2013-01-01

Comparison Of Bayesian Nonparametric Density Estimation Methods

Adel Bedoui
University of Texas at El Paso, abedoui@miners.utep.edu

COMPARISON OF BAYESIAN NONPARAMETRIC DENSITY ESTIMATION METHODS

ADEL BEDOUI

Department of Mathematics

APPROVED:

Ori Rosen, Chair, Ph.D.

Joan Staniswalis, Ph.D.

Martine Ceberio, Ph.D.

Benjamin C. Flores, Ph.D.
Dean of the Graduate School

COMPARISON OF BAYESIAN NONPARAMETRIC DENSITY ESTIMATION METHODS

by

ADEL BEDOUI

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Mathematics

THE UNIVERSITY OF TEXAS AT EL PASO

August 2013

Acknowledgements

My warmest and unreserved thanks go to my supervisor Dr. Ori Rosen, for his guidance and support. His vast knowledge of Bayesian computational methods (Markov Chain Monte Carlo) and his faith in me were vital for the completion of my Master.

I would also like to extend my appreciation to members of my committee: Dr. Joan Staniswalis and Dr. Martine Ceberio for their support, suggestions and being extremely supportive. I thank the department of Mathematics and Dr. Ori Rosen for the funding they provided through my Master degree program. I am grateful to all of the professors, graduate students, and staff in the Department of Mathematics, especially to Professor Mohamed Amine Khamsi.

I cannot forget to thank my friends for providing me with entrainement and companionship along the way, in particular Mohamed Abdoulah Khamsi. Lastly, and most importantly, I wish to thank my parents, my brothers and sister for their unconditional support during all phases of my life. To them I dedicate this thesis.

Abstract

Density estimation has a long history in statistics. There are two main approaches to density, estimation parametric and nonparametric. The first approach requires specification of a family of densities $f(\cdot|\theta)$ and estimation of the unknown parameter θ using a suitable estimation method, for example, maximum likelihood estimation. This approach may be prone to bias that arises from either estimation of the parameter or from incorrect specification of the probability distribution. The second approach, does not assume a specific parametric family.

In this thesis, we implement three density estimation methods that use Bayesian nonparametric approaches utilizing Markov Chain Monte Carlo methods. Specifically, these methods are the Dirichlet process prior, a method that converts density estimation to a regression problem, and a mixture of normal densities with known means and variances whose mixing weights are logistic with unknown parameters.

We briefly review two traditional methods that are used to obtain density estimates. The first is the density histogram which is one of the simplest and oldest methods. The second method is kernel estimation. In addition, we compare the three nonparametric methods by simulation and use them to estimate the density underlying the 1872 Mexican Hidalgo Stamp. The thesis concludes with a summary.

Table of Contents

Chapter 1

Introduction

1.1 Traditional Methods for Density Estimation.

1.1.1 Density Histogram

The histogram is considered one of the most widely used density estimators. It was first introduced in Pearson (1895). Suppose a density f has its support on the interval [a,b]. Let m be an integer and B_j be bins such that

$$B_1 = \Big[a, \frac{1}{m}\Big),\ B_2 = \Big[\frac{1}{m}, \frac{2}{m}\Big), ..., B_m = \Big[\frac{m-b}{m}, 1\Big).$$

The density histogram is defined by

$$\hat{f}_n(x) = \sum_{k=1}^{m} \frac{\hat{t}_j}{h} I(x \in B_j),$$

where $h = \dfrac{1}{m}$ is the bandwidth and $\hat{t}_j = \dfrac{\#\{X_i \in B_j\}}{n}$.

Example

Figure 1.1 is an example of a density histogram of a sample of size 400 simulated from a Chi-squared distribution on 6 degrees of freedom.

1.1.2 Kernel Estimator

Nonparametric kernel density estimation is a way of estimating a density function without assuming a standard parametric model. The kernel estimator was first introduced in Parzen (1962) and is given by

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K\Big(\frac{x - x_i}{h}\Big).$$

