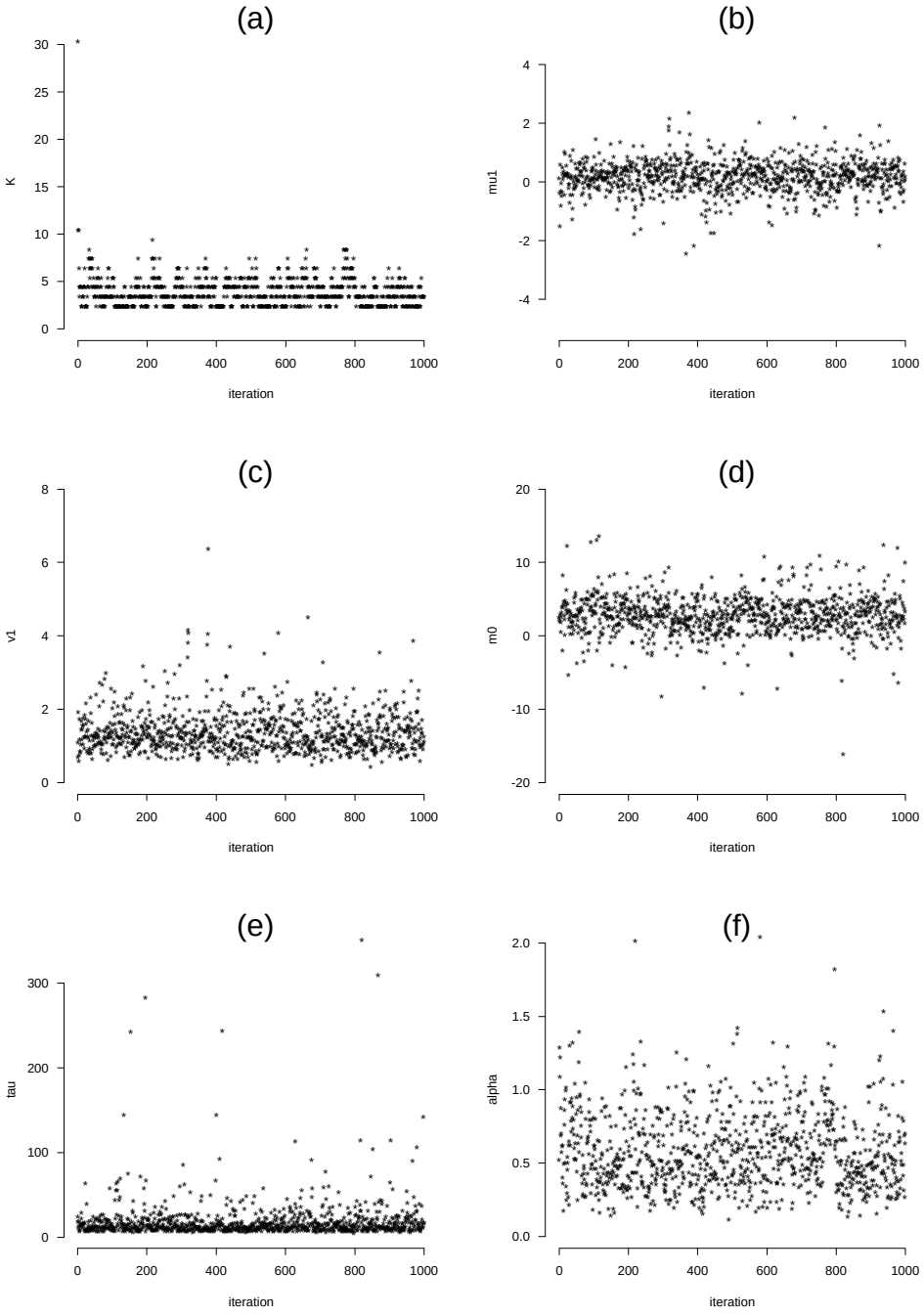


Figure 3.4: Convergence of Gibbs Sampler (III)

3.4 Proofs of Theorems

Proof of Theorem 3.2. The proof is similar to Theorem 2.2. Define $\Im = \{1, 2, \ldots, n\}$ and $\chi = [\otimes_{i=1}^{n}(\Im)_i] \times \Im$, and B_χ as the σ-field on χ. Let $A = B \in B_\chi$, ρ be a measure concentrated on B and ϵ be a positive constant. Let B_i, D_i and ρ_i as the i-th components of B, D, and ρ, respectively, where $B_i = \Im$. For all sets $D_i \subset \Im$, let

$$\rho_i(D_i) = \#\{D_i\}/n,$$

where $\#\{D_i\}$ means the number of elements in D_i. Notice that q_0 in equation (3.3) is different for each i-th component, we denote q_0 as q_{0i}. Since the q_{0i}'s are a function of c, k, let us make this relationship explicit by writing $q_{0i}(c, k)$. Since B is a finite set, we define a_i as the minimun of $q_{0i}(c, k)$ for $(c, k) \in B$. Since $q_0(c, k)$ is strictly positive for all $(c, k) \in B$, the minimum a_i is positive also. Finally, define ϵ as

$$\epsilon \;=\; n \prod_{i=1}^{n} a_i.$$

Therefore, for all $(c(0), k(0)) \in B$, and for all $D \subset B$, we have

$$\begin{aligned} \Pr((c(1), k(1)) \in D | (c(0), k(0)), Y) \;&>\; \sum_{(c,k)\in D} \prod_{i=1}^{n} a_i \; \delta_D(c, k) \\ &=\; \epsilon\rho(D). \end{aligned}$$

Thus, one condition is satisfied. With the definition of A, B, and ρ defined above, other conditions are ture since $\Pr((c(1), k(1)) \in A | (c(0), k(0)), Y) > 0$ for all $c(0), k(0)$. Theorem 3.2 follows by using the egodic theorem for Harris chains.

Proof of Theorem 3.3. The proof is very similar to Theorem 3.2. The set χ is now defined as $[\otimes_{i=1}^{n}(\Im)_i] \times \Im \times [\otimes_{i=1}^{n}\Re^+{}_i]$ to account the parameter v. Let Define B as the set

$$B = \{(c,k,v)|c_i,k \in \Im; v_i \in [b_1,b_2]; i = 1,\ldots,n\},$$

where $b_1, b_2 > 0$. Since $q_{0i} = \alpha T_{s_0}(y_i|m,M_0)/\sum_{j=1}^{k^{(i)}} n_j^{(i)} N(y_i|m_j^{(i)}, z_j^{(i)})$, the q_{0i}'s are a function of $c^{(i)}, k, v^{(i)}$, make this relationship explicit by writing $q_{0i}(c^{(i)}, k, v^{(i)})$. Let $B^{(i)}$ be defined as

$$B^{(i)} = \{(c^{(i)}, k, v^{(i)})|\exists(c_i, v_i)\ s.t.(c,k,v) \in B)\}.$$

The function $q_{0i}(c^{(i)}, k, v^{(i)})$ is a continuous function of $v^{(i)}$. For the fixed $c^{(i)}, k$, the infimum exist for $(c^{(i)}, k, v^{(i)}) \in B^{(i)}$, and since $q_{0i}(c^{(i)}, k, v^{(i)})$ is strictly positive for all $(c^{(i)}, k, v^{(i)}) \in B^{(i)}$, the infimum is strictly positive also. Since (c,k) is a finite set, so is the σ-field of (c,k). So we can define a_i as

$$a_i = \min_{c^{(i)},k \in B_1} \inf_{v^{(i)} \in B_2} q_{0i}(c^{(i)}, k, v^{(i)}),$$

where B_1 and B_2 are the components for (c,k) and v respectively. Refering to the proof of Theorem 3.2, choosing ρ as before, we can prove Theorem 3.3 immediately.

Proof of Theorem 3.4. Following the proof of Theorem 3.3, choosing $\chi = [\otimes_{i=1}^{n}\Im_i] \times \Im[\otimes_{i=1}^{n}(\Re \times \Re^+)_i]$ and defing B as

$$B = \{(c,k,\pi)|c_i,k \in \Im; v_i \in [b_1,b_2]; \mu_i \in [-a,a]; i = 1,\ldots,n\},$$

we can obtain this theorem very easily.

Proof of Theorem 3.6. The proof is very similar to the one by Escobar and West (1991). The only difference is the definitions of q_0' and $G_i(\pi_i)$ in equation

(3.10). The detailed proof is as follows: Define $\chi = \otimes_{i=1}^{n}(\Re \times \Re^{+})_i$ and B_χ as the Borel field on χ. In order to use the ergodic theorem for Harris chains, we have to specify the sets $A, B \in B_\chi$, the measure ρ concentrated on B, and the constant $\epsilon > 0$. Let $A = B$, and define B_i, D_i and ρ_i as the i-th components of B, D and ρ, respectively. Let $B_i = \Re \times (b, \infty)$ is the open interval from b to infinity, and b is some positive number. For all sets $D_i \subset \Re \times \Re^{+}$, let

$$\rho_i(D_i) = G_i[D_i \cap (\Re \times (b, \infty))],$$

where G_i is defined in equation (3.10). Also, in equation (3.10), q_0' is different for each i-th component. Define a_i as the value of q_0' evaluated at

$$\pi^{(i)} = ((y_i, b), \ldots, (y_i, b)).$$

The value of a_i is strictly positive and for all $\pi \in B, q_0' > a_i$. Finally take ϵ as

$$\epsilon = \prod_{i=1}^{n} a_i G_i[\Re \times (b, \infty)].$$

Therefore, for all $\pi(0) \in B$, and for all $D \subset B$,

$$\begin{aligned} \Pr(\pi(1) \in D | \pi(0), Y) &> \int_D \prod_{i=1}^{n} a_i G_i(d\pi_i) \\ &= \epsilon \rho(D). \end{aligned}$$

After checking the other conditions of the ergodic theorem for Harris chains, Theorem 3.6 follows.

Proof of Theorem 3.7. This proof can be obtained by a minor modification of the proof of Theorem 2.3. Replacing the (τ, m) in Theorem 2.3 by (τ, m, α), we can use Theorem 3.6 to prove Theorem 3.7. The detailed proof is not given here; see also Escobar and West (1991).

Chapter 4

Order Statistics

4.1 Introduction

Posterior samples of π are obtained from the Gibbs samplers which were described in previous chapters. Inferences about the means of components μ^* require further thought and theory. Each has a dimension of k, which itself is random, so the issue of identifiability arises. One way to alleriate the problem is to consider ordered values of the component locations; for any given k, $\mu^*_{(1)} < \mu^*_{(2)} < \ldots < \mu^*_{(k)}$ where $\mu^*_{(j)}$ denotes the j^{th} largest mean of the k means in μ^*. The relevant posterior distribution here, namely $f(k, \mu^*_{(1)}, \ldots, \mu^*_{(k)})$, is analytically intractable, especially due to the fact that the dimension k of the μ^* is uncertain. However, the Gibbs sampling approach helps enormously in the inferences about μ^* and π as well.

Each time $f(\mu^*|Y)$ is sampled we effectively draw a sample directly from the posterior $f(k, \mu_{(1)}, \ldots, \mu_{(k)}|Y)$, together with the other information identifying each of the μ_i with one μ^*_j, the corresponding number of such identical components n_j, etc. The corresponding distributions of the ordered means $\mu^*_{(j)}$ are then, those of the order statistics in independent but non-identically dis-

tributed normal samples. Theoretical and computational issues arising in the evaluation of such distributions, and their densities, appear in the next section. The Monte Carlo estimates of posteriors $f(\mu^*_{(j)}|Y)$ are averages of the densities of such conditional distributions.

Apart from very general theoretical formulae (David, 1981, p.22), little work appears to have been done on the computation of distribution and density functions of independent but not identically distributed order statistics. In contrast to the independent identically distributed case, there are considerable computational complications in the independent non-identically distributed context except in very special cases. Recently, recurrence relations for order statistics from independent non-identically distributed case have been obtained by Balakrishnan (1988), Balakrishnan, Bendre and Malik (1992). These authors relate distributions of order statistics in samples of size k to those in samples of sizes $k-1$, generalizing relationships in the standard independent identically distributed case. Though interesting, these results are not of direct use in many practical problems involving a fixed sample size. We derive new recurrence relationships among the distribution functions of order statistics of independent, but not identically distributed, random quantities. These results extend existing theory and provide computationally practicable algorithms for a variety of problems.

4.2 Recurrence Relations

Suppose random quantities $x_1, \ldots, x_k$ to be independent, and that x_i has continuous distribution function F_i, respectively, $(i = 1, \ldots, k)$. Let $x_{(1)} \leq \cdots \leq x_{(k)}$ denote the ordered values of $x_1, \ldots, x_k$, and write $F_{(i)}$ for the distribution

function of $x_{(i)}$. Further, for any $r = 1, \ldots, k$ and integers $i_1, \ldots, i_r$ such that $1 \leq i_j \leq k$ for $j = 1, \ldots, r$, we denote by I_r the set of all possible values of these r indices, writing $I_r = (i_1, \ldots, i_r)$, and by J_r the set of all possible distinct values of these r indices, writing $J_r = (i_1 \neq \cdots \neq i_r)$.

In order to facilitate the proofs of our main result, we give the following two lemmas.

Lemma 4.1 *Suppose a_i $(i = 1, \ldots, k)$ are real numbers. Define*

$$H_r = \sum_{1 \leq i_1 < \cdots < i_{r-1} \leq k} a_{i_1} \cdots a_{i_{r-1}}, \tag{4.1}$$

where $r = 2, \cdots, k$. Also we define, for $r = 1, H_1 = 1$. Then

$$(r-1)! H_r = \sum_{1 \leq i_1 \neq \cdots \neq i_{r-1} \leq k} a_{i_1} \cdots a_{i_{r-1}}, \tag{4.2}$$

where $r = 2, \cdots, k$.

Proof. For $j = 1, \ldots, k$, i_j can be $1, \ldots, k$. For fixed indices $i_1 \neq \cdots \neq i_{r-1}$ and each product $a_{i_1} \cdots a_{i_{r-1}}$, there are $(r-1)!$ combinations of the product, each of them is equal to $a_{i_1} \cdots a_{i_{r-1}}$ where $i_1 < \cdots < i_{r-1}$. Notice the indices $i_1, \ldots, i_{r-1}$ in the following equation

$$\sum_{1 \leq i_1 \neq \cdots \neq i_{r-1} \leq k} a_{i_1} \cdots a_{i_{r-1}},$$

can be exchanged without changing the result. So we change the indices by $i_1 < \cdots < i_{r-1}$. There are $(r-1)!$ combinations of terms. Each one resulting in the sum has the same result as H_r, so Lemma (4.1) follows.

Lemma 4.2 *There are recurrence relations between the* H_r $(r = 2, \ldots, k)$ *given by*

$$H_r = \frac{1}{r-1} \sum_{i=1}^{r-1} (-1)^{i+1} L_i H_{r-i}, \tag{4.3}$$

where $H_1 = 1$. *For* $r = 1, \ldots, k-1$,

$$L_r = \sum_{i=1}^{k} a_i^r. \tag{4.4}$$

Proof. Note the following sequence of identities by following the proof of Lemma (4.1):

$$\begin{aligned}
(r-2)!H_{r-1}L_1 &= \textstyle\sum_{J_{r-1}} a_{i_1} a_{i_2} \cdots a_{i_{r-2}} a_{i_{r-1}} + (r-2) \sum_{J_{r-2}} a_{i_1}^2 a_{i_2} \cdots a_{i_{r-2}} \\
&= (r-1)!H_r + (r-2) \textstyle\sum_{J_{r-2}} a_{i_1}^2 a_{i_2} \cdots a_{i_{r-2}}, \qquad (1) \\
(r-3)!H_{r-2}L_2 &= \textstyle\sum_{J_{r-2}} a_{i_1}^2 a_{i_2} \cdots a_{i_{r-3}} a_{i_{r-2}} + (r-3) \sum_{J_{r-3}} a_{i_1}^3 a_{i_2} \cdots a_{i_{r-3}}, \qquad (2) \\
(r-4)!H_{r-3}L_3 &= \textstyle\sum_{J_{r-3}} a_{i_1}^3 a_{i_2} \cdots a_{i_{r-4}} a_{i_{r-3}} + (r-4) \sum_{J_{r-4}} a_{i_1}^4 a_{i_2} \cdots a_{i_{r-4}}, \qquad (3) \\
&\vdots
\end{aligned}$$

$$\begin{aligned}
(3)!H_4L_{r-4} &= \textstyle\sum_{J_4} a_{i_1}^{r-4} a_{i_2} a_{i_3} a_{i_4} + 3 \sum_{J_3} a_{i_1}^{r-3} a_{i_2} a_{i_3}, \qquad (r-4) \\
(2)!H_3L_{r-3} &= \textstyle\sum_{J_3} a_{i_1}^{r-3} a_{i_2} a_{i_3} + 2 \sum_{i_1 \neq i_2} a_{i_1}^{r-2} a_{i_2}, \qquad (r-3) \\
H_2L_{r-2} &= \textstyle\sum_{i_1 \neq i_2} a_{i_1}^{r-2} a_{i_2} + \sum_{i_1} a_{i_1}^{r-1}, \qquad (r-2) \\
H_1L_{r-1} &= \textstyle\sum_{i_1} a_{i_1}^{r-1}. \qquad (r-1)
\end{aligned}$$

Referring to equation labels above, we note that $\text{eq}(1) - (r-2) \times \text{eq}(2) + (r-2)(r-3) \times \text{eq}(3) \cdots$, then we add them together and the following results

yield

$$(r-2)!\sum_{i=1}^{r-1}(-1)^{i+1}L_iH_{r-i} = (r-1)!H_r.$$

These lemmas are used in the key result,

Theorem 4.1 *Suppose* $x_i \sim F_i(x)$ $(i = 1, \cdots, k)$ *independently, then for* $r = 2, \ldots, k$,

$$F_{(r)}(x) = F_{(r-1)}(x) - H_r(x)\{1 - F_{(1)}(x)\}, \tag{4.5}$$

and

$$F_{(1)}(x) = 1 - \prod_{i=1}^{k}\{1 - F_i(x)\}, \tag{4.6}$$

where, for $r = 2, \ldots, k$,

$$H_r(x) = \frac{1}{r-1}\sum_{i=1}^{r-1}(-1)^{i+1}L_iH_{r-i}, \tag{4.7}$$

with $H_1(x) = 1$, *for* $r = 1, \ldots, k-1$,

$$L_r \quad = \sum_{i=1}^{k}\{\frac{F_i(x)}{1-F_i(x)}\}^r. \tag{4.8}$$

Proof. First, and as is well known, $F_{(1)}(x) = 1 - \prod_{i=1}^{k}\{1 - F_i(x)\}$.

Then, for each $r \geq 2$,

$$\begin{aligned} F_{(r)}(x) &= \Pr(X_{(r)} \leq x) \\ &= 1 - \Pr(X_{(r)} > x) \\ &= 1 - \Pr(X_{(r)} > x,\ X_{(r-1)} > x) - \Pr(X_{(r)} > x,\ X_{(r-1)} \leq x) \\ &= F_{(r-1)}(x) - \Pr(X_{(r)} > x,\ X_{(r-1)} \leq x). \end{aligned}$$

The second term on the right hand side is

$$\begin{aligned}
\Pr(X_{(r)} > x,\ X_{(r-1)} \le x) &= \sum_{1=i_1<\cdots<i_{r-1}\le k} F_{i_1}(x)\cdots F_{i_{r-1}}(x) \prod_{j\notin I_{r-1}} \{1-F_{i_j}\} \\
&= \sum_{1=i_1<\cdots<i_{r-1}\le k} (\frac{F_{i_1}}{1-F_{i_1}})\cdots(\frac{F_{i_{r-1}}}{1-F_{i_{r-1}}}) \prod_{i=1}^{k}\{1-F_i\} \\
&= H_r(x)\{1-F_{(1)}(x)\},
\end{aligned}$$

by use of Lemmas 4.1 and 4.2, this completes the proof.