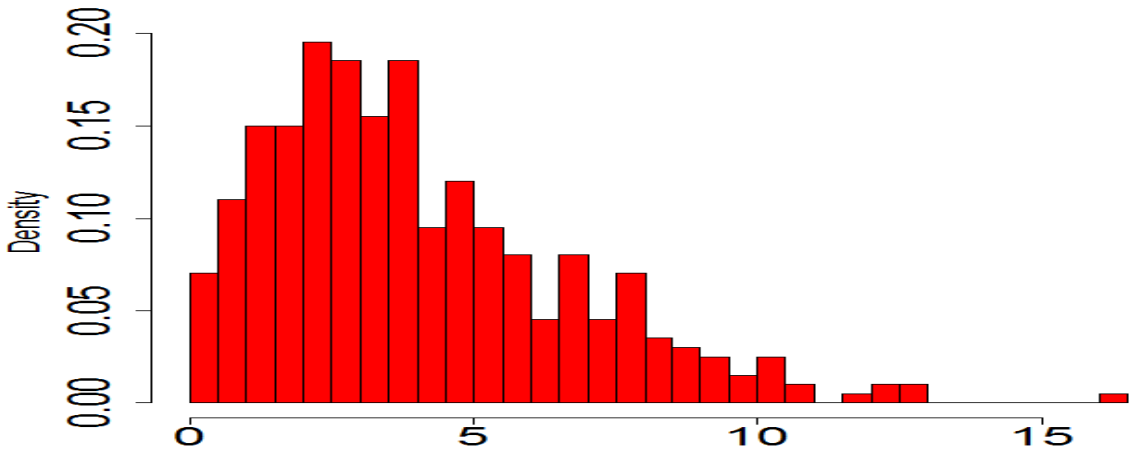

Figure 1.1: A density histogram based on a sample of size 400 from a Chi-squared distribution on 6 degrees of freedom.

• Usually K(·), the kernel function, is a symmetric pdf of a random variable with a finite second moment.

• h is the smoothing parameter or bandwidth.

Example

Figure 1.2 is an example of a kernel estimator fitted to a sample of size 1000 from $N(0,1)$ using three different bandwidths, $\frac{1}{2}$, 1 and 2.

1.2 Basic definitions

1.2.1 Bayes Theorem

Definition

Bayes theorem facilitates the calculation of posterior probabilities. Let θ be a parameter, and let X be a sample from $P(X|\theta)$. The posterior distribution of θ given X is

$$P(\theta|X) = \frac{P(\theta \cap X)}{P(X)} = \frac{P(X|\theta)P(\theta)}{P(X)}$$

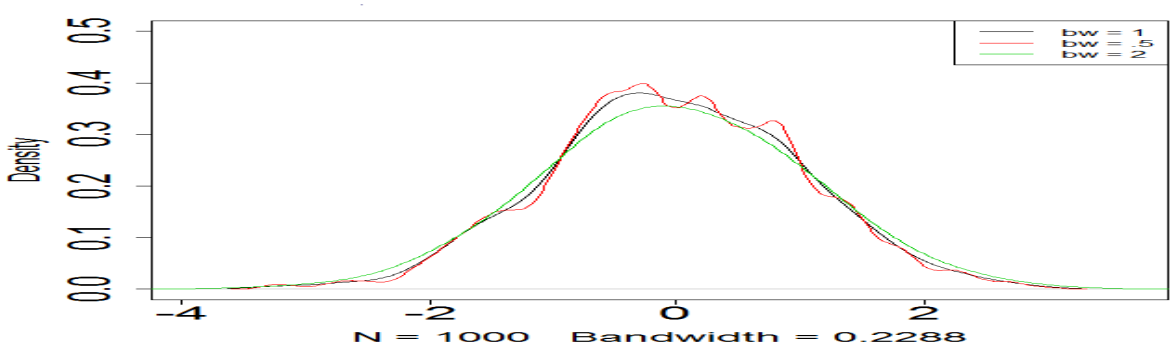

Figure 1.2: Kernel density estimates for a sample of size 1000 from N(0,1), using three different smoothing bandwidths.

where $P(X|\theta)$ is the likelihood and $P(\theta)$ is the prior on θ.

1.2.2 Hierarchical Bayes

In a Bayesian approach, a hierarchical model can be either fully parametric or semi-parametric.

A Fully Parametric Model

In a fully parametric hierarchical model, the i^{th} sample point, X_i, is sampled from a known probability density parametrized by θ_i. The prior π on θ_i is parametrized by γ. The hyper prior on γ has a density ψ. The form of a fully parametric model is

$$\begin{aligned} X_i|\theta_i &\sim f(X_i|\theta_i) \\ \theta_i|\gamma &\sim \pi(\theta_i|\gamma) \\ \gamma \quad &\sim \psi(\delta). \end{aligned}$$

A Semi-parametric Model

In a semi-parametric hierarchical model, the i^{th} sample point, X_i, comes from a probability density parametrized by θ_i. The parameter θ_i is generated from an unknown distribution

G which in turn comes from a Dirichlet process with parameters G_0 and α. The form of this hierarchical model is

$$\begin{aligned} X_i|\theta_i &\sim f(X_i|\theta_i) \\ \theta_i|G &\sim G \\ G \quad &\sim DP(G_0, \alpha). \end{aligned}$$

where G_0 is the base distribution and α is the concentration parameter.

1.2.3 Markov Chain Monte Carlo (MCMC) Methods

Markov Chain Monte Carlo (MCMC) methods enable integration problems in large dimensional spaces (Andrieu et al., 2003). There are two main methods, Gibbs Sampling and the Metropolis-Hasting algorithm.

Gibbs Sampling

Gibbs sampling is a MCMC method that allows us to obtain dependent samples from a posterior distribution when direct sampling is difficult. To sample from the posterior $P(\beta_1, ..., \beta_K|\boldsymbol{X})$ where $\boldsymbol{X} = (X_1, X_2, ..., X_K)$

1. Specify an initial value $\beta^0 = (\beta_1^0, ..., \beta_K^0)$

2. Repeat for $j = 1, 2, ..., K$:
   Generate β_1^{j+1} from p$(\beta_1|\beta_2^j, \beta_3^j, ..., \beta_K^j, \boldsymbol{X})$
   Generate β_2^{j+1} from $p(\beta_2|\beta_1^{j+1}, \beta_3^j, ..., \beta_K^j, \boldsymbol{X})$
   $\vdots$
   Generate β_K^{j+1} from $p(\beta_K|\beta_1^{j+1}, \beta_3^{j+1}, ..., \beta_{K-1}^{j+1}, \boldsymbol{X})$

3. Repeat (2) M times for some large number, M.

The Metropolis-Hasting Algorithm

The Metropolis-Hastings Algorithm (Metropolis et al., 1953) and (Hastings, 1970) follows the following steps:

1. Choose a starting value $\beta^{(0)}$.

2. At iteration j, draw a candidate β^* from a proposal density (jumping distribution), $Q(\beta^*|\beta^{(j-1)})$.

3. Compute the acceptance ratio

$$r = \frac{\pi(\beta^*)/Q(\beta^{(j-1)}|\beta^*)}{\pi(\beta^{(j-1)})/Q(\beta^*|\beta^{(j-1)})},$$

   where π is a stationary distribution.

4. Accept $\beta^{(j)} = \beta^*$ with probability $min(1, r)$. If β^* is not accepted, then $\beta^{(j)} = \beta^{(j-1)}$.

5. Repeat steps 2-4 N times to obtain N draws from $p(\boldsymbol{\beta}|y)$.

When an independence M-H step is used, i.e., when $Q(\beta^*|\beta^{(j-1)}) = Q(\beta^*)$, an optimization method is often used to obtain the proposal β^*. More details are given in Section 4.3.3.

1.2.4 Broyden-Fletcher-Goldfard-Shano (BFGS)

The BFGS method (Nocedal and Wright, 1999) is used for solving unconstrained nonlinear optimization problems. From an initial guess $\boldsymbol{\beta_0}$ and an approximate Hessian matrix H_0 the following steps are repeated until $\boldsymbol{\beta}$ converges to a solution.

1. Obtain a direction P_j by solving: $H_j P_j = -\nabla f(\boldsymbol{\beta_j})$

2. Perform a line search to find an acceptable step size α_j in the direction found in the first step, then update $\beta_{j+1} = \beta_j + \alpha_j P_j$

3. Set $S_j = \alpha_j P_j$

4. $\boldsymbol{y_j} = \nabla f(\boldsymbol{\beta}_{j+1}) - \nabla f(\boldsymbol{\beta_j})$

5. $H_{j+1} = H_j + \dfrac{\boldsymbol{y_j}\boldsymbol{y_j'}}{\boldsymbol{y_j'}S_j} - \dfrac{H_j S_j S_j' H_j}{S_j' H_j S_j}$,

where $f(\boldsymbol{\beta})$ is the objective function to be maximized.