Corollary 4.1 *A reverse version of Theorem 4.1 gives*

$$\begin{aligned}
F_{(r)}(x) &= F_{(r+1)}(x) + H_r^*(x)F_{(k)}(x) \qquad (k-1 \ge r \ge 1), \\
F_{(k)}(x) &= \prod_{i=1}^{k} F_i(x),
\end{aligned}$$

where

$$\begin{aligned}
H_r^*(x) &= \frac{1}{k-r}\sum_{i=1}^{k-r}(-1)^{i+1}L_{-i}H_{r+i}^* \qquad (k-1 \ge r \ge 1), \\
H_k^*(x) &= 1, \\
L_{-r} &= \sum_{i=1}^{k}\{\frac{F_i(x)}{1-F_i(x)}\}^{-r} \qquad (1 \le r \le k-1).
\end{aligned}$$

This Theorem and Corollary provide the means for recursive computation of the cumulative distribution functions. In practice, it is most efficient to combine the forward formulae of the Theorem with the reverse formulae of

the Corollary, applying the former for $r \leq n/2$ and the latter for $r \geq n/2$. Computation of density functions is not so straightforward. General recursions for the density functions lead to

$$f_{(r)}(x) = \frac{1}{(r-1)!} \sum_{i_1 \neq \cdots \neq i_r} f_{i_1}(x) F_{i_2}(x) F_{i_3}(x) \cdots F_{i_r}(x) \prod_{j \notin I_r} \{1 - F_{i_j}(x)\},$$

for $r = 1, \ldots, k$ (David, 1981, p.22). It is then obvious that density computations are much more intensive, except in very special cases. We sometimes use these recursions, though often resort to numerical methods. If the distribution functions are evaluated, using Theorem 4.1 and Corollary 4.1, over appropriate and fine enough ranges, then densities can be approximately computed by direct differencing.

Finally, note that evaluation of the terms $F_i(x)/\{1 - F_i(x)\}$, and their reciprocals, may require care in the tails if over or underflows are to be avoided. Direct development of appropriate analytic approximations (such as Mills' ratio) can be used here.

4.3 Examples

Example 4.1. Suppose x_i is exponential with $F_i(x) = 1 - e^{-\alpha_i x}$ for some $\alpha_i > 0$ $(i = 1, \ldots, k)$, then

$$\begin{aligned} F_{(1)}(x) &= 1 - e^{-k\ \overline{\alpha}\ x}, \\ F_{(r)}(x) &= F_{(r-1)}(x) - H_r(x) e^{-k\ \overline{\alpha}\ x}, \end{aligned}$$

where

$$\overline{\alpha} = \frac{\sum_{i=1}^{k} \alpha_i}{k},$$

$$
\begin{aligned}
H_r(x) &= \frac{1}{r-1}\sum_{i=1}^{r-1}(-1)^{i+1}L_i H_{r-i} \qquad (2 \le r \le k), \\
H_1(x) &= 1, \\
L_r &= \sum_{i=1}^{k}(e^{\alpha_i x}-1)^r \qquad 1 \le r \le (k-1).
\end{aligned}
$$

Example 4.2.

We have seen that inference about the means of normal components of a mixture involves computations of distributions of order statistics of a collection of normally distributed variates. For example, one set of normals drawn from an analysis in West and Cao (1992) has $k = 6$ and independent normal posteriors (for distinct means $x_i = \mu_i^*$ in that case) given by

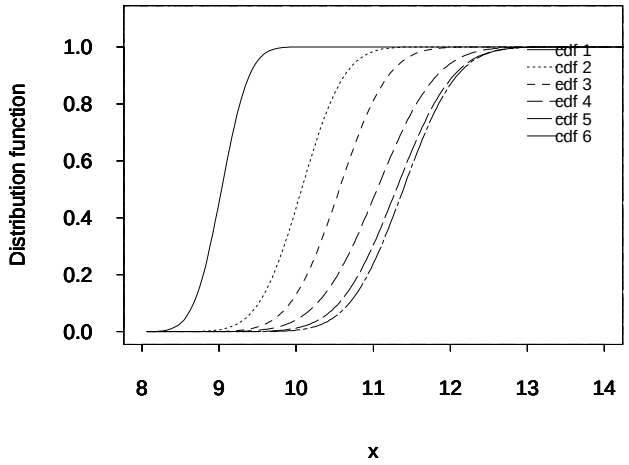
$$
\begin{aligned}
x_1 &\sim N(\ 9.03, 0.08), \quad & x_2 &\sim N(10.07, 0.19), \\
x_3 &\sim N(10.56, 0.25), \quad & x_4 &\sim N(11.05, 0.37), \\
x_5 &\sim N(11.30, 0.34), \quad & x_6 &\sim N(11.39, 0.30).
\end{aligned}
$$

Figures 4.1 (a) and 4.2 (a) display the unordered cumulative distributions and density functions respectively. Figure 4.1 (b) and 4.2 (b) display the ordered cumulative distributions and density functions respectively.

We note that, in the Monte Carlo analysis of mixture models for density estimation, computation of distribution and density functions of such ordered means are repeated over typically several thousand simulation runs, so that efficient algorithms are radically needed. Recursions developed here are implemented in C programs appropriate to the density estimation context, though

these may be adapted for applications in other problems.

(a) Unordered cdfs



(b) Ordered cdfs

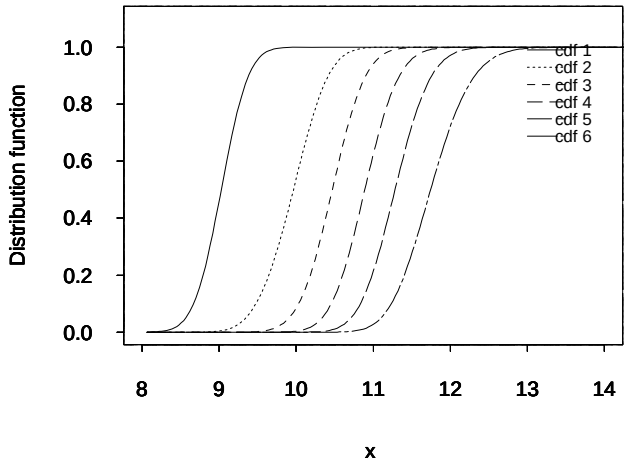


Figure 4.1: Cumulative Distribution Functions

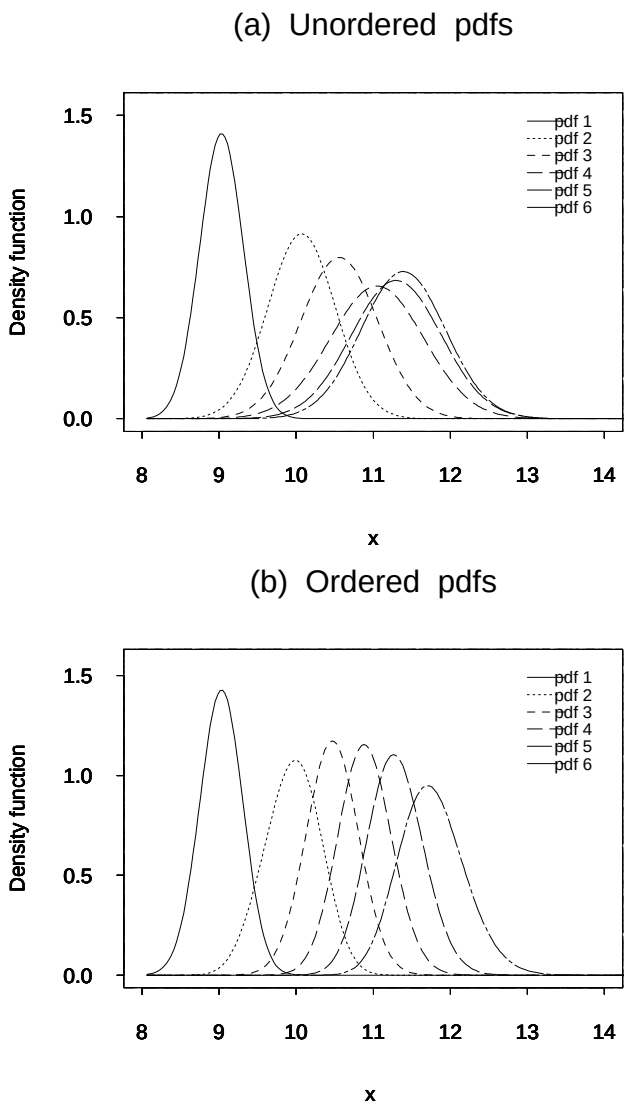


Figure 4.2: Probability Density Functions

Chapter 5

Neurophysiological Case Study

5.1 Background

Statistical analysis of mixtures has become an important aspect of neurophysiological studies according to Redman (1990). West and Cao (1992), West and Turner (1992) and Turner and West (1993) give additional references and introduce Bayesian approaches to inferences in this field. The main interest has been on modeling neural synaptic activity.

A single nerve cell, or neuron, can be viewed as a black-box that converts input electrical signals to outputs. The box is the cell body, or soma. Inputs are received through a root-like system, called dendrites, that contact other neurons at nerve junctions, called synapses. An average neuron forms about 1000 synaptic connections and receives even more. A chemical transmitter released from individual release sites on these "input" neurons induces an electrical conductance change throughout the membrane of the receiving neuron which may then "trigger" or "excite" an action or potential change on the soma. This "evoked potential response" passes through as output along the axon of the cell which ends in branch-like structures contacting further

synapses. This is how the "message" is communicated throughout the nervous system.

It is widely posited that quanta of the neurotransmitter are released from presynaptic release sites, with the probability of release either remaining constant, or varying with time or location. Each quantum produces a postsynaptic membrane potential change, whose amplitude at the recording site is also either constant or variable with time or with the location of the release site. The quantal potentials arising from different sites then generally sum linearly, as long as they remain small in relation to the driving potentials and the threshold for regenerative currents. Finally, the signal sampled via the postsynaptic microelectrode is contaminated by the background noise. If the noise amplitude is sufficiently small relative to the synaptic potential fluctuation and the number of trials recorded, then the following parameters can be estimated: the quantal amplitude, the total number of active presynaptic sites and the probability of release at each site. A quantal hypothesis which describes transmitter release occurs in basic, quantal units is also assessed. If so, overall responses will be in terms of multiples of a base or quantal unit. If transmission occurs according to a non-quantal model, or if the quantal amplitude or release probabilities are non-stationary in space or time, then some of the parameters describing these conditions can also be estimated. Kullman and Nicoll (1992), Kullmann (1989) discuss above issues and some direct generalizations such as variations of release probabilities and pre- and post- physiological influences on release characteristics.

5.2 Experiments and Data Sources

A typical experiment involves recording the changes in potential of postsynaptic membranes in response to electrical stimuli induced by small variations in ionic makeup of the neural environment, along with base readings with no stimulus. A section of tissue identified and isolated for experimentation is subject to repeat stimulation with concurrent intracellular recording of postsynaptic response potentials. Stimuli based on square wave-form currents evoke responses in terms of postsynaptic electrical potentials that typically exhibit an initial and rapid rise, then exponentially decay to base or background potentials. There are often significant fluctuations in background potentials that obscure the assessment of response characteristics – responses to spontaneous transmitter release at, typically, low levels. Recordings with no direct stimulus are made to provide prior information about base levels of background transmission that may be incorporated into inferences about the effects of non-zero stimulus. These effects are measured in several ways, one key measure being the maximum level of response attained for a given input stimulus. The typical time span between the pre- and post-stimulus periods is about 50-100 milliseconds . Basically, the levels of pre-stimulus are about zero millivolts. The peak levels of stimulus response vary widely across tissue type and according to levels of controlled stimulation. For example, levels of one data set, displayed in Figure (5.1), are between 9 and 12 millivolts.

A single experiment will be set up to record postsynaptic responses at a given neural, neuromuscular or neuro-glandular junction, the recordings being replicated possibly several hundred or more times at a given input electrical stimulus. Thus, at a single junction, recorded data consists of possibly many

replications of this sort of time series. The large amount of data generated in a single experiment is subject to drastic reduction and summarization prior to applying currently popular and accepted statistical analyses designed to investigate response mechanisms. A particular approach that has gained wide acceptance is to simply consider maximum levels of stimulus response, compared with the background, nominally zero-mean levels, and to estimate the maximum levels obtained from the time series records. This is typically done by identifying a window in time that covers most of the peak in response, and then simply averaging the response values across that window. This results in a single post-response summary measure. The maximum response level is estimated as the mean of the 50 or 60 observations in this window. For each such measurement, a pre-stimulus counterpart is evaluated by using a window of the same length located at a fixed lag from the post-stimulus region, providing a single estimate of pre-stimulus level. Across replications, the two random samples of pre- and post-stimulus response levels are assumed to be independent. The post-stimulus neural levels are assumed to be measures of the extent of response of the collective of neurons in the experimental region, and variation in these measures is expected to be informative about the scientific hypotheses of neural response. Figure (5.1) displays histograms of the pre-stimulus noise levels and post-stimulus signal data. In developing models and methods to address the neuroscientific issues, results from previous chapters have led to models involving mixtures of normal distributions for the signal plus noise data, and subsequent analyses have focused on methods of mixture deconvolution. References to the underlying neuroscience, and including currently popular and accepted methods of statistical analysis, include McClachlan (1978), Turner and Schlieckert (1990), Walmsley, Edwards and

Tracey (1987), and Wong and Redman (1980). Early development of Bayesian approaches considered here can be found in West and Cao (1992), West and Turner (1992), Turner and West (1993).

5.3 Mixture Modeling of Signal Data

Let us assume that the fluctuation of the synaptic potential in the steady state is discrete, and that the background noise in the cell is approximately normal, the signal data set of size n, say, some few hundred observations, is assumed to come from a mixture of an unknown number of normal distributions, with uncertain means and variances. The problem becomes one of resolving the parameters of the mixture, that is, the discrete amplitudes between which the underlying signal fluctuates, their mixing proportions, and the variance associated with each distribution. A simplified constraint can be added by setting the variance of each component to be equal to that of the background noise, in other words, by assuming that each discrete amplitude has no variance other than that accounted for by the background noise. Kullmann (1989) gives a detailed discussion for such models. Denote the signal data by y_i and write μ_i and v_i as their means and variances respectively; thus $(y_i|\mu_i, v_i) \sim N(y_i|\mu_i, v_i)$, $i = 1, \cdots, n$. As described in Chapters 2 and 3, a Dirichlet process prior can be used for π in the context of Bayesian nonparametric mixture modeling.

Since the nonconjugate prior analysis was advantageous from the previous discussion, our analysis adopts the normal-inverse gamma prior
$G_0(\mu, v) \equiv N(\mu|m, \tau)\ IG(v|s_0/2, V_0/2)$, and assumes that the three additional parameters m, τ, α are uncertain. Thus, the analysis involves computing the posterior distributions for $\{k, \pi, m, \tau, \alpha\}$ given the data Y. Here τ is an im-

portant parameter; a large τ leads to greater dispersion among the μ_i. A full technical discussion can be found in previous chapters, together with details of the Gibbs sampling based techniques of computation of posterior distributions for the parameters k, μ, v, τ, m and α. Posterior quantities, such as mean, variance, mode, median, of some interesting parameters can be directly estimated by Monte Carlo averages.

With respect to the quantal hypothesis of neural responses, it is the ordered values of the distinct μ_i^* that are of direct interest. Rough equal spacing of the ordered levels is consistent with the quantal hypothesis. Posterior distributions for the ordered values, and summary inferences for the spacings $\mu^*_{(j+1)} - \mu^*_{(j)}$, directly address this scientific issue. The relevant posterior distribution here, namely $p(k, \mu^*_{(1)}, \cdots, \mu^*_{(k)}|Y)$, is extremely complex, especially so, due to the fact that the dimension k of the μ^* vector is uncertain. A technical discussion about evaluating posterior distributions for the ordered values was given in Chapter 4. Recurrence relationships among the distribution functions of order statistics of independent, but not identically distributed random quantities can be used to save computational time. The Monte Carlo estimates of posterior $f(\mu^*_{(j)}|Y)$ are averages of the densities of $f(\mu^*_{(j)}|m(r), \tau(r), v(r), \alpha(r), Y)$, where $m(r), \tau(r), v(r), \alpha(r)$ are posterior samples at the r-th Gibbs iteration.

Our data analysis example is shown here. Table (5.1) gives the signal data. Figure (5.1) summarizes some basic information of the noise and signal samples. The sample size for each of the noise and signal samples is $n = 159$. The former has a sample variance of 0.047. The prior to signal data analysis for v based on these values was determined by $s_0 = 150$ and $V_0 = 150 \times 0.0467 = 7.0$. For the other parameters, we choose $\tau \sim IG(\tau|1/2, 5/2), \alpha \sim G(\alpha|4, 8)$ and a reference prior for m. About other prior distributions for τ and α , see the discussion of

this chapter.

Figure 5.2(a) displays the histogram of the signal data again. Figure 5.2(b) exhibits the histogram of the signal data plus the approximate predictive density $f(y_{n+1}|Y)$. The Monte Carlo sample size used here was 5,000 after an initial 2,000 draws to 'burn-in' from starting values. The convergence of the Gibbs sampler was monitored by using the method described in Chapter 3. Thus the predictive density $f(y_{n+1}|Y)$ is the average of 5,000 evaluations of the following distribution

$$\frac{\alpha}{\alpha+n}N(y_{n+1}|m,\tau+z)+\sum_{j=1}^{k}\frac{n_j}{\alpha+n}N(y_{n+1}|\mu_j^*,v_j^*), \tag{5.1}$$

where z is a sample from $IG(z|s_0/2,V_0/2)$. Each evaluation of (5.1) corresponds to a different value of the vector parameter $\{k,\pi^*,m,\tau,\alpha\}$ that is an approximate draw from the posterior for the parameters. The predictive cumulative density function (cdf) associated with the predictive probability density function (pdf) appears in Figure 5.2(c) which displays range bands for the cdf. The upper band (or lower band) can be obtained from the cdf plus (or minus) the standard deviation of the estimated cdf. Figure 5.2(d) plots similar range bands for the pdf which gives the posterior uncertainty corresponding to the density estimate. Figure 5.2(f) gives the average density along with the locations of the modes of the mixture components deconvolution. The heights of the lines give the estimated component weights. For example, the first posterior mean has approximate $E(\mu_{(1)}^*|Y)=9.37$, with weight 0.12; thus the first vertical line is located at 9.37, with height 0.12.

In Figure 5.2(e), the exact prior probabilities for k are displayed as '$\cdot$' while the posterior probabilities are displayed by lines. The prior distribution of k

can be easily obtained from Antoniak (1974) by using $f(k) = \int f(k|\alpha)f(\alpha)d\alpha$. $(k-1|\alpha)$ has the asymptotic conditional Possion distribution with mean $\alpha(\gamma + log(n))$ if $k = o(log(n))$ (West, 1992), where γ is the Euler's constant. Suppose α has a gamma distribution with a shape parameter a_0 and a scale parameter b_0, then we have the following asymptotic unconditional distribution:

$$f(k) = \frac{(\gamma + log(n))^x}{x!} e^{-x} \frac{\Gamma(a_0 + x)}{\Gamma(a_0) b_0^x}, \quad k = 1 + x, \quad k = o(log(n)), \tag{5.2}$$

In our case, the prior distribution of k indicates a wide range of possible values from 1 to 9, with the mode at 3. The posterior mode is evidently $k = 4$ with a probability 0.786. As discussed in West and Cao (1992), West and Turner (1992), mixture models will naturally tend to over-estimate the number of components in an attempt to produce fidelity to even minor features of non-normality in the data. As a consequence, the posterior mass on larger values of k might be informally somewhat discounted in summary inferences. Here four components are strongly indicated.

Prior and posterior distribution plots of α are displayed in Figure 5.3(a). This is consistent with the fact that, quite typically, the prior mean of α should be small in order to favor lesser number of components. Figure 5.3(b) shows prior and posterior distributions of τ. The prior of τ is $IG(\tau|0.5, 2.5)$. The posterior distribution of τ has a peaked mode. If the prior mean of τ is larger, it is more likely to have many components. One would expect the posterior of τ to concentrate on small values. The posterior density function of m is displayed in Figure 5.3(c). Recall that the prior distribution of m is a reference prior. The posterior distribution plot looks like a normal distribution with mean about 10.34, while the average value of the signal data is about 10.285.

Figure 5.4 shows the predictive mixture distribution (a) and the correspond-

Table 5.1: Signal Data

10.452	10.544	10.623	10.132	11.002
10.184	10.977	10.294	10.483	10.272
10.049	11.844	11.121	10.092	11.075
10.892	10.867	10.184	9.305	10.004
11.075	10.367	11.517	10.690	10.452
9.543	10.693	10.358	10.876	10.547
10.794	10.947	10.010	10.126	9.784
10.138	10.767	9.708	10.886	9.879
9.628	10.202	10.333	10.071	9.360
10.190	9.827	10.123	10.263	10.150
10.028	9.924	10.748	9.348	11.346
10.205	10.175	10.443	10.965	10.345
9.802	10.114	10.391	10.330	10.706
10.300	10.068	9.689	10.342	11.005
10.764	9.702	10.416	10.587	10.516
10.977	10.364	10.699	10.516	10.223
10.361	10.123	10.089	11.081	10.602
9.924	10.056	10.065	10.602	9.802
8.908	10.583	11.493	10.394	11.038
10.541	10.349	9.946	10.419	9.744
10.083	9.012	10.138	10.831	9.421
10.632	9.387	10.571	9.543	10.150
9.766	10.358	9.955	9.985	9.213
10.205	10.068	10.632	9.940	10.471
10.721	10.153	10.669	10.010	10.532
10.077	10.507	9.128	10.162	11.066
10.126	9.738	10.675	10.461	10.889
10.541	9.909	10.587	9.882	10.721
10.843	10.355	9.830	10.608	10.455
10.141	10.559	10.480	9.943	9.378
9.372	10.416	8.905	10.349	9.402
10.269	10.757	10.040	9.381	

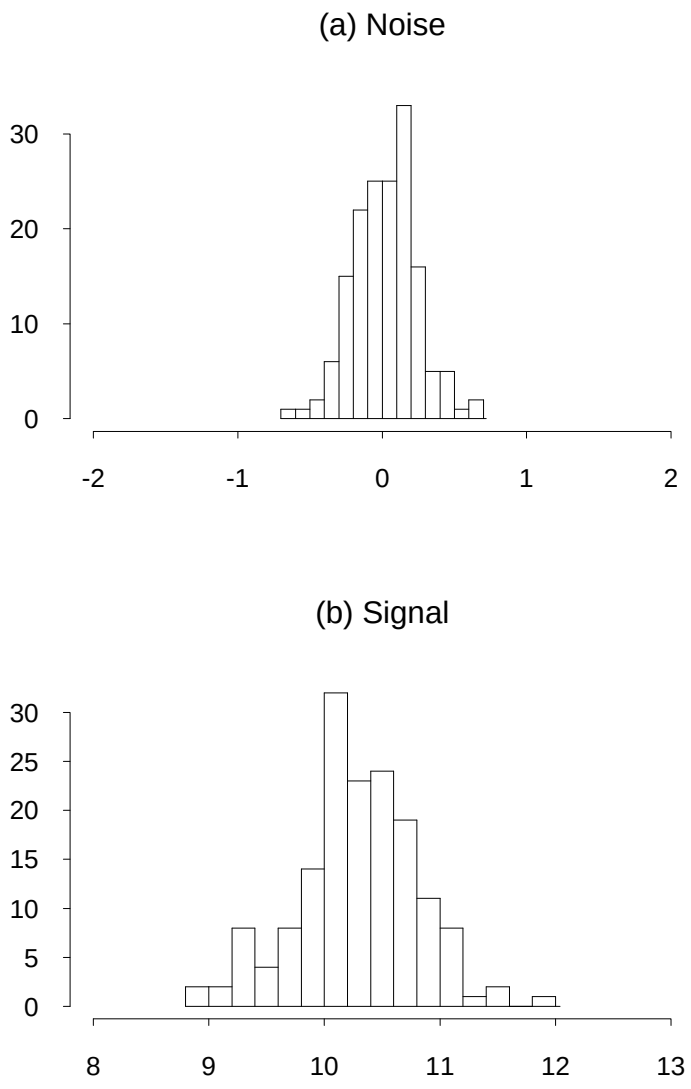


Figure 5.1: Neural Noise and Signal Histograms

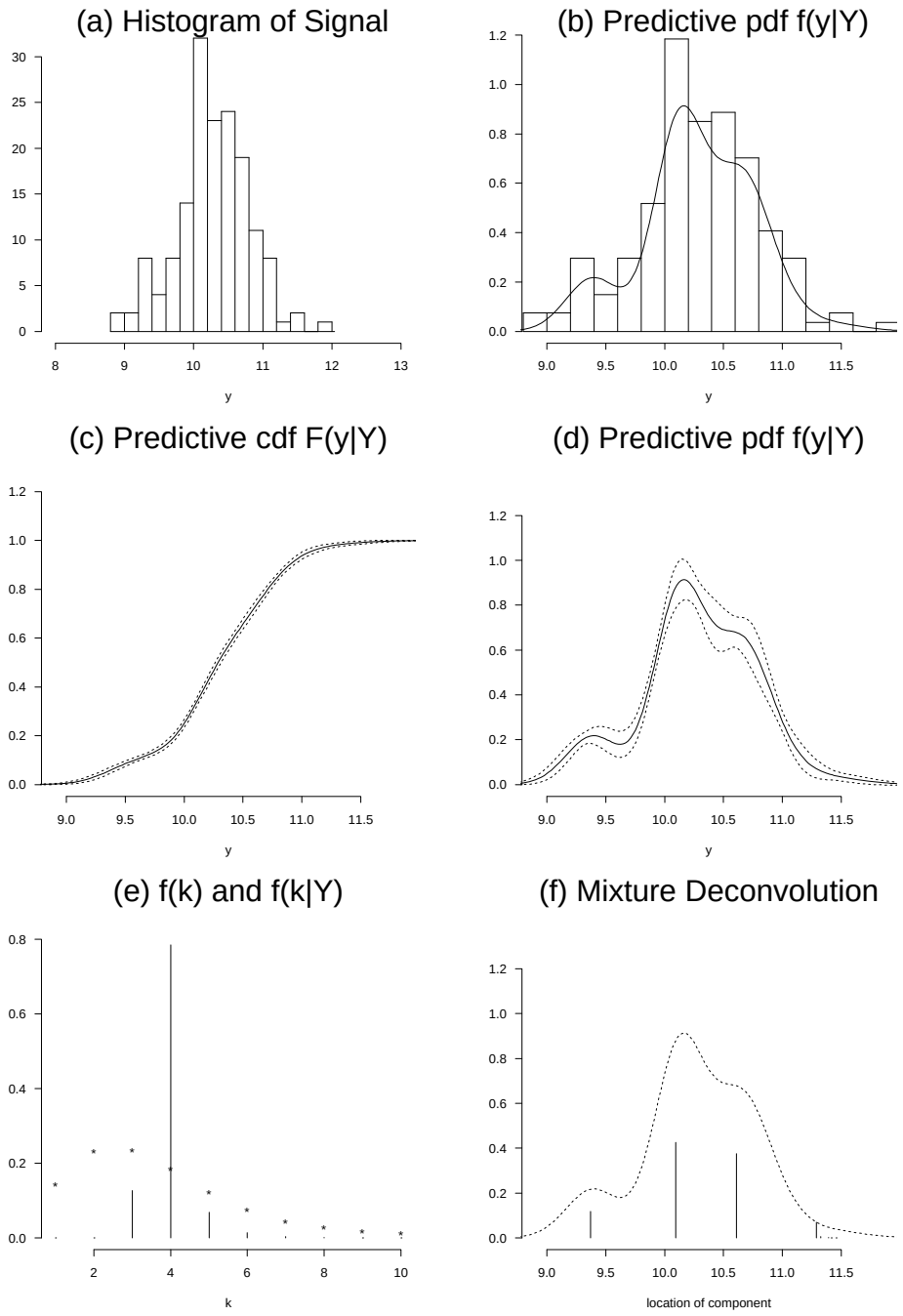


Figure 5.2: Summaries of Posterior and Predictive Distributions

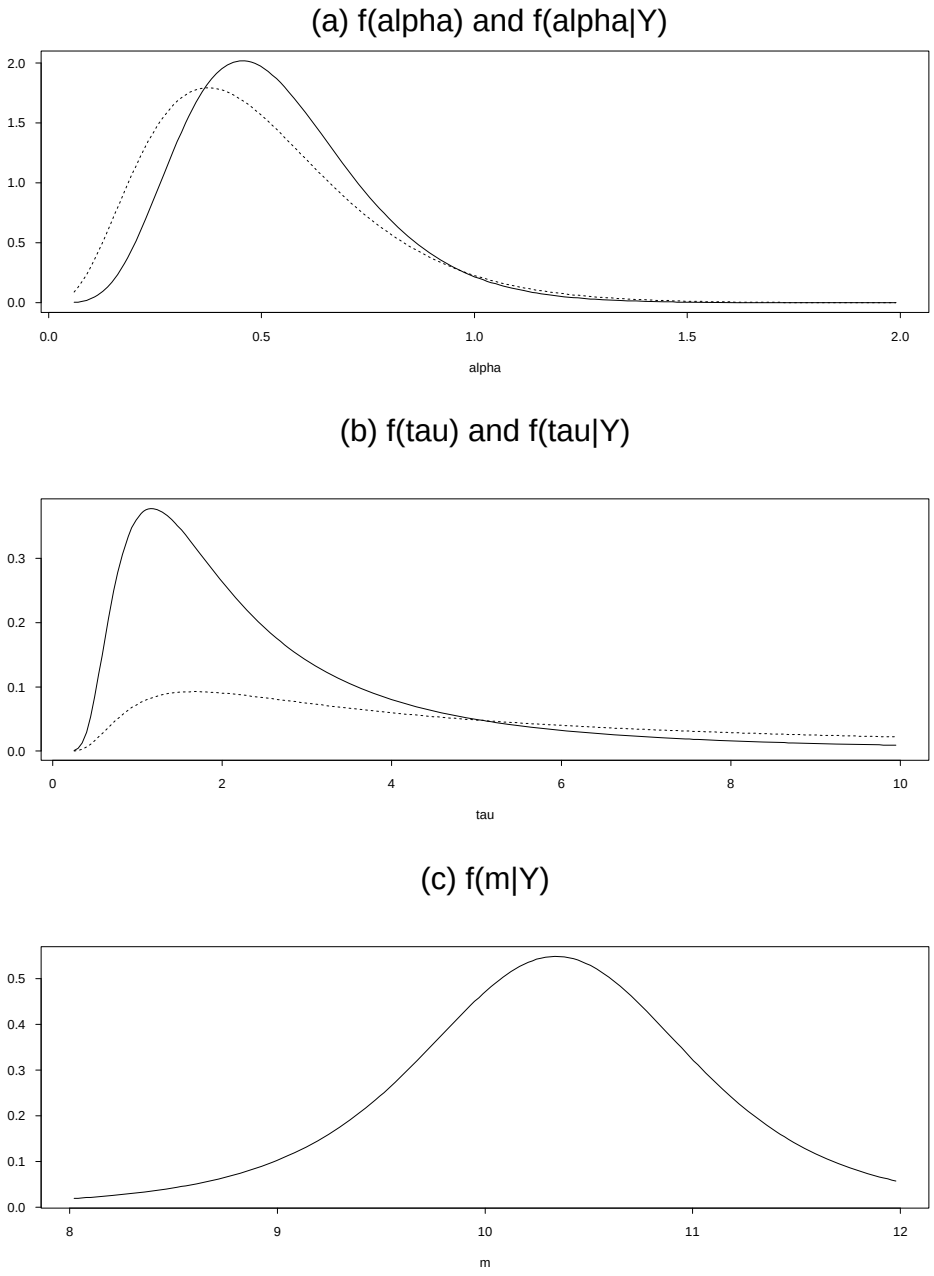


Figure 5.3: Summaries of Posterior Distributions for Hyperparameters

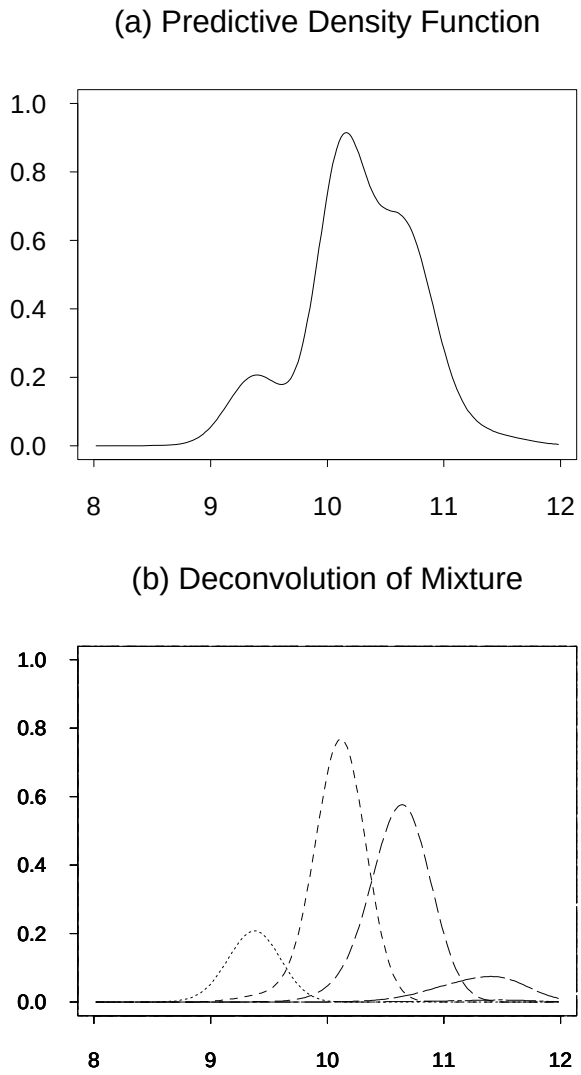


Figure 5.4: Predictive Mixture Deconvolution

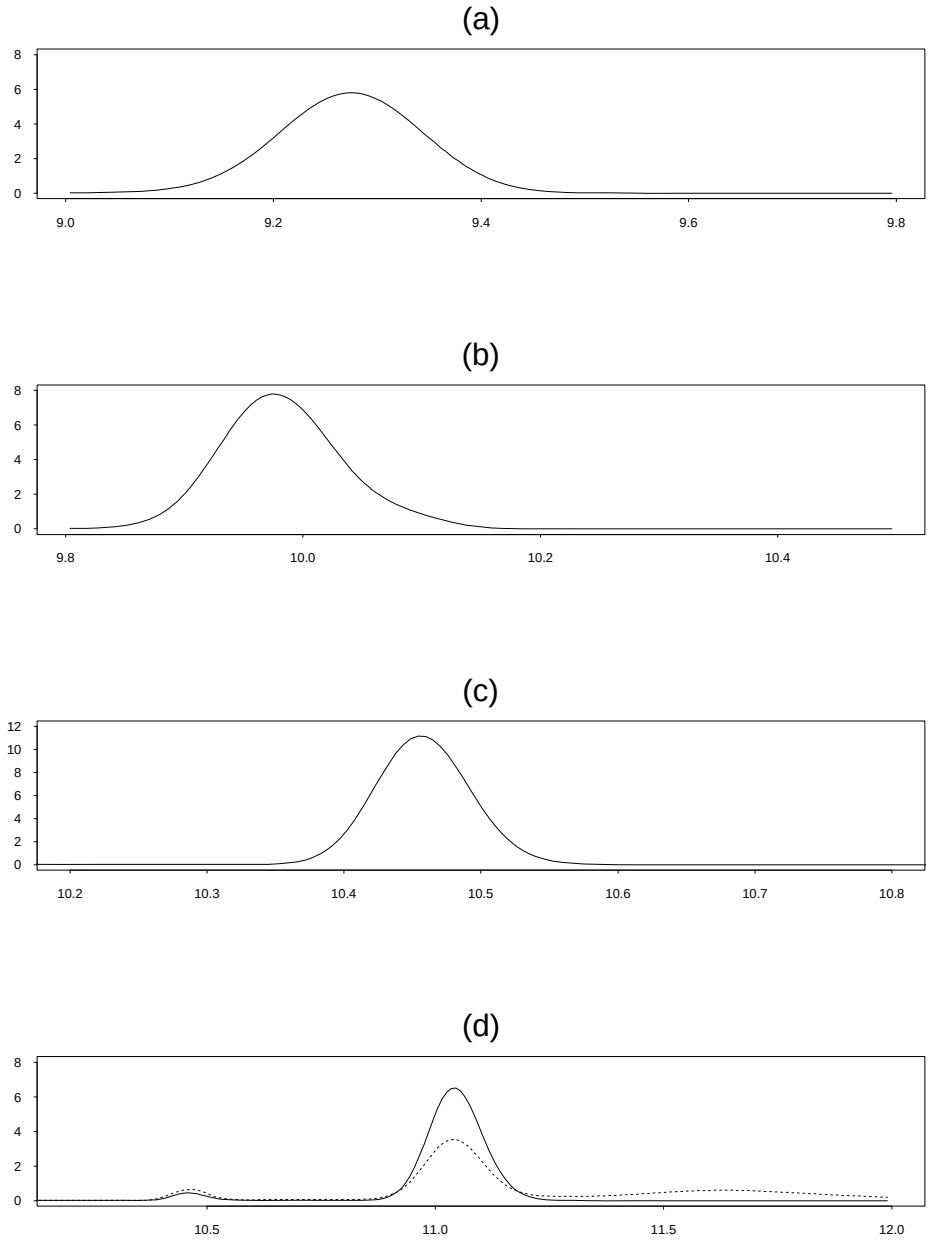


Figure 5.5: Posteriors for Ordered Component Means

Table 5.2: Quantiles for $\mu^*_{(j)}$

	$\mu_{(1)}$	$\mu_{(2)}$	$\mu_{(3)}$	$\mu_{(4)}$
quantile 10%	9.28017	9.94000	10.45500	10.91205
quantile 50%	9.37988	10.12805	10.68660	11.34889
quantile 90%	9.47683	10.12805	10.68660	11.62241

Table 5.3: Quantiles for $\delta_j = \mu^*_{(j)} - \mu^*_{(j-1)}$

	δ_2	δ_3	δ_4
quantile 10%	0.65122	0.33420	0.45705
quantile 50%	0.74818	0.55855	0.68031
quantile 90%	0.82370	0.55855	0.93580

ing deconvolution (b). The weighted average of the components displayed in Figure 5.4 (b) gives the predictive density of Figure 5.4 (a).