1.2.5 Kullback Leibler Divergence (KLD)

The Kullback Leibler divergence is an asymmetric measure of the discrepancy between two probability density functions P and Q

$$D_{KL}(Q||P) = -\int \log\Big(\frac{Q(x)}{P(x)}\Big) \cdot P(x)dx, \text{ where}$$

- P is the true density
- Q is the estimated density

Chapter 2

The Dirichlet Process Prior

The Dirichlet process prior (DPP) defines a distribution over distributions, i.e. a draw from the DPP is a distribution. It is used to define a prior on an unknown probability distribution and is considered the basis of Bayesian nonparametrics. Its principal advantage comes from the simple form that the posterior distribution takes and can be represented in the following ways:

1. Chinese Restaurant Process
2. Pólya Urn Scheme
3. Stick Breaking Prior.

The main features of the Dirichlet process are summarized in this chapter. See Escobar (1994) and Escobar and West (1995) for further details.

2.1 The Dirichlet Process

We first introduce the definition of the Dirichlet distribution (Kotz et al., 2000).

The Dirichlet Distribution

The Dirichlet distribution of order $K \geq 2$ with parameters $\alpha_1, \alpha_2, ..., \alpha_K > 0$ has a probability density function

$$f(X_1, X_2, ..., X_{K-1}|\alpha_1, ..., \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} X_i^{\alpha_i - 1}, \quad \forall X_i \in [0, 1],$$

where $\sum_{i=1}^{K} X_i = 1$ *and* $B(\alpha) = \dfrac{\Gamma(\alpha_1, \alpha_2, ..., \alpha_K)}{\Gamma(\alpha_1)\Gamma(\alpha_2)...\Gamma(\alpha_K)}$.

The mean and variance of the Dirichlet distribution are

$$E(X_i) = \frac{\alpha_i}{\sum_{j=1}^{K} \alpha_j} \tag{2.1}$$

and

$$\text{var}(X_i) = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} \text{ where } \alpha_0 = \sum_{j=1}^{K} \alpha. \tag{2.2}$$

Ferguson (1973) introduced the Dirichlet process as a probability measure on the space of all probability measures.

2.1.1 Definition

Let $(\mathcal{X}, \mathcal{A})$ *be a measurable space,* G_0 *a probability measure on* $(\mathcal{X}, \mathcal{A})$ *and* α *a real positive number. We say that* G *is a Dirichlet process distributed with base* G_0 *and concentration parameter* α *if for any partition* $A_1, A_2, ...A_r$ *of* $\mathcal{X}$*, the vector of random probabilities* $G(A_1), G(A_2), ...G(A_r)$ *follows a Dirichlet distribution*

$$(G(A_1), G(A_2), ...G(A_r)) \sim \mathcal{D}(\alpha G_0(A_1), \alpha G_0(A_2), ..., \alpha G_0(A_r)), \tag{2.3}$$

where $\mathcal{D}(\alpha_1, \alpha_2, .., \alpha_r)$ *denotes the Dirichlet distribution with parameters* $\alpha_1, \alpha_2, ..., \alpha_r$*. For simplicity, we denote*

$$G \sim \mathcal{DP}(G_0, \alpha).$$

From equations (2.1) and (2.2), for all $B \in \mathcal{A}$

$$\begin{aligned} \text{E}[G(B)] &= G_0(B) \\ \text{var}[G(B)] &= \frac{G_0(B)(1 - G_0(B))}{1 + \alpha} \end{aligned}.$$

2.2 Finite Mixture Models

Let $\boldsymbol{X} = \{X_1, X_2, ..., X_n\}$ be a random sample of size n, K be the the number of components, and z_i be an unobservable indicator variable for X_i where $z_i \in \{1, 2, ..., K\}$. Let $\boldsymbol{\pi} = \{\pi_1, \pi_2, ..., \pi_K\}$ be a vector of probabilties where

$$p(z_i = j) = \pi_j \tag{2.4}$$

is the probability that X_i comes from component j, $j = 1, 2, .., K$, with $\sum_{j=1}^{K} \pi_j = 1$. The density function of an observation X_i coming from the j^{th} component is

$$P(X_i|\phi_j) = P(X_i|z_i = j). \tag{2.5}$$

Using equations (2.4) and (2.5), the marginal distribution is

$$\begin{aligned} P(X_i) &= \sum_{j=1}^{K} P(X_i|z_i = j)P(Z_i = j) \\ &= \sum_{j=1}^{K} P(X_i|\phi_j)\pi_j. \end{aligned}$$