Figure 5.5 displays graphs relevant to inference about the component means μ^*, providing plots of the approximate posterior densities for ordered values $f(\mu^*_{(j)}|Y)$. The related theory can be found from Chapter 4. Plots (a), (b) (c) and (d) give these densities for $\mu^*_{(j)}$ $j = 1, \cdots, 5$ respectively. The first three locations are clearly distinguished. The posteriors for $\mu^*_{(4)}$ shown by the solid line in Plot (d) and $\mu^*_{(5)}$ displayed by dotted line in Plot (d) are apparently substantially similar; the posteriors here indicate that a fifth component location would be very close to the fourth. This strongly suggests a model with $k = 4$ to be reasonable. Therefore the total number of active presynaptic sites is about 4.

Table 5.2 shows the quantiles for $\mu^*_{(j)}$. Since four components are suggested from previous figures, only four components give the quantiles here.

Table 5.3 displays the quantiles for spacing δ_j which is equal to $\mu^*_{(j)} - \mu^*_{(j-1)}$. From above two tables, the quantal model can not be accepted.

5.4 Discussion

From a practical point of view, the posterior inferences involving apparently rather complex calculations can be reduced to easy routines using the posterior sampling approach. Our Bayesian analysis shares the underlying mixture models with other analysis techniques, particularly the method of maximum likelihood estimation (MLE) of the parameters of an unknown mixture distribution. Unlike other classical methods, the Bayesian analysis has many advantages, such as incorporation of prior information into analysis, evaluation of conditional distributions for number of components and posterior distributions of some interesting hyperparameters. However, the prior distributions for some parameters may affect our analysis. It is suitable to choose the reference prior for m, but we may concern the prior distributions for τ and α. Instead of choosing $\tau \sim IG(\tau|0.5, 2.5)$ and $\alpha \sim G(\alpha|4, 8)$, we do our analysis again based on $\tau \sim G(\tau|0, 0)$ and $\alpha \sim G(\alpha|2, 8)$. We will compare the posterior distributions of k and the predictive distributions of y under two different priors.

Let us denote prior (I) as $\alpha \sim G(\alpha|4, 8)$ and $\tau \sim IG(\tau|1/2, 5/2)$, and prior (II) $\alpha \sim G(\alpha|2, 8)$ and a reference prior for τ.

Figures 5.6, 5.7, 5.8 and 5.9 show the predictive density functions, posterior distributions of k, predictive mixture deconvolutions and posterior distributions for hyperparameters τ, m and α based on priors (I) and (II) respectively. Under these two prior information, we suggest that there are four components for the mixture model. All these displays of posterior provide neurologists

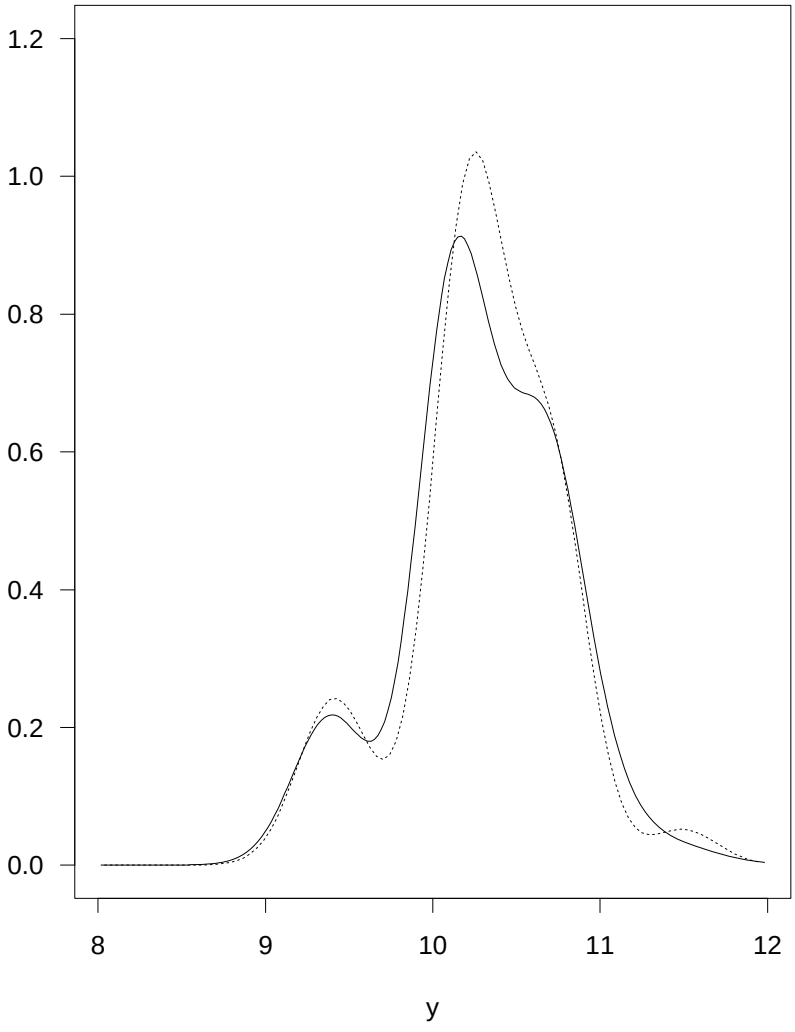
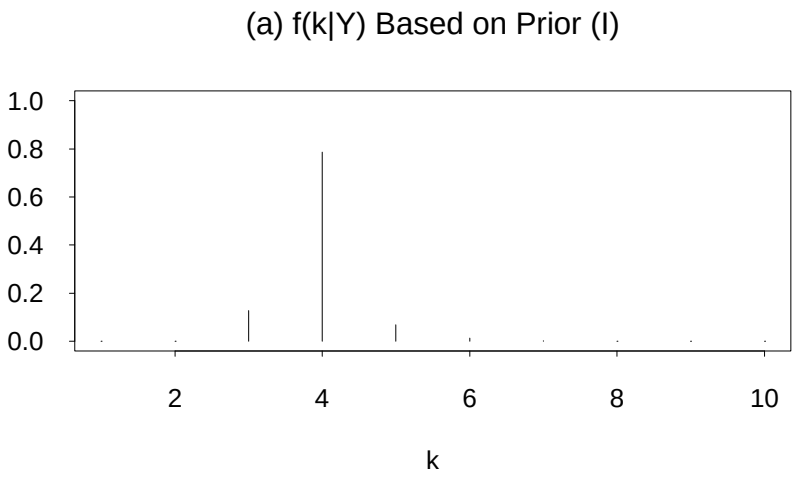


Figure 5.6: Comparisons for Predictive Density Functions

—— predictive density function based on prior (I)
..... predictive density function based on prior (II)



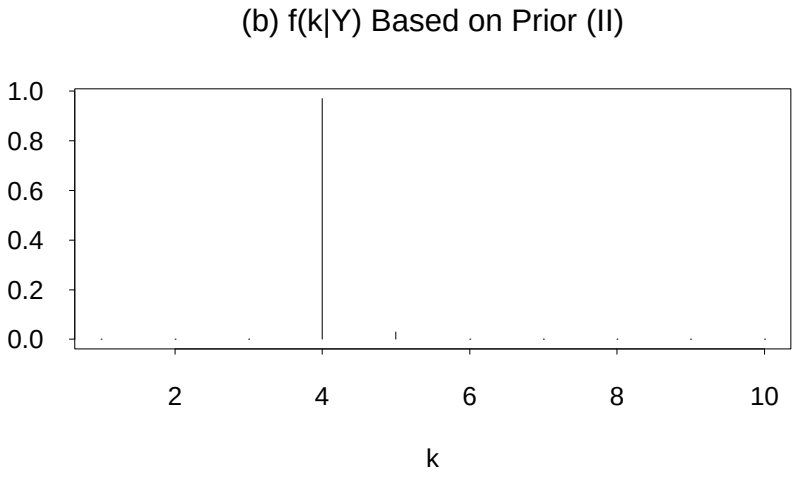


Figure 5.7: Comparisons for Posterior Distributions of k

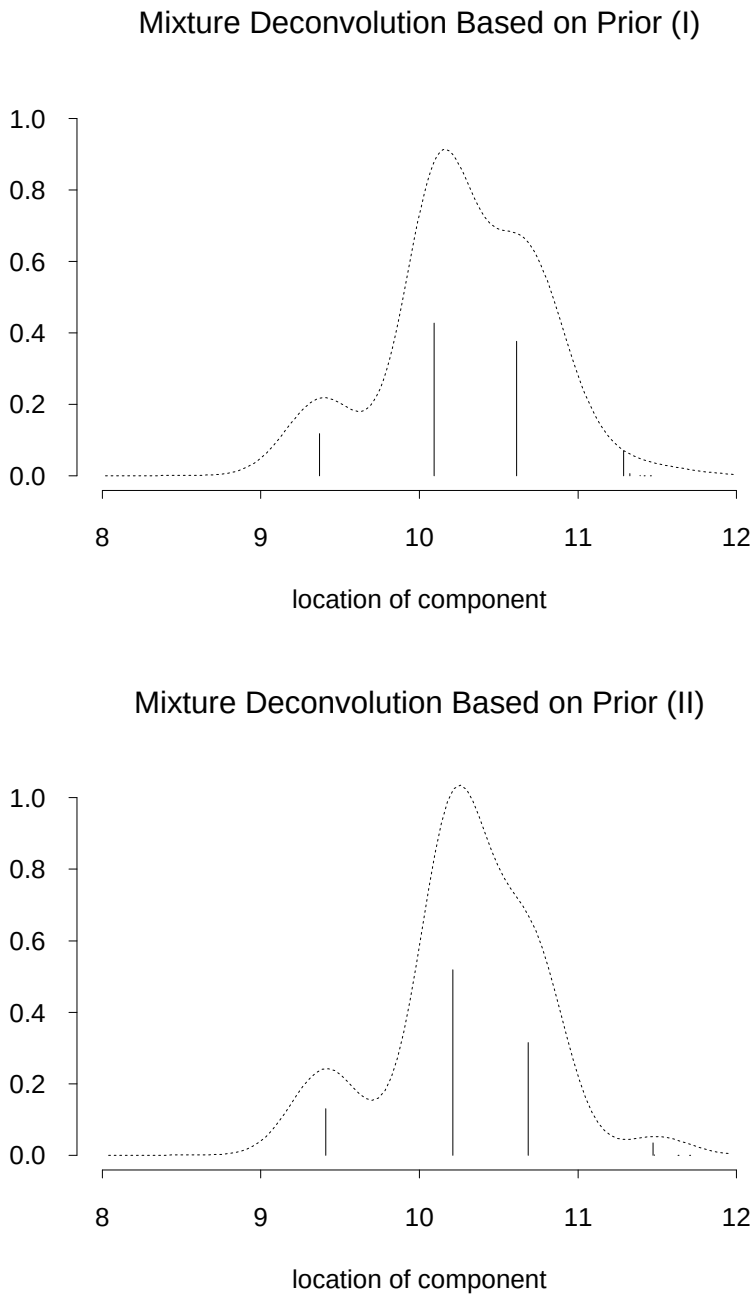


Figure 5.8: Comparisons for Predictive Mixture Deconvolutions

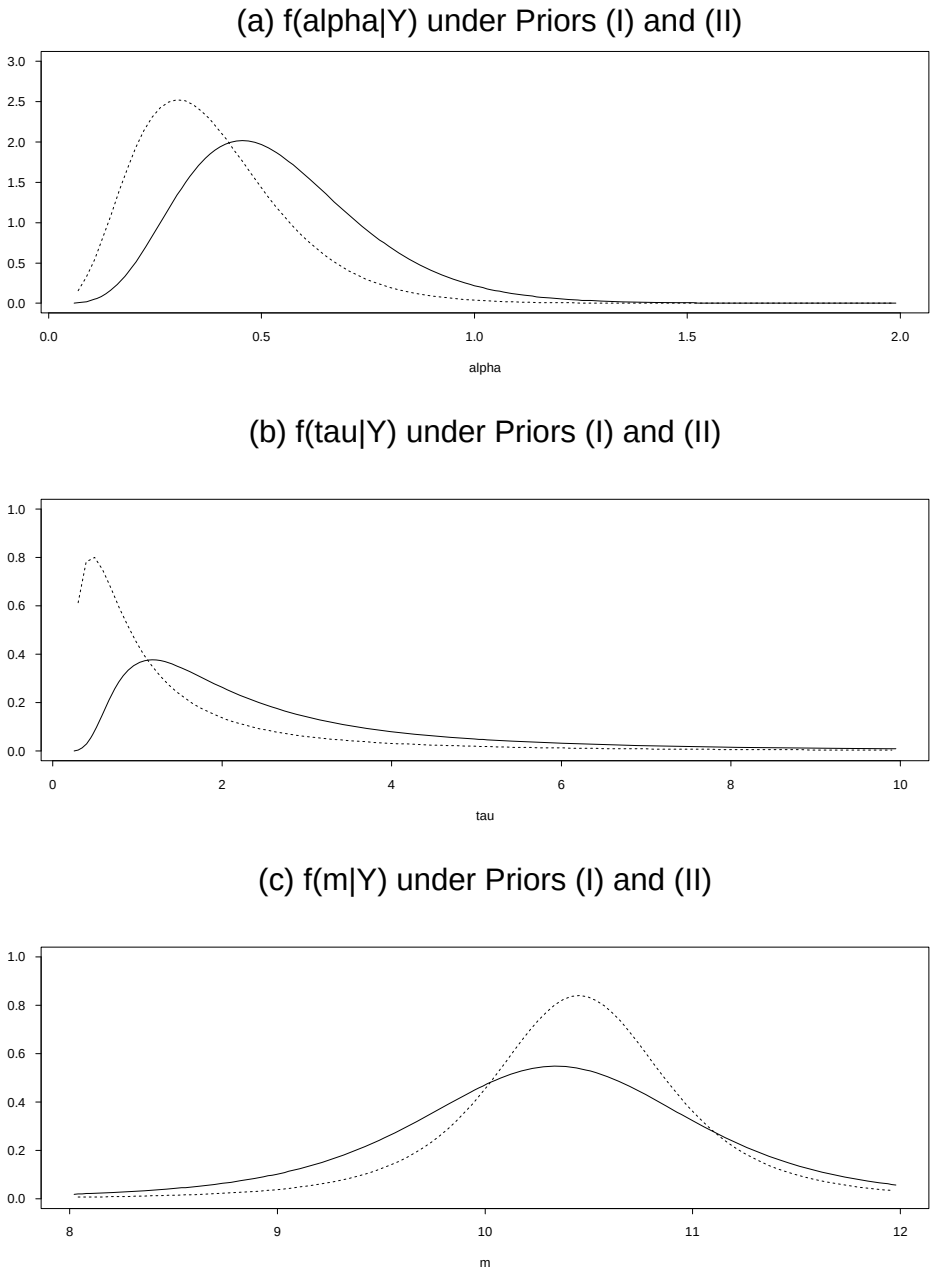


Figure 5.9: Comparisons for Posterior Distributions of Hyperparameters

—— posterior distributions based on prior (I)
..... posterior distributions based on prior (II)

with basic tools to explore changes in neural response mechanisms induced by changes in neural stimulation or physical changes in neural tissue.

Chapter 6

Bayesian Density Estimation for Grouped Data

We have studied Bayesian density estimation based on complete data information which was described in Chapter 3. However, sometimes, we only have incomplete data information such as binned data. In this chapter, we will develop the model which was discussed in Chapter 3 for grouped data that includes the possibility of censoring.

6.1 Introduction

Let us assume that observations can be grouped into N non-overlapping intervals of the form $[a, b)$. This sampling scheme generalizes the binned data setup of Kullmann (1992).

Johnson and Christensen (1986) study Bayesian nonparametric survival analysis for grouped data. Ferguson and Phadia (1979) consider Bayesian nonparametric estimation based on censored data. Titterington and Mill (1983) describe the use of non-parametric methods to estimate multivariate density functions for incomplete continuous data. We show that the posterior infer-

ences for the grouped data can be easily obtained by treating missing data as 'parameters' in the Gibbs sampling scheme. As with other treatments of grouped data it will be assumed that the data are missing "at random" and "noninformative"; see Titterington and Mill (1983).

6.2 Grouped Data Models

Suppose that, instead of observing the actual data Y, we only have following information I. Unknown data are from N intervals $[a_1, b_1), \ldots, [a_N, b_N)$, where $a_i \leq b_i \leq a_{i+1},\ i = 1, \ldots, N-1$, and $a_i,\ b_i,\ N$ are known fixed constants. There are n_i data points in the i-th interval for $i = 1, \ldots, N$. Let us assume that unobserved data $y_1, y_2, \ldots, y_{n_1}$ are from interval $[a_1, b_1)$, ..., $y_{n_{N-1}+1}, \ldots, y_n$ are from interval $[a_N, b_N)$, with $n = n_1 + \ldots + n_N$. We call I the binning mechanism. Assuming Y has a parametric distribution $f(Y|\pi)$ and π, a prior defined by $f(\pi)$. The Bayesian model is given by

$$f(I|\pi)f(\pi), \tag{6.1}$$

where $f(I|\pi)$ is induced by $f(Y|\pi)$. According to Bayes' theorem,

$$f(\pi_i|I, \pi_j, j \neq i) \propto f(I|\pi)f(\pi), \tag{6.2}$$

which is not easy to use because $f(I|\pi)$ is not an explicit expression in terms of π. If we treat the unknown data Y as parameters, the Bayesian model then becomes

$$f(I|Y)f(Y|\pi)f(\pi), \tag{6.3}$$

and the full conditionals are given by

$$f(\pi_i|I, Y, \pi_j, j \neq i) \propto f(Y|\pi)f(\pi), \tag{6.4}$$

together with the conditionals for y_j given $y_i,\ i \neq j$, derived from

$$f(Y|I,\pi) \propto f(I|Y)f(Y|\pi). \tag{6.5}$$

Sampling is now easier. The following sections discuss some specific cases dealing with the density estimation for the problems of binned data.

6.3 Normal Mixture Modeling

As described in Chapter 3, we assume that $y_1,\ldots,y_n$ are conditionally independent and normally distributed, $(y_i|\pi_i) \sim N(\mu_i, v_i),\ \pi_i = (y_i|\mu_i, v_i)$, where $\pi_1,\ldots,\pi_n$ are samples from a Dirichlet process $D(\alpha, G_0(\cdot))$. See also Chapters 2 and 3. Then

$$(Y|\pi) = \prod_{i=1}^{n} N(y_i|\mu_i,\ v_i), \tag{6.6}$$

where π denotes a vector of parameters $(\pi_1,\ldots,\pi_n)$. Therefore, equation (6.5) implies that we can sample y_j from $N(y_j|\mu_j, v_j)$ restricted to $[a_i, b_i)$ if $\sum_{l=1}^{i-1} n_l < j \leq \sum_{l=1}^{i} n_l$ for $j = 1,\ldots,n$.

Based on the results of Chapter 3, we can implement the Gibbs sampling scheme as before, except for sampling y_i; $i = 1,\ldots,n$. A random variable y_j of the truncated univariate normal $N(y_j|\mu_j, v_j)$ with $a_i \leq y_j \leq b_i$ can be obtained by generating $U \sim$ uniform $U(0,1)$ and setting

$$y_j = \mu_j + \sqrt{v_j}\Phi^{-1}[\Phi(\frac{a_i-\mu_j}{\sqrt{v_j}}) + U\{\Phi(\frac{b_i-\mu_j}{\sqrt{v_j}}) - \Phi(\frac{a_i-\mu_j}{\sqrt{v_j}})\}], \tag{6.7}$$

where Φ is the $N(0,1)$ cdf, and Φ^{-1} is the inverse $N(0,1)$ cdf. (e.g. see Devroye, 1986, p.39). Similar to the Theorems 3.4 and 3.5, the convergence of the Gibbs sampler can be obtained and omitted here.

Table 6.1: Binned Data

i	$i-th$ interval	n_i	i	$i-th$ interval	n_i
1	[-5.0, -4.5)	1	21	[5.0, 5.5)	24
2	[-4.5, -4.0)	1	22	[5.5, 6.0)	36
3	[-4.0, -3.5)	0	23	[6.0, 6.5)	26
4	[-3.5, -3.0)	1	24	[6.5, 7.0)	34
5	[-3.0, -2.5)	0	25	[7.0, 7.5)	29
6	[-2.5, -2.0)	1	26	[7.5, 8.0)	23
7	[-2.0, -1.5)	2	27	[8.0, 8.5)	17
8	[-1.5, -1.0)	2	28	[8.5, 9.0)	18
9	[-1.0, -0.5)	5	29	[9.0, 9.5)	24
10	[-0.5, 0.0)	8	30	[9.5, 10.0)	12
11	[0.0, 0.5)	4	31	[10.0, 10.5)	15
12	[0.5, 1.0)	6	32	[10.5, 11.0)	13
13	[1.0, 1.5)	13	33	[11.0, 11.5)	4
14	[1.5, 2.0)	7	34	[11.5, 12.0)	7
15	[2.0, 2.5)	19	35	[12.0, 12.5)	8
16	[2.5, 3.0)	25	36	[12.5, 13.0)	3
17	[3.0, 3.5)	23	37	[13.0, 13.5)	4
18	[3.5, 4.0)	33	38	[13.5, 14.0)	1
19	[4.0, 4.5)	23	39	[14.0, 14.5)	1
20	[4.5, 5.0)	25	40	[14.5, 15.0)	2

As a simple simulation analysis, we consider a model based on the conjugate baseline prior from Chapter 2. Kullmann (1992) uses the following idea for simulation analysis. Suppose we have 500 data points y_j's from a mixture of distributions $\sum_{i=0}^{4} w_i N(y_j|\mu_i, v_i)$ with $\mu_i = 3i$, $v_i = 0.25i + 1$ and $w_i = \binom{4}{i} 0.5^4$, for $i = 0, \ldots, 4$. We can group them into a histogram as shown in Figure 6.1(a). Now we can ignore the data and only keep the information I as displayed in Table (6.1).

Our goal is to estimate the density function based on the above information. The conjugate baseline prior is assumed by $\mu \sim N(\mu|m, \tau v)$ and

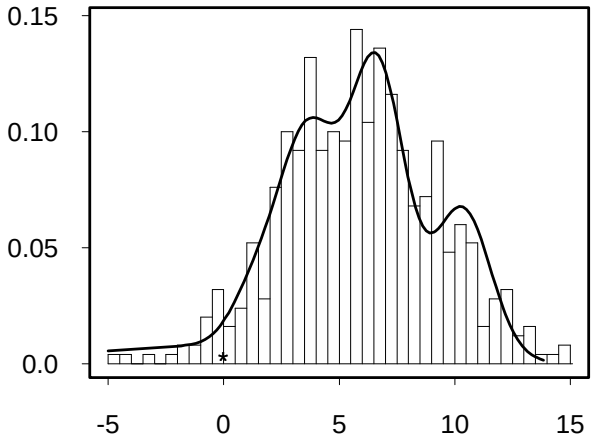
$v \sim IG(v|s_0/2, V_0/2)$. We also assume prior distributions for some hyperparameters: $\tau \sim IG(\tau|t_0/2, R_0/2)$, $\alpha \sim G(\alpha|a_0, b_0)$ and m has a reference prior. We set $t_0 = R_0 = 0.0$ such that τ has a noninformative prior. We choose $s_0 = 3.0$ and $V_0 = 1.0$. Due to a large data size and few components, we choose $a_0 = 10$ and $b_0 = 5000$.

A nonparametric Bayes' estimate of the predictive density function for binned data is the dotted line in Figure 6.1(a). The Monte Carlo sample size used here was 5,000 after an initial 2,000 'burn-in' cycles. Figure 6.1(a) displays the density estimation for the exact data Y since it is the best analysis to make a comparison. We use the exact data analysis to compare grouped data analysis to know how much information is lost. The algorithm can be found in Chapter 3. Figure 6.1(b) shows the density estimation for binned data. Since we treat data Y as parameters and randomly sample Y in certain intervals, some peaks of the estimated density in Figure 6.1(a) disappear. This is because we have lesser information from the grouped data than the exact data.

6.4 Classical Methods vs. Bayesian Methods

In this section, we will compare a classical method with the above Bayesian approach. We noted some classical methods and Bayesian methods for mixture analysis in Chapter 1. Classical methods to deal with grouped data have been studied by Kullmann and Nicoll (1992), Kullmann (1992), who provide references. Kullmann (1992) provides a maximum entropy noise deconvolution (MEND) which is an ad hoc approach to solve the mixture problem. The approach operates as follows: The data are binned in N intervals. Each of

(a) Dirichlet Process Prior for Exact Data



(b) Dirichlet Process Prior for Binned Data

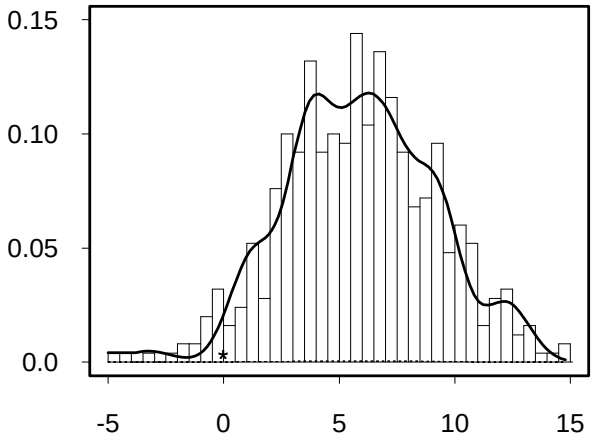


Figure 6.1: Simulation Analysis for Grouped Data

them has the same bin width. There are n_i data points in the i-th interval. Let k_0 denote the interval at 0 value and g indicate the known noise distribution. The required solution is a probability density function f ($f_j,\ j = 1, \ldots, N$) such that the convolution of f and g gives the maximum likelihood fit to the histogram, while simultaneously maximizing the entropy of f. The MEND method gives a recursion to calculate the f,

$$f_j = \left(\frac{1}{\sum_i n_i} \sum_i n_i \frac{f_j g_{i-j+k_0}}{\sum_j f_j g_{i-j+k_0}} \right)^{\lambda} \left(\frac{1}{N} \right)^{1-\lambda}, \tag{6.8}$$

where $f_i,\ g_i$ are evaluated at the center of the i-th interval, λ can take values between 0 and 1. If $\lambda = 1$, equation (6.8) becomes the recursion of the maximum likelihood estimation. If $\lambda = 0$, equation (6.8) gives the maximum entropy solution. If $0 < \lambda < 1$, the MEND solution is a trade-off between the maximum likelihood estimation and the entropy solution based on geometric average. The proof of this result is deferred to the end of this chapter.

Suppose $(f_1, \ldots, f_N)$ is from a Dirichlet distribution with parameter $(\lambda/N, \ldots, \lambda/N)$. It is straightforward to deduce the posterior distribution of $(f_1, \ldots, f_N)$ as follows:

$$(f_1, \ldots, f_N | I) \propto \prod_i (\sum_j f_j g_{i-j+k_0})^{n_i} \prod_j f_j^{(\frac{\lambda}{N}-1)}, \tag{6.9}$$

subject to $\sum_j f_j = 1$. It is not difficult to verify that the posterior modes of f_j have the following recursion,

$$f_j = \frac{\sum_i n_i \frac{f_j g_{i-j+k_0}}{\sum_j f_j g_{i-j+k_0}} + (\lambda/N - 1)}{\sum_i n_i + (\lambda/N - 1)N}. \tag{6.10}$$

If $\lambda = N$, equation (6.10) becomes the recursion of the maximum likelihood estimation. If $\lambda \to \infty$, equation (6.10) gives the entropy solution $1/N$.

Otherwise, the right hand side of the above equation is the weighted average of the MLE (with weight $\frac{\sum_i n_i}{\sum_i n_i+(\lambda/N-1)N}$) and the entropy solution $1/N$ (with weight $\frac{(\lambda/N-1)}{\sum_i n_i+(\lambda/N-1)N}$).

The common characteristic of Equations (6.8) and (6.10) is a trade-off between the MLE and the entropy solution. However, equation (6.8) is obtained by using the geometric average while the equation (6.10) is given by using the arithmetic average.

In order to make a comparison, we use Kullmann's simulation model. Let g be the standard normal distribution. The information for n_i and N can be obtained from Table (6.1). According to Equation (6.8), we have the MEND solution in Figure 6.2(a). Start with a flat distribution and an intermediate value of λ, say, 0.5. The recursion is applied until it passes a convergence threshold; the solution f is normalized to sum to 1, reconvolved with the noise distribution g, and compared to the amplitude histogram with a goodness-of-fit test. If this fails at a predetermined level of significance, λ is increased, and the recursion is applied again until the goodness-of-fit criterion is satisfied. In this case, a χ^2 test is used by Kullman to assess the goodness-of-fit. Figure 6.2(b) exhibits the Bayesian parametric solution based on the Dirichlet distribution prior. We choose a prior as $\lambda/N = 2$. There are two peaks in the plot. Other choices of λ/N might be considered here.

6.5 Discussion

We have presented a general algorithm of density estimation for grouped data. Such an algorithm is very easily implemented. For the simulated data, Bayesian nonparametric and parametric models are compared with the classical 'MEND'

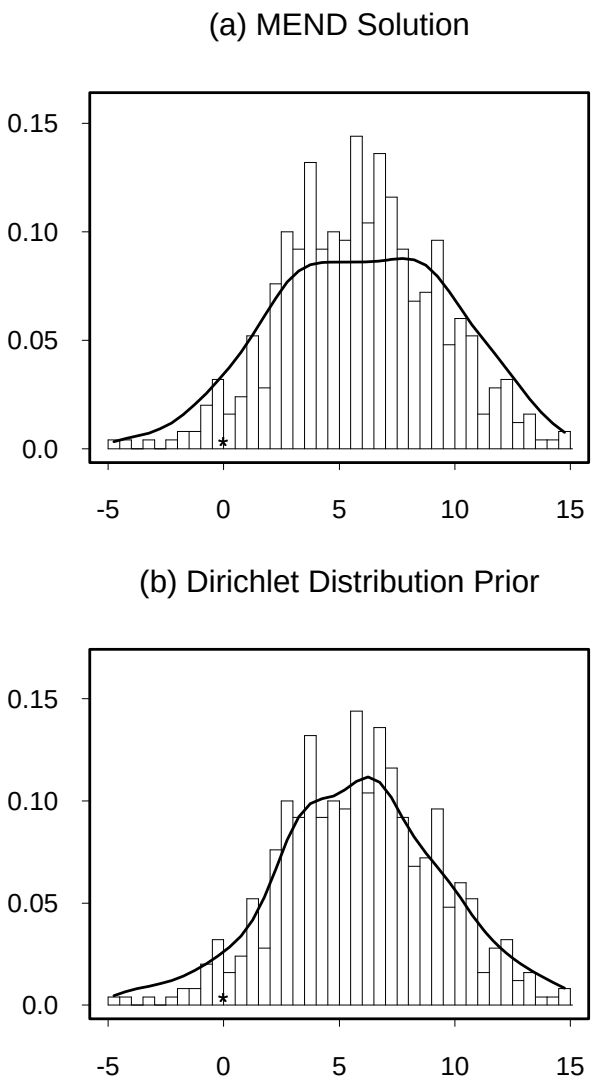


Figure 6.2: Classical Methods vs. Bayesian Methods for Grouped Data

method. The model used for Bayesian nonparametric analysis here is based on a conjugate baseline prior. As we pointed out in Chapter 3, a nonjugate baseline prior is much better than a conjugate baseline prior in the context of density estimation. Of course, we should develop such a new model in the near future.

6.6 Appendix

6.6.1 Proof of Equation (6.8)

The maximum likelihood estimation of f_j for $j = 1, \cdots, N$ subject to $\sum f_j = 1$ can be obtained by maximizing a following equation

$$\sum_i n_i log \sum_j f_j g_{i-j+k_0} + B_1(\sum_j f_j - 1), \tag{6.11}$$

where B_1 is a constant. Let the derivative of f_j from above equation equal zero, we have

$$\sum_i n_i \frac{g_{i-j+k_0}}{\sum_j f_j g_{i-j+k_0}} + B_1 = 0. \tag{6.12}$$

Using above equation, we have

$$f_j = -\frac{1}{B_1} \sum_i n_i \frac{f_j g_{i-j+k_0}}{\sum_j f_j g_{i-j+k_0}}. \tag{6.13}$$

Since $\sum_j f_j = 1$, $B_1 = -\sum_i n_i$. Therefore the MLE of f_j can be calculated by

$$f_j = \frac{1}{\sum_i n_i} \sum_i n_i \frac{f_j g_{i-j+k_0}}{\sum_j f_j g_{i-j+k_0}}. \tag{6.14}$$

Once we get the MLE, Equation (6.8) yields immediately.

6.6.2 Proof of Equation (6.10)

Taking the *log* operation of Equation (5.9), we want to maximize the goal function.

$$G(f) = \sum_i n_i log(\sum_j f_j g_{i-j+k_0}) + \sum_j (\frac{\lambda}{N} - 1) log(f_j) - B_2(\sum_j f_j - 1). \quad (6.15)$$

Let $dG(f)/df_j = 0$, the following results are given for $j = 1, \cdots, N$.

$$\sum_i n_i \frac{g_{i-j+k_0}}{\sum_j f_j g_{i-j+k_0}} + \sum_j \frac{\lambda/N - 1}{f_j} - B_2 = 0, \quad (6.16)$$

thus $B_2 = \sum_i n_i + (\lambda/N - 1)N$. Equation (6.10) follows.

Chapter 7

Bayesian Analysis of Mixtures of Mixtures

Suppose data values y_j arising as $y_j = \mu_j + x_j$ where the x_j's come from a Dirichlet process mixture of normals and the μ_j's arise from another independent Dirichlet process. This induces *a Dirichlet mixture of mixtures of normals*, whose analysis may be developed using Gibbs sampling.

7.1 Introduction

Mixtures of mixtures of distributions may be used as models in some practical situations where data y_j can be viewed as $\mu_j + x_j$, when the μ_j's take values from discrete states and the x_j's are from a mixture of distributions. For example, in the neurophysiological context, suppose the analysis of fluctuations in the amplitudes of excitatory post-synaptic potentials (EPSP) has discrete states μ_j, because of random perturbations (x_j), the observed measurements take on value $y_j = \mu_j + x_j$. In some cases, the term x_j can be assumed from a single normal distribution to simplify the statistical analysis, but in many cases, this may not be adequate.

Techniques for estimation of mixture models of normal distribution have been reviewed in previous chapters. In the rest of sections we describe the Bayesian models to develop a new set of estimates based on mixtures of mixtures. Simulation analysis is first presented. Then a data set provided by the Duke University Medical Center is used to obtain predictive distributions for noise x_j, signal y_j and to make other inferences.

7.2 Mixtures of Mixtures Modeling

To simplify the notation, we will use the subscripts 0 and 1 to indicate the parameters of noise x_j and signal y_j respectively. Let x_i be i-th noise observation, and y_j be j-th signal observation, where $i = 1, \ldots, n_0$ and $j = 1, \ldots, n_1$. We denote X as $(x_1, \ldots, x_{n_0})$, and Y as $(y_1, \ldots, y_{n_1})$. The signal data y_j can be modeled as $\mu_j + x_{j+n_0}$ where the x_{j+n_0} is the unknown noise observation and $j = 1, \ldots, n_1$. Let $n = n_0 + n_1$. Then for $i = 1, \ldots, n$, the noise data x_i can be modeled as arising from a Dirichlet process mixture of normals. Explicitly, x_i is from a conditional normal distribution with mean ξ_i and variance v_i. We assume (ξ_i, v_i) is from the Dirichlet process with location distribution $G_0(\xi_i, v_i)$ which is an independent bivariate distribution with $\xi_i \sim N(\xi_i | m_0, \tau_0)$ and $v_i \sim IG(v_i | s_0/2, V_0/2)$ and a positive precision parameter α_0. The properties of such a process are described in Chapter 2. On the other hand, conditional on $\mu_j, \xi_{n_0+j}, v_{n_0+j}$, the signal data y_j has a normal distribution with mean $\mu_j + \xi_{n_0+j}$ and variance v_{n_0+j}.

In addition to a Dirichlet process $G_0(\cdot)$, we model μ_i as arising from another independent Dirichlet process. We will assume that the process has a precision α_1 and a location distribution $G_1(\cdot)$, where $G_1(\cdot)$ is a normal distribution with

mean m_1 and variance τ_1. If we know those hyperparameters $\tau_0, m_0, \tau_1, m_1, \alpha_0$ and α_1, then a full analysis can be performed.

For those unknown hyperparameters τ_0 and m_0, the usual assumptions are used to set $\tau_0 \sim IG(\tau_0|t_0/2, R_0/2)$, and $\tau_1 \sim IG(\tau_1|t_1/2, R_1/2)$. Usually, we choose noninformative priors for τ_0 and τ_1. That is, $t_0 = R_0 = t_1 = R_1 = 0.0$. In most cases, we can put non-informative priors on m_0 and m_1. About α_0 and α_1, we choose the prior distributions as $\alpha_0 \sim G(\alpha_0|a_0, b_0)$ and $\alpha_1 \sim G(\alpha_1|a_1, b_1)$ where a_0/b_0 and a_1/b_1 are typically small such as 0.5 or 0.1.

It is obvious that Bayesian hierarchies are used in the current model. Results from Chapter 3 can be used to develop the above model. In the next section, a modified algorithm will be presented.

7.3 Computational Methods

Using the above model, we have to find a way to calculate the posterior distributions for some interesting parameters and predictive density functions of noise and signal data. It is impossible to get analytical results, so a Monte Carlo approximation is employed. Denoting

$$\begin{aligned} \Xi_0 &= (\xi_1, \ldots, \xi_{n_0}), & v_0 &= (v_1, \ldots, v_{n_0}), \\ \Xi_1 &= (\xi_{n_0+1}, \ldots, \xi_n), & v_1 &= (v_{n_0+1}, \ldots, v_n), \\ \Xi &= (\Xi_0, \Xi_1), & v &= (v_0, v_1), \\ \mu &= (\mu_1, \ldots, \mu_{n_1}). \end{aligned}$$

Similar to Chapter 3, a Gibbs sampler can be implemented by noticing some differences between those models. We have the following information from the current model:

(a) known data: noise data X and signal data Y.

(b) unknown data: noise data x_{j+n_0} for $j = 1, \ldots, n_1$.

(c) conditional distributions for noise data: $(x_i|\xi_i, v_i) \sim N(x_i|\xi_i, v_i)$ for $i = 1, \ldots, n$ and $(y_i|\mu_j, \xi_{n_0+j}, v_{n_0+j}) \sim N(y_i|\mu_j+\xi_{n_0+j}, v_{n_0+j})$ for $j = 1, \ldots, n_1$.

(d) Dirichlet processes: $(\xi_1, v_1), \ldots, (\xi_n, v_n)$ is a sample of size n from $D(\alpha_0, G_0(\cdot))$ and $\mu_1, \ldots, \mu_{n_1}$ is a sample of size n_1 from $D(\alpha_1, G_1(\cdot))$.

Our main goals here are to obtain the predictive density functions for noise data and signal data, the distributions of the number of components for noise data and signal data, and a mixture deconvolution of the predictive density function for the signal data.

Firstly let us assume that $\tau_0, \tau_1, m_0, m_1, \alpha_0, \alpha_1$ are all known. In order to make inferences about the noise data, the posterior samples of (Ξ, v) are necessary. Conditional on $\mu = (\mu_1, \ldots, \mu_{n_1})$, $x_{j+n_0} = y_j - \mu_j$ is known for $j = 1, \ldots, n_1$. Now we have data X and data $(x_{n_0+1}, \ldots, x_n)$ with a sample size of n such that x_i is conditional independent distributed with a normal distribution with mean ξ_i and variance v_i, $i = 1, \ldots, n$. Using the ideas of Chapter 3, we can obtain the posterior samples of (Ξ, v).

To complete Gibbs sampling, we must simulate $(\mu|\Xi, V, X, Y)$. Given (Ξ, V), μ is independent of X, so $(\mu|\Xi, v, X, Y) = (\mu|\Xi_1, v_1, Y)$. Then given μ_j and v_{n_0+j}, $(y_j - \xi_{n_0+j})$'s are conditional independent. $y_j - \xi_{n_0+j}$ is from the normal distribution with mean μ_j and variance v_{n_0+j} where μ_j is from $D(\alpha_1, G_1(\cdot))$. Again, this is the framework discussed in Chapter 3. Now, the Gibbs sampler is completed by drawing μ from $(\mu|\Xi, V, X, Y)$.

However, some modifications are needed in order to obtain the predictive

density functions for the noise data and the signal data. Suppose k_0, k_1 are the number of components for noise and signal data, respectively. There are k_0 distinct values (ξ_j^*, v_j^*) among (Ξ, v). Following Chapter 3, we have the predictive distribution of noise x by Monte Carlo average $p(x|\Xi, v, m_0, \tau_0, X, Y)$, namely

$$\begin{aligned} f(x|\Xi, v, m_0, \tau_0, X, Y) &= \tfrac{\alpha_0}{\alpha_0+n} \int N(x|m_0, \tau_0 + z) IG(z|s_0/2, V_0/2) dz \\ &\quad + \tfrac{1}{\alpha_0+n} \textstyle\sum_{j=1}^{k_0} n_{0j} N(x|\xi_j^*, v_j^*), \end{aligned}$$

where n_{0j} is the number of elements of (Ξ, v) with value (ξ_j^*, v_j^*). As we said in Chapter 3, there is no conjugate posterior distribution for z. The integration can be replaced by the Monte Carlo method. For every Gibbs sampler cycle, sampling z from the prior distribution $IG(z|s_0/2, V_0/2)$, we can calculate the normal density function at the sampled value z. Here, we should use a proper prior of v in order to draw prior samples of v.

By assuming the independence of $D(\alpha_0, G_0(\cdot))$ and $D(\alpha_1, G_1(\cdot))$, we have

$$f(\mu_{n_1+1}, \xi_{n+1},\ v_{n+1}|\mu, \xi, v) = f(\mu_{n_1+1}|\mu)\ f(\xi_{n+1},\ v_{n+1}|\xi, v). \qquad (7.1)$$

Once we have the posterior samples of μ and other parameters, the predictive density function of y_{n_1+1} can be obtained by Monte Carlo average:

$$
\begin{aligned}
f(y_{n_1+1}|\mu,\Xi,v,X,Y) \;=\; & \\
\frac{1}{(\alpha_0+n)(\alpha_1+n_1)}[\quad & \alpha_0\alpha_1 d_0 + \alpha_0 \sum_{j=1}^{k_1} n_{1j} d_j \\
& + \alpha_1 \sum_{i=1}^{k_0} n_{0i} N(y_{n_1+1}|m_1+\xi_i^*, v_i^*+\tau_1) \\
& + \sum_{i=1}^{k_0}\sum_{j=1}^{k_1} n_{0i} n_{1j} N(y_{n_1+1}|\xi_i^*+\mu_j^*, v_i^*)],
\end{aligned}
$$

where $d_0 = \int N(y_{n_1+1}|m_0+m_1, z+\tau_0+\tau_1)\ IG(z|s_0/2, V_0/2)dz$,
$d_j = \int N(y_{n_1+1}|\mu_j^*+m_0, \tau_0+z)\ IG(z|s_0/2, V_0/2)dz$, n_{1j} is the number of elements of μ with value μ_j^* and k_1 is the number of distinct values μ_j^* among μ. We can verify above equation by using the equation (7.1) very easily. The first two terms of the expression can be obtained by sampling z from $IG(z|s_0/2, V_0/2)$ and calculating $N(y_{n_1+1}|m_0+m_1, z+\tau_0+\tau_1)$ and $N(y_{n_1+1}|\mu_j^*+m_0, \tau_0+z)$.

Secondly, we can sample these hyperparameters $\tau_0, \tau_1, m_0, m_1, \alpha_0, \alpha_1$ if they are unknown. Suppose the prior distributions of these hyperparameters are given in the previous section, the following distributions are relevant:

$$
\begin{aligned}
\tau_0 &\sim IG(\tau_0|(t_0+k_0-1)/2, (R_0+S_0)/2), \\
(m_0|\tau_0) &\sim N(m_0|\overline{\xi^*}, \tau_0/k_0), \\
\tau_1 &\sim IG(\tau_1|(t_1+k_1-1)/2, (R_1+S_1)/2), \\
(m_1|\tau_1) &\sim N(m_1|\overline{\mu^*}, \tau_1/k_1),
\end{aligned}
$$

where $S_0 = \sum_{j=1}^{k_0}(\xi_j^* - \overline{\xi^*})^2$, $\overline{\xi^*} = \sum_{j=1}^{k_0} \xi_j^*/k_0$, $S_1 = \sum_{j=1}^{k_1}(\mu_j^* - \overline{\mu^*})^2$ and $\overline{\mu^*} = \sum_{j=1}^{k_1} \mu_j^*/k_1$.

We can also sample α_0 and α_1 from equation (3.8) by using

$$\begin{aligned}
(z_0|\alpha_0) &\sim Beta(\alpha_0+1, n), \\
(\alpha_0|z_0, k_0) &\sim w_{z0}G(\alpha_0|a_0+k_0, b_0-log(z_0)) \\
&\quad +(1-w_{z0})G(\alpha_0|a_0+k_0-1, b_0-log(z_0)), \\
(z_1|\alpha_1) &\sim Beta(\alpha_1+1, n_1), \\
(\alpha_1|z_1, k_1) &\sim w_{z1}G(\alpha_1|a_1+k_1, b_1-log(z_1)) \\
&\quad +(1-w_{z1})G(\alpha_1|a_1+k_1-1, b_1-log(z_1)),
\end{aligned}$$

with weights w_{z0} and w_{z1} defined by

$$\begin{aligned}
\frac{w_{z0}}{1-w_{z0}} &= \frac{a_0+k_0-1}{n\ (b_0-log(z_0))}, \\
\frac{w_{z1}}{1-w_{z1}} &= \frac{a_1+k_1-1}{n_1\ (b_1-log(z_1))}.
\end{aligned}$$

Now, a complete Gibbs sampler can be accomplished for unknown hyperparameters τ, m, α. A simulation analysis is given in the next section.

7.4 Simulations

We discuss some numerical aspects and present the results of a small simulation study to show the effects of mixtures of mixtures of modeling. Since two different models, for example, the standard normal distribution $N(y|0,1)$ and a mixture of two normal distributions $0.5N(y|-0.5,1)+N(y|0.5,1)$, may produce the same random samples, we do not use random samples. We analyze the empirical quantiles which represent the distribution quite well. The empirical quantiles $z_1, \ldots, z_n$ can be defined by $F(z_i) = i/(n+1)$ for $i = 1, \ldots, n$ where

$F(\cdot)$ is the cumulative distribution associated with the quantiles. Even though the empirical quantiles have the same drawback, they are better than random samples from the distribution. It is insightful to consider first the empirical quantiles of each component for the mixture model. For example, quantiles of a normal distribution are available from many sources. We can use the quantiles of each normal distribution to obtain the quantiles of mixture of normal distributions. Generally, we can verify the following properties of a mixture of two distributions.

Suppose the mixture consists of two components. the p-quantile of i-th component is z_i for $i = 1,\ 2$. Then the p-quantile of the mixture is between z_1 and z_2. Using those z_1 and z_2 as initial research points, we can use usual numerical methods, say, the bisection method, to obtain the p-quantile of $F(z)$.

150 noise data points were computed from the mixture of two normal distributions $0.5N(x|-1.5,1)+0.5N(x|1.5,1)$. The histogram of simulated noise data is shown in Figure 7.1(a).

At this time, we treat the noise data as 'Y' in the model of Chapter 5, using following prior distributions: $v \sim IG(v|10/2, 10/2)$, $\tau \sim IG(\tau|0.0, 0.0)$ and $\alpha \sim G(\alpha|4, 8)$ where the notations v, τ, α are defined in Chapter 5. Then the posterior distribution of the number of components is given by column 2 in Table 7.1. The predictive density function of the data, displayed by the dotted line, is presented in Figure 7.1(c). The solid line plots the true density function of above mixture.

Now we can sample μ from two discrete states 0 and 10, with probabilities 0.5 and 0.5, respectively. The histogram of signal is plotted in Figure 7.1(b). If we treat the signal alone as data 'Y' in the model of Chapter 5, given the

Table 7.1: Posterior Distributions of the Number of Components

	noise		*signal*	
k	Mixture	Mixture of Mixture	Mixture	Mixture of Mixture
1	0.0000	0.00000	0.00000	0.00000
2	0.9889	0.99010	0.00000	0.62177
3	0.0092	0.00777	0.00000	0.27919
4	0.0016	0.00191	0.92732	0.07756
5	0.0003	0.00016	0.05876	0.01771
6			0.01025	0.00324
7			0.00296	0.00049
8			0.00065	0.00004
9			0.00005	

prior information $v \sim IG(v|10/2, 10/2)$, $\tau \sim IG(\tau|0.0, 0.0)$ and $\alpha \sim G(\alpha|4, 8)$, the posterior distribution of the number of components is shown in column 4 in Table 7.1. The predictive density functions of the signal is plotted by the dotted line in Figure 7.1(d). The solid line plots the true density function of signal mixture.

In order to follow the general approach of this chapter, the prior information for all parameters is given by $v \sim G(v|10/2, 10/2)$, $\tau_i \sim G(\tau_i|0, 0)$, for $i = 0,\ 1$. $\alpha_0 \sim G(\alpha_0|4, 8)$ and $\alpha_1 \sim G(\alpha_1|4, 12)$. We performed simulation studies comparing results of mixtures of mixtures of modeling and the true distribution. Although we use different prior information for mixtures of modeling and mixtures of mixtures of modeling, comparisons to the true distribution are also provided where available. We chose the initial value for ξ_i as i^{th} noise data, μ_j as the difference of j^{th} signal data and noise data. For each analysis, the number of burn-in cycle was 2,000 and Monte Carlo sample size used was 8,000. The convergence of the Gibbs sampler was monitored by using the method of Chapter 3.

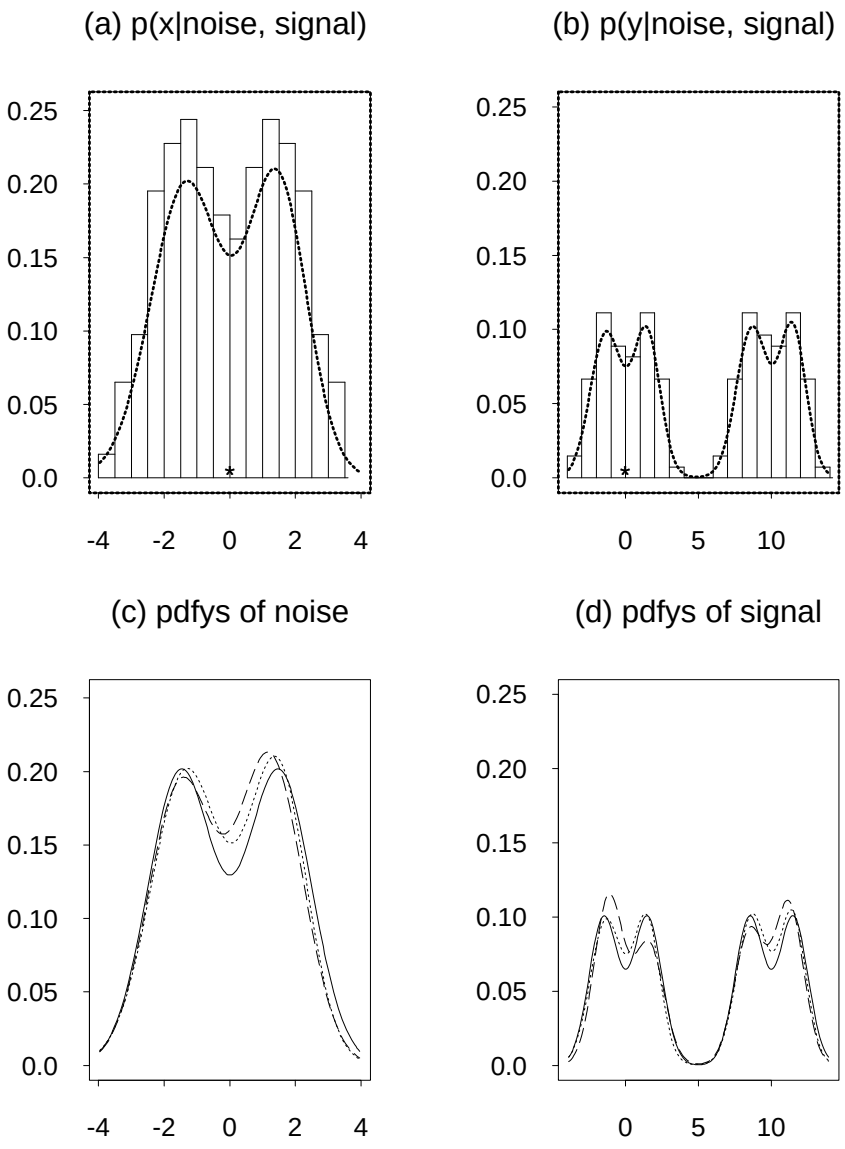


Figure 7.1: Simulation Analysis

—— true density functions for plot (c) and (d)
..... predictive density functions of mixture of mixture method
- - - predictive density functions of method of Chapter 5.

The posterior distributions for k_0 and k_1 are given by columns 3 and 4 in Table 7.1. It is clear that there are two components for the noise data. The posterior mode for k_1 is 2. The chance for more than two components for signal data is about 0.378. However, if we use a mixture analysis presented in Chapter 5, column 4 in Table 7.1 shows that the posterior mode for k_1 is about 4. Figure 7.1 shows the predictive density functions by using different models. As can be seen from the results, the model of mixtures of noise is better than the model of a single normal noise. The predictive density functions of noise and signal based on mixtures of mixtures of modeling are much closer to their true density functions, respectively. Figure 7.1(a) and 7.1(b) show how the predictive density functions from mixtures of mixtures modeling match up their histograms. This simulated data is very clearly bimodal in noise and signal. The goal of such a test was really just to test the program and theory.

7.5 Application to Neurophysiological Data

In this section we shall explore analysis of mixtures of mixtures for a real data set. The primary source for the data is the neurophysiological study conducted by Dr. Dennis Turner, a neuroscientist at Duke University Medical Center. The background can be found in Chapter 5.

As Kullmann (1989) points out two principal forms of departure from a normal distribution can occur in experimental records of background noise of physiological phenomena. These correspond to the skewness and kurtosis, which can also be described by the third and fourth moments of the distribution about its mean. A positive skewness can arise when there are frequent spontaneous EPSPs contaminating the noise, since the rapid depolarization

will on occasion fall in the sampling window. Spontaneous EPSPs, on the other hand, are much smaller when expressed as the peak voltage change, and consequently do not influence the noise distribution to the same extent. Kurtosis can be positive or negative. A positive kurtosis implies that the center and tails of the noise distribution are over-represented in relation to a normal distribution with the same variance. This occurs more frequently than a negative kurtosis, and may represent large step changes in measurement, as may occur from electrode artefacts. It can also arise from non-stationary noise processes, since part of the sample is collected at a time when the membrane potential is relatively noise free, contributing a large number of observations to the center of the distribution, and another part when there are large voltage deviations, adding to the tails of the distribution. A negative kurtosis, on the other hand, implying a relatively square distribution, can occur if a stereotyped noise process contributes frequent depolarizing and hyperpolarizing deviations of relatively constant gradient. This can arise from periodic oscillations in the background synaptic input to the cell, or from electrical artefacts. The data given in Figure 7.2(a) and 7.2(b) were collected by Dr. Turner, and represent the postsynaptic potentials in physiological preparation. In this case, the noise sample size is the same as the signal sample size, $n_0 = n_1 = 651$. This is quite a large sample. The noise histogram indicates a non-normal distribution that is multimodal. The signal histogram shows evidence of several components which can be modeled via mixtures of mixtures.

Reference priors for the parameters τ_0, τ_1, m_1 should apply reasonably well to noise and signal analysis, since we do not have prior information about these parameters. we can fix $m_0 = 0.0$. The prior distribution of v should be considered very carefully. In order to get the prior information of v, we can

Table 7.2: Distributions of k_0 and k_1

	noise		*signal*	
K	Prior	Posterior	Prior	Posterior
1	0.96173	0.00000	0.94079	0.00000
2	0.03135	0.89000	0.04979	0.00000
3	0.00802	0.11000	0.00765	0.00000
4			0.00142	0.00000
5			0.00027	0.48805
6			0.00005	0.26520
7				0.24775

assess the expected range of the data. In our case, we choose $2\sqrt{v} = 0.22$ and obtain $v = 0.012$. This estimation of variance can be used to provide the prior information of v. In our case, We can choose $s_0 = 10$ and $V_0 = 0.012 \times s_0 = 0.12$. Also, we use $\alpha_0 \sim G(\alpha_0|0.1, 10)$ and $\alpha_1 \sim G(\alpha_1|0.1, 20)$.

Table 7.2 shows posterior distributions of k_0 and k_1 given the noise and signal data.

The predictive distribution of noise was estimated using the methods of this chapter. Figure 7.3(a) illustrates the match between the predictive density function and the raw data. The number of bins of the histogram is roughly equal to 25. The distribution is positively skewed. A normal distribution for the noise data is not suitable.

Figure 7.3(b) displays the estimates of the predictive density function of signal data. It is obvious that there are components for this data.

Figure 7.3(c) gives the deconvolution of the predictive distribution of noise data. Since the weight is about 0.0 at location 0.45, so there are about two components with location around -0.17 and 0.17.

Figure 7.3(d) gives the deconvolution of the predictive distribution of signal.

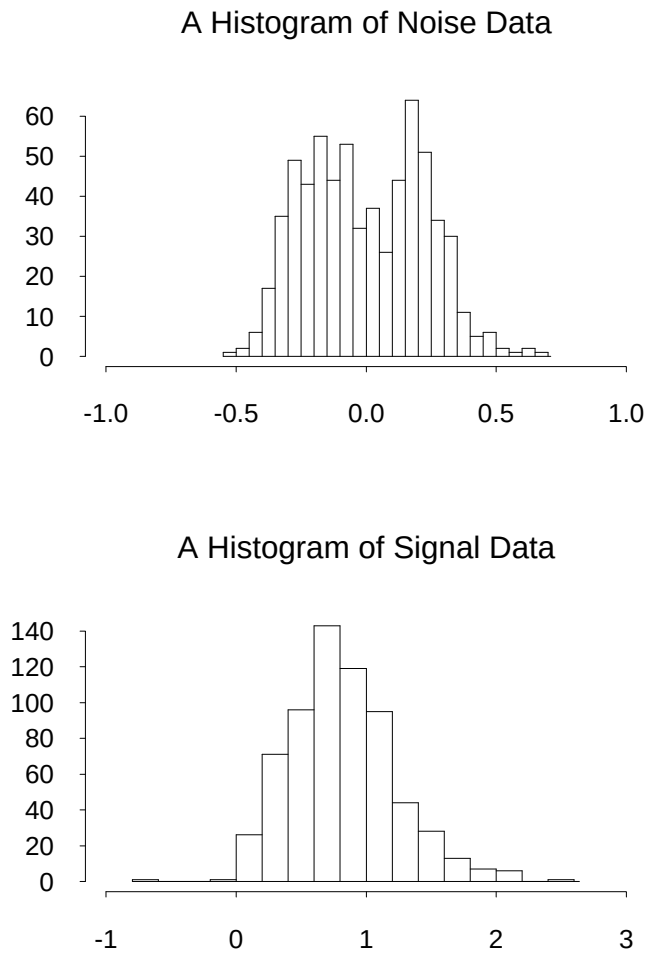


Figure 7.2: Histograms of Mixtures of Mixtures

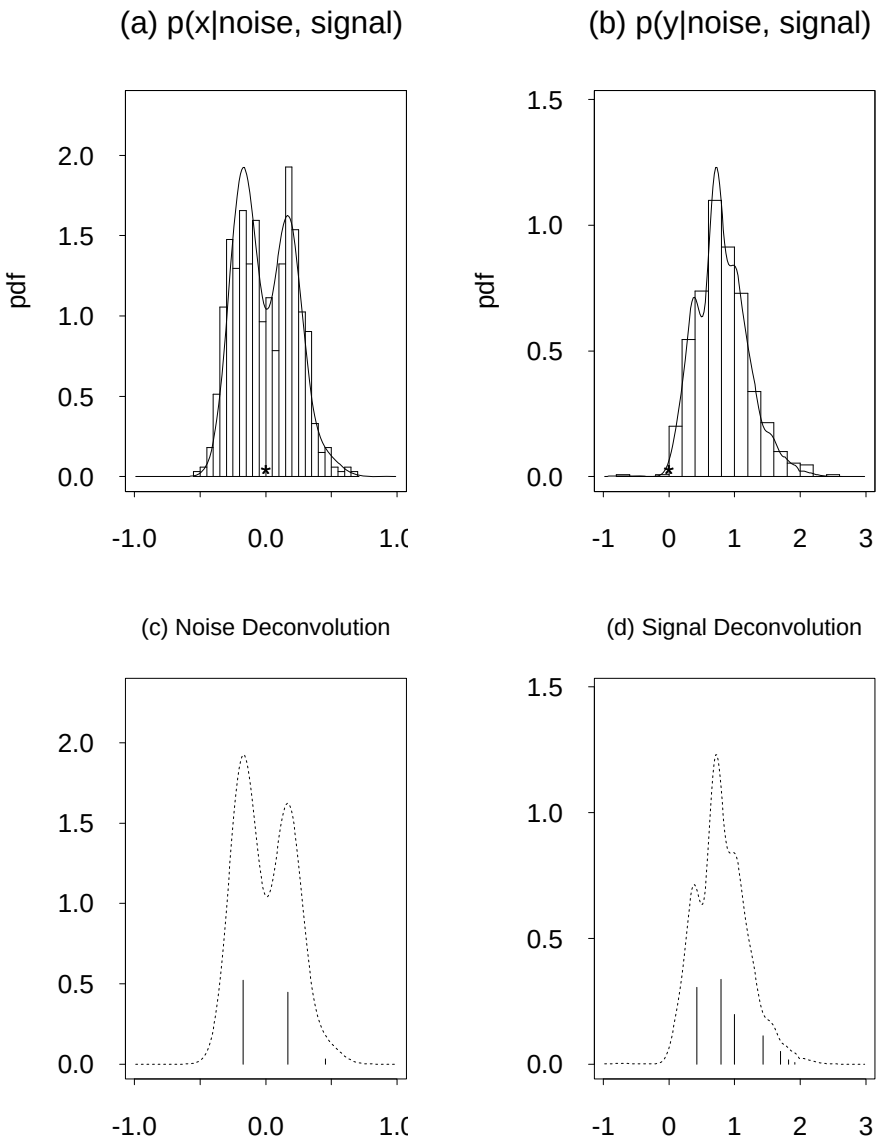


Figure 7.3: Analysis of Mixtures of Mixtures

There seem to be 5 components since the sixth and seventh components with very small weights 0.017 and 0.007.

7.6 Discussion

This is an initial exploration of these mixtures of mixtures of modeling. The major finding of these analyses is that two independent Dirichlet process priors can be applied to some parameters of noise and signal data. Hierarchical Bayesian models can be used to solve such mixtures of mixtures modeling. Gibbs sampling technique brings one way to deal with computational difficulties. The advantage of the mixtures of mixtures of modeling over mixtures of modeling is the non-normal noise distribution can be incorporated. The disadvantage of the above model is that it takes a long time to compute some interesting quantities. The real problem of the implementation is how to choose the prior distributions for v, α_0 and α_1, how to monitor the convergence of the Gibbs sampler and how to overcome the overfitting problem. One way to solve the overfitting problem is to choose very small prior means for α_0 and α_1. Of course, a special case of a normal noise distribution should be applied by the mixtures of mixtures of model.

Chapter 8

Conclusions and Further Work

This dissertation examined both nonparametric Bayesian analysis for mixture modeling and the use of the Gibbs sampling scheme to approximate posterior estimates. The definition and some characteristics of the Dirichlet process are given in Chapter 2. It is extremely difficult to obtain analytic results for posterior distributions, so approximations are required. The Gibbs sampling method is used to solve this problem.

My contributions are as follows:

(a) For the Gibbs sampling method based on unknown means and variances, I proved the convergence of the Gibbs sampler based on the configuration space, demonstrated that the Gibbs sampler based on configurations is relatively fast and developed the Gibbs sampler for non-conjugate base prior.

(b) I proved the recurrence relationships among the distributions of order statistics of independent, but not identically distributed random quantities.

(c) I showed the practicality of the nonparametric Bayesian analysis for mixture modeling, and developed various methodological and computational aspects.

(d) I discussed Bayesian density estimation for grouped data, compared Bayesian methods with classical approaches and investigated Bayesian features of MEND approach.

(e) I introduced Bayesian analysis of mixtures of mixtures. Some computational difficulties were easily overcome by a Gibbs sampling scheme.

My further research areas are as follows,

(1) Moving in a more theoretical direction, it may be possible to find the recurrence relationships among the distributions of spacing statistics. It may be also possible to generalize the baseline distribution $G_0(.)$ to any other interesting distributions. Throughout this dissertation, Y was assumed a univariate random variable. It may be very easily to considered as a multivariate random variable Y.

(2) Moving in a more computational direction, We should consider how fast the Gibbs sampler based on the configuration space converges.

(3) Moving in a more practical direction, we must consider how to choose the prior information about s_0 and V_0 for the analysis of mixtures of mixtures. Different prior values of s_0 and V_0 change the results dramatically. For other fixed parameters, smaller V_0/s_0 is , we have more components for the noise data we have. The prior distribution of α is critical as well. If the prior expectation of α is small, the number of components for

the signal (or noise) will be large. It is imperative to understand the background of the situation that produced the data. Consulting with experts in the field would be very important.

Appendix

Chapter 1

C Program — An Example

This appendix contains a computer program for implementing the algorithm (3.1a) in Chapter 3. It was used to generate the results for Example 3.1 that appear in Section 3.1. It is written in the C language. The program calls many subroutines which are not given here, but are told where you can find them. The rest of algorithms for (3.1b), (3.1c), EW, non-conjugate prior case, grouped data analysis, mixture of mixture analysis and large simulation analysis, can be modified from this 'core' program.

```
/*
 * required header files:
 */


#include      <stdlib.h>
#include      <stdio.h>
#include      <math.h>
```

```
/*
 * defined constants:
 */

#define MAXK      8       /* the maximum of number of components */
#define NZ        200     /* number of grid points for plots     */

double    ran1(int *);    /* This subroutine generates random
                             samples from uniform U(0,1). It can be
                             found from Numerical Recipes in C
                             (Press, et al., 1990) and is not given here */
double    gasdev(int * ); /* This subroutine generates random
                             samples from normal N(0,1). It can be
                             found from Numerical Recipes in C
                             (Press, et al., 1990) and is not given here */

double    gamdev(double *, int * ); /* This subroutine generates random
                                       samples from gamma distribution
                                       G(a, 1). See Numerical Recipes in
                                       C (Press, et al., 1990) and is not
                                       given here */
double    gammln(double );          /* This subroutine returns the value
                                       of log(Γ(.)), It can be found
                                       from Numerical Recipes in C
```

```
                                        (Press, et al., 1990) and is not
                                        given here */
double      pnorm(double *);            /* This subroutine returns cdf for
                                        the standard Gaussian distribution.
                                        See also Kennedy & Gentle p.93 */


void gibbscon(int *, int *, double *, double *, double *, double *,
              double *, double *, double *, double *, double *, int *,
              int *, int *, double *, double *, double *, double * ,
              double *, double *, double *, double *);
              /* See following definition */
void revisepy( int *, int *, double *, double * , double *, double *,
              double *, double *, double *, int *, double *, double *,
              double *, double *, int *, double *, double *, double *);
              /* See following definition */


              /* Following subrountines can be found from Numerical
              Recipes in C (Press, et al., 1990) and are not given here */
void        nrerror(char *);
void        free-dvector(double *, int, int);
void        free-ivector(int *, int, int);
double      *dvector(int, int);
int         *ivector(int, int);
double      **dmatrix(int, int, int, int);
void        free-dmatrix(double **, int, int, int, int);
```

```
    main( )
{

    int     i,          /* loop counter */
            j,          /* loop counter */
            icyc,       /* loop counter */
            N,          /* sample size (eg. number of data) */
            NMC,        /* Monte Carlo (MC) sample size (eg. 5000) */
            UN,         /* maximum of sample size */
            INI,        /* number of burn-in cycles */
            ndis,       /* components counter */
            *nth,       /* simulation counter for components */
            *ni,        /* array for number of data for each component */
            *im,        /* array for index */
            maxk,       /* maximum for number of components */
            MAXK1,      /* dummy variable */
            ik,         /* counter */
            inmc;       /* counter */
double      *clust,     /* array for P(k|y) */
            *dmu,       /* array for different means of components */
            *dv,        /* array for different variances of components */
            *Emu,       /* array for posterior mean of μ */
            *Ev,        /* array for posterior mean of v */
            *sdmu,      /* array for posterior standard deviation of μ */
```

```
*sdv,      /* array for posterior standard deviation of v */
*ordmu,    /* array for dummy variable */
*ordv,     /* array for dummy variable */
*ordni,    /* array for dummy variable */
*dmu1,     /* array for dummy variable */
*modes,    /* array for posterior distribution of number of modes
              based on posterior predictive density function */
*th,       /* array for output of component means */
*sdth,     /* array for output of component
              standard deviation */
*pi,       /* array for posterior estimation of
              component weights */
*sdpi,     /* array for standard deviation of posterior
              estimation of component weights */
*dni,      /* array for dummy variable */
x, a,      /* dummy variables */
b, z,      /* dummy variables */
tau,       /* τ variable */
aa0,
ab0,       /* prior distribution of α G(aa0, ab0) */
ALPHA,     /* dummy variables for α */
ALPHA0,    /* dummy variables for α */
s0,
V0,        /* prior information for v, IG(s0/2, S0/2) */
cs0,       /* dummy variable */
t0,
```

```
            R0,         /* prior distribution of τ IG(t0/2, R0/2) */
            m0,         /* dummy variable for m */
            *baryi,     /* dummy variable */
            *SSi,       /* dummy variable */
            *y,         /* array for data Y */
            *ry,        /* grid points for pdf of y */
            *pdfy,      /* array for pdf of y */
            *cdfy,      /* array for cdf of y */
            ay,
            by;         /* predictive density function over (ay, by) */
FILE        *fp;


if ((fp = fopen("dat.ch3.1", "r")) == NULL)
            nrerror("Data file dat.ch3.1 not found");
fscanf(fp,"%d%d%d%d%le%le%le%le%le%le%le%le%le%le", &N,
            &NMC, &INI, &NIT, &s0, &V0, &t0, &R0, &aa0, &ab0);
y           = dvector(1, N);
for (i=1; i<= N; i++) {
            fscanf(fp, "%lf", &y[i]);
}
fclose(fp);
fflush(stdout);


/* general setup and initialisations */
UN                  = N+1;
```

```
nth         = ivector(1, UN);
im          = ivector(1, UN);
ni          = ivector(1, UN);
Emu         = dvector(1, UN);
Ev          = dvector(1, UN);
sdmu        = dvector(1, UN);
sdv         = dvector(1, UN);
dmu         = dvector(1, UN);
dmu1        = dvector(1, UN);
dv          = dvector(1, UN);
ordmu       = dvector(1, UN);
ordv        = dvector(1, UN);
ordni       = dvector(1, UN);
modes       = dvector(1, UN);
clust       = dvector(1, UN);
th          = dvector(1, UN);
sdth        = dvector(1, UN);
pi          = dvector(0, UN);
sdpi        = dvector(0, UN);
dni         = dvector(1, UN);
baryi       = dvector(1, UN);
SSi         = dvector(1, UN);
ry          = dvector(1, NZ);
pdfy        = dvector(1, NZ);
cdfy        = dvector(1, NZ);
```

```
if( s0 !=0.e0){
        cs0= gammln((s0+1.e0)/2.e0)-gammln(s0/2.e0);
        cs0 -= 0.5e0*log(s0*3.1415926e0);
        cs0 = exp(cs0);
} else {
        puts("Note: 1/v is Gamma(0,0)");
}

ALPHA  =  ALPHA0 = aa0/ab0;
/*      initial setup for plots */
ay      = y[1];
by      = y[1];
for (i=2;i<= N; i++) {
        if (ceil(y[i]) > by){
            by = ceil(y[i]);
        }
        if (floor(y[i]) < ay){
            ay = floor(y[i]);
        }
}
for (i =1;  i<= NZ; i++) {
        pdfy[i]= cdfy[i]= 0.e0;
        ry[i]=ay+(by-ay)*i/(double)(1+NZ);
}
a       = t0/2.e0;
b       = R0/2.e0;
```

```
inmc     = NMC;
maxk     = 0;
ndis     = N;
if (t0 >  1.e0){
          tau = R0/(t0-1.0);
} else {
          tau = 5.e0;
}
pi[0]    = sdpi[0] = 0.e0;
for ( i = 1;  i<= N; i++) {
          dmu[i] = y[i];
          dv[i] = V0/s0;
          Emu[i] = Ev[i] = sdmu[i] = sdv[i] = 0.e0;
          ni[i] = 1;
          dni[i]= 1.e0;
          baryi[i]= y[i];
          SSi[i] =0.e0;
          im[i] = i;
          clust[i] = th[i] = sdth[i] = pi[i] = sdpi[i] = 0.e0;
          nth[i]= 0;
}
m0       = (ay+by)/2.0;
for ( icyc= 1; icyc<= INI; icyc++){
          gibbscon(&N, &ndis, &ALPHA, &m0, &s0, &V0, &t0,
               &R0, &tau, dmu, y, &icyc, ni, im,
               &aa0, &ab0, dv, ralpha, palpha, &cs0, baryi, SSi);
```

```
        if (icyc <= INI/2) {
            ALPHA = ALPHA0;
        }
}
puts("  Burn-In complete, cycling ...");
/*      NOW THE MAIN LOOP – ONE LONG RUN */
for ( icyc= 1;  icyc <= inmc; icyc++) {
        gibbscon(&N, &ndis, &ALPHA, &m0, &s0, &V0, &t0,
            &R0, &tau, dmu, y, &icyc, ni, im,
            &aa0, &ab0, dv, ralpha, palpha, &cs0, baryi, SSi);
        clust[ndis] += 1.0e0;
        revisepy(&N, &ndis, &m0, &s0, &V0, &tau, dv,
            &ALPHA, dmu, ni, ry, pdfy, cdfy,
            modes, &icyc, &cs0, baryi, SSi);
        if (ndis> 1) {
            myorder(&ndis, dmu,ordmu,dv, ordv, ni, ordni);
        }
        if (maxk < ndis){
            maxk = ndis;
        }
        for (i=1; i <=ndis; i++){
            x = ordmu[i];
            th[i] += x;
            sdth[i] += x*x;
            nth[i] +=1;
        }
```

```
        MAXK1= MAXK;
        if (ndis < MAXK){
            MAXK1=ndis;
        }
        for (i=1; i <= ndis; i++){
            x = ordni[i]/(ALPHA0+ (double) N);
            pi[i] += x;
            sdpi[i] += x*x;
        }
        for(i=1; i<= N; i++){
            j = im[i];
            x = dmu[j];
            Emu[i] += x;
            sdmu[i] += x*x;
        }
}


free-dvector(y, 1, N);
free-dvector(dmu, 1, UN);
free-dvector(dmu1, 1, UN);
free-ivector(im, 1, N);
free-dvector(dv, 1, UN);
free-ivector(ni, 1, N);


for ( i=1; i<= maxk; i++){
```

```
            x = (double )(nth[i]);
            th[i] /= x;
            z = sdth[i]-x*th[i]*th[i];
            sdth[i]=sqrt(z/(x-1.e0));
            pi[i] /= (double )(nth[1]);
            z = sdpi[i]-pi[i]*pi[i]* (double )(nth[1]);
            sdpi[i]=sqrt(z/(nth[1]-1.e0));
}
x           = (double )(NMC);
for ( i=1; i<= N; i++){
            clust[i] /= x;
            modes[i] /= x;
            Emu[i] /= x;
            z = sdmu[i] - Emu[i]*Emu[i]* x;
            sdmu[i] =sqrt(z/(x-1.0));
            puts("Here you can print Emu[i], sdmu[i], P(k|Y), weights pi[i], etc.");
}
for ( i=1; i<= NZ; i++){
            pdfy[i] /= x;
            cdfy[i] /= x;
}
puts("    you can print predictive density function");
puts("    end of main rountine");


/*
```

```
 *          routine:        gibbscon
 *          description:    one Gibbs sampling — resulting draw for
 *                          distinct values in (dmu, dv) based on
 *                          algorithm (3.1a)
 */


void gibbscon(int *N, int *ndis, double *ALPHA, double *m0, double *s0,
              double *V0, double *t0, double *R0, double *tau, double
              *dmu, double *y, int *icyc, int *ni, int *im, double *aa0,
              double *ab0, double *dv, double *ralpha, double *palpha,
              double *cs0, double *baryi, double *SSi)
{
              int i, j, jj, id, ncom, nn;
              double df, e, z, z1, z2, yy, x, dumal, *pr, vj, csj, vj0, dmuj,
                   s, taupt, dumv, mj, baryj, SSj, sj, nk;
              static int ir=0;
              ncom = *ndis;
              taupt = *tau;
              nn = *N;
              jj = *icyc;
              pr= dvector(0, nn);
              if( *s0 != 0.e0){
                   vj0 = sqrt((1.0e0+ taupt)*(*V0)/(*s0));
              }
              for (j=1; j<= nn; j++){
```

```
id = im[j];
yy=y[j];
if( *s0 != 0.e0){
    e =log(1.e0+(yy- *m0 )*(yy- *m0)/( *s0 *vj0*vj0));
    e = (*ALPHA)*(*cs0)*exp(-0.5e0*(*s0+1.e0)*e)/vj0;
} else{
    e = (*ALPHA)/fabs(yy- *m0);
    /* gamma(0) = 1. gamma(0.5)= sqrt(pi) */
}
z= e;
pr[0] = e;
x = (double)(ni[id]);
baryi[id] = (x*baryi[id]-yy)/(x-1.e0);
SSi[id] -=((x-1.0)/x)*(baryi[id]-yy)*(baryi[id]-yy);
–ni[id];
if ( ni[id] == 0){
    –ncom;
    for (i = id; i<= ncom; i++){
        dmu[i] = dmu[i+1];
        dv[i] = dv[i+1];
        ni[i] = ni[i+1];
        baryi[i] = baryi[i+1];
        SSi[i] = SSi[i+1];
    }
    for (i=1; i<= nn; i++){
        if (im[i] > id ){
```

```
                –im[i];
            }
        }
    }
    for(i=1; i<= ncom; i++){
        nk = (double) (ni[i]);
        baryj = baryi[i];
        SSj = SSi[i];
        sj = nk+ (*s0);
        csj= gammln(( sj+1.e0)/2.e0)-gammln(sj/2.e0);
        csj -= 0.5e0*log(sj*3.1415926e0);
        csj = exp(csj);
        s = 1.e0+taupt * nk;
        mj = (taupt *nk *baryj + *m0)/s;
        vj = 1.e0+ taupt/s;
        vj *= (*V0+SSj+nk*(baryj- *m0)*(baryj- *m0)/s);
        vj /= sj;
        vj = sqrt(vj);
        e =log(1.e0+(yy-mj)*(yy-mj)/(sj*vj*vj));
        e = nk * csj* exp(-0.5e0*(sj+1.e0)*e)/vj;
        z += e;
        pr[i] = e;
    }
    for(i=0; i<= ncom; i++){
        pr[i] /= z;
    }
```

```
        multgen(&ncom, pr, &ir);
        if(ir ){
            im[j] = ir;
            nk = (double)(ni[ir]);
            baryj = baryi[ir];
            SSi[ir] +=((baryj-yy)*(baryj-yy)*nk/(nk+1.0));
            baryi[ir] = (yy+ nk*baryj)/(nk+1.e0);
            ni[ir] +=1;
        } else {
            ++ncom;
            ni[ncom] = 1;
            im[j] = ncom;
            baryi[ncom] = yy;
            SSi[ncom] = 0.e0;
        }
    }
    free-dvector(pr, 0, nn);
    for(j=1; j<= ncom; j++){
        x = (double )(ni[j]);
        SSj = SSi[j];
        baryj = baryi[j];
        s = 1.e0+taupt * x;
        df = (*s0+ x) / 2.e0;
        vj = *V0+SSj+x*(baryj- *m0)*(baryj- *m0)/s;
        vj /=(2.e0*gamdev(&df,&jj));
        dv[j] = vj;
```

```
        mj = (taupt *x*baryj + *m0)/s;
        vj * = (taupt/s);
        dmu[j] = mj + sqrt(vj)*gasdev(&jj);
}
/* joint distribution P(tau, m0, dmu, dv)
        =P(dmu, dv)*P(tau, m0|dmu, dv) */
yy = 0.e0;
z = 0.e0;
for(j=1; j<= ncom; j++){
        dmuj = dmu[j];
        vj = dv[j];
        yy += dmuj/vj;
        z += 1.e0/vj;
}
yy /= z;
taupt = (*R0);
df = (*t0+ncom -1.0)/2.e0;
for(j=1; j<= ncom; j++){
        e = yy - dmu[j];
        taupt +=e*e/dv[j];
}
taupt /= (2.e0*gamdev(&df, &jj));
*m0 = yy +gasdev(&jj)*sqrt(taupt/z);
/* and now sample (α|D) */
x=(*ALPHA)+1.e0;
x= gamdev(&x, &ncom);
```

```
                df= (double )(nn);
                dumal= x/(x+gamdev(&df, &ncom));
                z = (*aa0)+(double )(ncom);
                z1 = (*ab0-log(dumal));
                z2 = z-1.e0;
                e= ran1(&ncom);
                yy=(z2)/(z2+(nn)*z1);
                if (e < yy) {
                    e= z;
                } else {
                    e = z2;
                }
                *ALPHA = gamdev(&e, &ncom)/z1;
                *ndis = ncom;
                *tau = taupt;
                *N = nn;
}


/*
 *          routine:        revisepy
 *          description:    to obtain the predictive density function
 */


void revisepy(int *N, int *ndis, double *m0, double *s0, double *V0,
                double *tau, double *dv, double *ALPHA, double *dmu,
```

```
int *ni, double *ry, double *pdfy , double *cdfy, double
*modes, int *icyc, double *cs0, double *baryi, double *SSi)
{
double cz, r2, m, pz, pzm, vn, vn0, w, x, yy, z, zero, ri,df;
double *pmode, dumv, nk, baryj, SSj,vj, csj, mj, s, sj,taupt, e;
int i, j, jj, nmode;
zero = 0.e0;
taupt = *tau;
jj = (*icyc);
r2=sqrt(2.0e0*3.14159e0);
pzm = zero;
pmode = dvector(1, NZ);
df=(*s0)/ 2.e0;
dumv = (*V0)/(2.e0*gamdev(&df,&jj));
if ( *s0 != 0.e0){
    vn0=sqrt((1.0e0+(*tau))*(*V0)/(*s0));
}
for ( j=1; j<= NZ; j++){
    yy=ry[j];
    ri=(*ALPHA);
    vn = sqrt((*tau +1.e0)* dumv);
    m=(yy- (*m0))/vn;
    cz = ri*pnorm(&m);
    if( *s0 != 0.e0){
        m=(yy-(*m0))/vn0;
        pz= -0.5*(1.0e0+(*s0))*log(1.0e0+m*m/(*s0));
```

```
        pz = (*ALPHA)*(*cs0)*exp(pz)/vn0;
    } else {
        pz = (*ALPHA)/ fabs(yy- *m0);
    }
    for(i=1; i<= *ndis; i++){
        nk = (double) (ni[i]);
        baryj = baryi[i];
        SSj = SSi[i];
        sj = nk+ (*s0);
        csj= gammln(( sj+1.e0)/2.e0)-gammln(sj/2.e0);
        csj -= 0.5e0*log(sj*3.1415926e0);
        csj = exp(csj);
        s = 1.e0+taupt * nk;
        mj = (taupt *nk*baryj + *m0)/s;
        vj = 1.e0+ taupt/s;
        vj *= (*V0+SSj+nk*(baryj- *m0)*(baryj- *m0)/s);
        vj /= sj;
        vj = sqrt(vj);
        e =log(1.e0+(yy-mj)*(yy-mj)/(sj*vj*vj));
        pz += nk * csj* exp(-0.5e0*(sj+1.e0)*e)/vj;
        vn = sqrt(dv[i]);
        m=(yy-dmu[i])/vn;
        cz += nk*pnorm(&m);
    }
    pz /= (*ALPHA +(*N));
    cz /= (*ALPHA +(*N));
```

```
            pdfy[j] += pz;
            cdfy[j] += cz;
            pmode[j] = pz - pzm;
            pzm = pz;
        }
        /* now count number of modes */
        nmode = 0;
        x = pmode[2];
        for (j=3; j <=(NZ-1);j++){
            z = pmode[j];
            w = x * z;
            if ( w < zero ) {
                ++nmode;
            }
            x = z;
        }
        nmode = (1 + nmode )/2;
        if (nmode > *ndis) nmode = *ndis;
        modes[nmode] += 1.e0;
        free-dvector(pmode, 1, NZ);
}


/*
 *      routine:        multgen
 *      description:    Select component i with prob pr(i)
```

```
 *                          component ir is selected from pr[0]....pr[*N]
 */


void multgen(int *N, double *pr, int *ir)
{
        double *v, x;
        int i ;
        v = dvector(0,*N);
        v[0] = pr[0];
        for (i=1; i<= *N; i++) {
            v[i] = v[i-1] +pr[i];
        }
        x = ran1(ir);
        *ir = 0;
        for (i=1; i<= *N; i++) {
            if (x >= v[i-1]){
                *ir =i;
            }else {
                free-dvector(v, 0, *N);
                return ;
            }
        }
        free-dvector(v, 0, *N);
}
```

```
void myorder(int *N, double *mu, double *ormu, double *dv,
             double *ordv, int *ni, double *orni)
{
             int i, j, n1, step;
             double *temp, xnum, nni, dvi, *tempdv, *tempni;
             temp = dvector(1,*N);
             tempdv = dvector(1,*N);
             tempni = dvector(1, *N);
             for (i=1; i¡= *N; i++){
                  temp[i] =mu[i];
                  tempdv[i] = dv[i];
                  tempni[i] = (double ) (ni[i]);
             }
             n1=(*N);
             step = 0;
             while(n1 > 0){
                  xnum = temp[1];
                  nni = tempni[1];
                  dvi = tempdv[1];
                  j = 1;
                  for (i=2; i<= n1; i++){
                       if (temp[i] < xnum){
                            xnum = temp[i];
                            j =i;
                            nni = tempni[i];
                            dvi = tempdv[i];
```

```
                    }
                }
                ++step;
                ormu[step] = xnum;
                orni[step] = nni;
                ordv[step] = dvi;
                n1 -=1;
                for (i=j; i<=n1; i++){
                    temp[i] = temp[i+1];
                    tempni[i] = tempni[i+1];
                    tempdv[i] = tempdv[i+1];
                }
            }
            free-dvector(temp, 1, *N);
            free-dvector(tempdv, 1, *N);
            free-dvector(tempni, 1, *N);
}
```

Bibliography

[1] Abramowitz, M. and Stegun, I. A. (1965) *Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables,* Dover: New York.

[2] Antoniak, C. (1974) Mixtures of Dirichlet processes with application to Bayesian nonparametric problems. *Ann. Statist.,* 2, 1152-1174.

[3] Bhattacharya, C. G. (1967) A simple method of resolution of a distribution into Gaussian components. *Biometrics,* 23, 115-135.

[4] Balakrishnan, N. (1988) Recurrence relations for order statistics from non independent and non-identically distributed random variables. *Ann. Inst. Statist. Math.,* 40, 273-7.

[5] Balakrishnan, N., Bendre, S. M. and Malik, H. J. (1992) General relations and identities for order statistics from non-independent non-identical variables. *Ann. Inst. Statist. Math.,* 44, 177-83.

[6] Berry, D. A. and Christensen, R. (1979) Empirical Bayes estimation of a binomial parameter via mixture of Dirichlet processes, *Ann. Statist.,* 7, 558-568.

[7] Cohen, C. 1985 On asymptotic normality of limiting density functions with Bayesian implications. *J. R. Statist. Soc. B,* 47, 540-546.

[8] Cramer, H. (1946) *Mathematical Methods of Statistics.* Princeton: Princeton University Press.

[9] Crawford, S. (1988) An approximate Bayesian analysis of finite mixture distributions, *Technical Report* 440, 1-140.

[10] David, H. A. (1981) *Order Statistics*, 2nd ed. New York: Wiley.

[11] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. R. Statist. Soc. B,* 39, 1-38.

[12] Devroye, L. (1986) *Non-Uniform Random Variate Generation*, Springer-Verlag, New York.

[13] Doss, H. (1991) A study of the convergence properties of successive substitution sampling based on Harris recurrence of Markov chains. Preprint.

[14] Escobar, M. D. (1991) Estimating normal means with a Dirichlet process prior. Technical Report 512, Department of Statistics, Carnegie Mellon University.

[15] Escobar, M. D. and West, M. (1991) Bayesian density estimation and inference using mixtures. To appear *J. Amer. Statist. Assoc.*.

[16] Everitt, B. S. and Hand, D. J. (1981) *Finite Mixture Distributions.* London: Chapman and Hall.

[17] Ferguson, T. S. (1973) A Bayesian analysis of some nonparametric problems. *Ann. Statist.,* 1, 209-230.

[18] Ferguson, T. S. (1983) Bayesian density estimation by mixtures of normal distributions. *Statistical Decision Theory and Related Topics* III, 1 (S. Gupta and J. Berger, eds.), 385-401.

[19] Gelfand, A. E. and Smith, A. F. M. (1990) Sampling based approaches to calculating marginal densities, *J. Amer. Statist. Assoc.,* 85, 398-409.

[20] Geweke, J. (1989) Bayesian Inference in Econometric Models Using Monte Carlo Integration. *Econometrika, 57,* 97-109.

[21] Johnson, W. and Christensen, R. (1986) Bayesian nonparametric survival analysis for grouped data. *The Canadian Journal of Statistics,* 14, 307-314.

[22] Kennedy, W. J. and Gentle, J. E. (1980) *Statistical Computing.*

[23] Kiefer, N. M. (1978) Discrete parameter vaiation: efficient estimation of a switching regression model. *Econometrica,* 46, 427-434.

[24] Kullmann, D. M. (1989) Applications of the expectation-maximization algorithm to quantal analysis of postsynaptic potentials, *J. Neurosci. Methods,* 30, 231-245.

[25] Kullmann, D. M. (1992) Quantal analysis using maximum entropy noise deconvolution, *J. Neurosci. Methods,* 44, 47-57.

[26] Kullmann, D. M. and Nicoll, R. A. (1992) Long-term potentiation is associated with increases in quantal content and quantal amplitude, *Nature,* 357, 240-244.

[27] Kuo, L. (1986) Computations of mixtures of Dirichlet processes, *SIAM J. Sci. Stat. Comput.,* 7, 60-71.

[28] Lavine, M. (1990) Some aspects of Polya tree distributions for statistical modelling, *Discussion Paper #91-A01*, ISDS, Duke University.

[29] Lavine, M. (1991) More aspects of Polya tree distributions for statistical modelling, *Discussion Paper #91-A11*, ISDS, Duke University.

[30] Lavine, M. and West, M. (1992) A Bayesian Method for Classification and Discrimination, *The Canadian Journal of Statistics,* 20, 451-461.

[31] Lehmann, E. L. (1980) Efficient likelihood estimators. *Amer. Statistician,* 34, 233-235.

[32] Lehmann, E. L. (1983) *Theory of Point Estimation.* New York: Wiley.

[33] Ling, L. and Tolhurst, D. J. (1983) Recovering the parameters of finite mixtures of normal distributions from a noisy record: an empirical comparison of differing estimating procedures, *J. Neurosci. Methods,* 8, 309-333.

[34] Louis, T. A. (1982) Finding the observed information matrix when using the EM algorithm. *J. R. Statist. Soc. B,* 44, 226-233.

[35] MacEachern, S. N. (1992) Estimating normal means with a conjugate style Dirichlet process prior, *Technical Report # 487*, Department of Statistics, The Ohio State University.

[36] McClachlan, E. M. (1978) The statistics of transmitter release at chemical synapses, *Int. Review of Physiology, Neurophysiology III,* 17, eds. R. Porter, Baltimore: University Park Press, 49-117.

[37] McLachlan, G. J. and Basford, K. E. (1987) Mixture models-inference and applications to clustering, New York and Basel: Marcel Dekker.

[38] Müller, P. (1991) A generic approach to posterior integration and Gibbs sampling, preprint, Duke University.

[39] Naylor, J. C. and Smith, A. F. M. (1982) Applications of a method for the efficient computation of posterior distribution. *Appl. Statist.,* 31, 214-225.

[40] Peters, B. C. and Walker, H. F. (1978) An iterative procedure for obtaining maximum-likelihood estimates of the parameters for a mixture of normal distributions. *SIAM J. Appl. Math.,* 35, 362-378.

[41] Press, W. H., Flannery, B. P. , Teukolsky, S. A. and Vetterling, W. T. (1990) *Numerical Recipes in C,* Cambridge.

[42] Redman, S. J. (1989) Quantal analysis of synaptic potentials in neurones of the central nervous system, Physiol. Rev..

[43] Redman, S. J. (1990) Quantal analysis of synaptic potentials in neurones of the central nervous system, *Physiol. Rev.,* 70, 165-198.

[44] Redner, R. A. and Walker, H. F. (1984) Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review* 26, 195-239.

[45] Ripley, B. (1986) *Stochastic Simulation*, New York: John Wiley.

[46] Tierney, L., Kass, R. and Kadane, J. B. (1986) Approximation of expectations and variances using Laplace's method. Technical Report, Department of Statistics, Carnegie Mellon University.

[47] Titterington, D. M. and Mill, G. M. (1983) Kernel-based Density Estimates from Incomplete Data, *J. R. Statist. Soc.* B, 45, 258-266.

[48] Turner, D. A. and Schlieckert, M. (1990) Data acquisition and analysis system for intracellular neuronal signals, *J. Neurosci. Methods,* 35, 241-251.

[49] Turner, D. A. and West, M. (1993) Bayesian analysis of mixtures applied to postsynaptic potential fluctuations, *J. Neurosci. Methods,* 47, 1-21.

[50] Walmsley, B., Edwards, F. R., and Tracey, D. J. (1987) The probabilistic nature of synaptic transmission at a mammalian excitatory central synapse, *J. Neuroscience,* 7, 1037-1046.

[51] West, M. (1990) Bayesian kernel density estimation, *ISDS Discussion Paper #90-A02*, Duke University.

[52] West, M. (1992a) Modelling with mixtures (with discussion), in *Bayesian Statistics 4*, eds: J.O. Berger, J.M. Bernardo, A.P. Dawid and A.F.M. Smith, Oxford University Press (to appear).

[53] West, M. (1992b) Learning the precision parameter of a Dirichlet process, with applications to Bayesian density estimation. *ISDS Discussion Paper #92-A03*, Duke University.

[54] West, M. and Cao, G. (1992) Assessing mechanisms of neural synaptic activity, to appear *Bayesian Statistics in Sciences and Technology: Case Studies.*

[55] West, M. and Harrison, P. J. (1989) *Bayesian Forecasting and Dynamic Models,* New York, Springer Verlag.

[56] West, M., Harrison, P. J. and Pole, A. (1987) BATS: Bayesian Analysis of Time Series. *The Professional Statistician,* 6, 43-46.

[57] West, M., Müller P. and Escobar, M. D. (1993) Hierarchical priors and mixture models, with application in regression and density estimation, *Aspects of Uncertainty: A Tribute to D. V. Lindley,* eds: Smith, A. F. M. and Freeman, P. R, Wiley, London.

[58] West, M. and Turner, D. A. (1992) Deconvolution of mixtures in analysis of neural synaptic transmission. To appear *The Statistician.*

[59] Wong, K., and Redman, S. J. (1980) The recovery of a random variable from a noisy record with application to the study of fluctuations in synaptic potentials, *J. Neurosci. Methods,* 2, 389-409.

[60] Yakowitz, S. J. and Spragins, J. D. (1968) On the identifiability of finite mixtures. *Ann. Math. Statist.,* 39, 209-214.

Biography

The author was born on April 8, 1963 in Jiangsu, China. He received his Bachelor of Science degree in Applied Mathematics in 1985 from Southeast University (formerly Nanjing Institute of Technology) located in Nanjing, China and his Master of Science degree in Statistics in 1988 from the same university. The author taught Statistics at the Department of Mathematics, Southeast University from 1988 to 1989. In September, 1989 he entered the Graduate School of Duke University located in Durham, North Carolina in pursuit of a Doctor of Philosophy degree in Statistics.