The number of indicator variables is at most n, whereas the number of components can

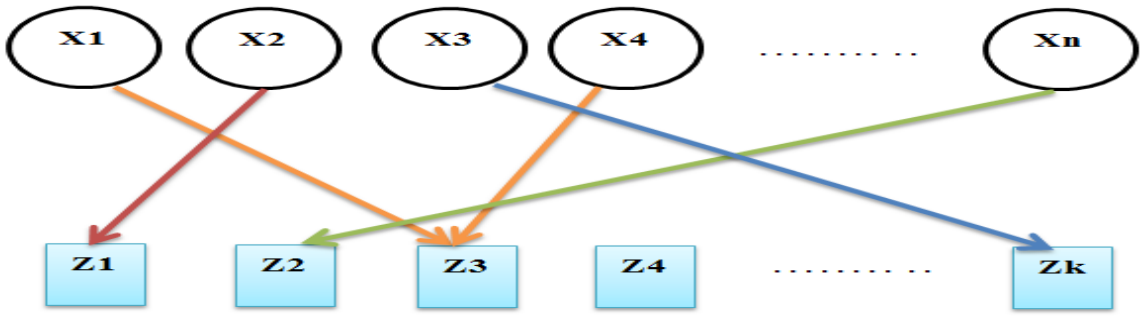

Figure 2.1: Each observation X_i comes from a specific component Z_i.

be infinite. The distribution of z_l is assumed to be multinomial

$$z_l|\boldsymbol{\pi} \sim \text{Multinomial}(1, \boldsymbol{\pi}). \tag{2.6}$$

The prior on $\boldsymbol{\pi}$ is a Dirichlet distribution with parameter $\frac{\alpha}{K}$

$$\boldsymbol{\pi}|\alpha \sim \text{Dirichlet}(\frac{\alpha}{K}, \frac{\alpha}{K}, ..., \frac{\alpha}{K}), \tag{2.7}$$

where α is a concentration parameter.

Claim: The marginal distribution of the indicators is given by

$$P(\boldsymbol{z}|\alpha) = \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \prod_{j=1}^{K} \frac{\Gamma(n_j + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})}, \tag{2.8}$$

where $n_j = \#\{z_i = j\}$ and $n = \sum_{j=1}^{K} n_j$.

Proof: From (2.7)

$$P(\boldsymbol{\pi}|\alpha) = \frac{\Gamma(\sum_{j=1}^{K} \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})^K} \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1}. \tag{2.9}$$

Using equations (2.6) and (2.9) and integrating out $\boldsymbol{\pi}$

$$\begin{aligned}
P(\boldsymbol{z}|\alpha) &= \int P(\boldsymbol{z}|\boldsymbol{\pi}, \alpha) P(\boldsymbol{\pi}|\alpha) d\boldsymbol{\pi} \\
&= \int \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{K})^K} \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1} \prod_{j=1}^{K} \pi_j^{n_j} d\boldsymbol{\pi} \\
&= \int \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{K})^K} \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1+n_j} d\boldsymbol{\pi} \\
&= \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{K})^K} \int \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1+n_j} d\boldsymbol{\pi} \\
&= \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{K})^K} \frac{\prod_{j=1}^{K} \Gamma(n_j + \frac{\alpha}{K})}{\Gamma(\sum_{j=1}^{K} (n_j + \frac{\alpha}{K}))} \int \frac{\Gamma(\sum_{j=1}^{K} n_j + \frac{\alpha}{K})}{\prod_{j=1}^{K} \Gamma(n_j + \frac{\alpha}{K})} \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1+n_j} d\boldsymbol{\pi} \\
&= \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \prod_{j=1}^{K} \frac{\Gamma(n_j + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})} \\
&= \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{K})^K} \frac{\prod_{j=1}^{K} \Gamma(n_j + \frac{\alpha}{K})}{\Gamma(n+\alpha)}
\end{aligned}$$

■

2.2.1 Posterior Predictive Distribution

A posterior predictive distribution is the distribution of a new data point assigned to the k^{th} component given that $(i-1)$ data points have already been allocated to their respective clusters. From Equation (2.8), the posterior predictive distribution is

$$P(z_i = k|z_1, z_2, ..., z_{i-1}, \alpha) = \frac{n_{ik} + \frac{\alpha}{K}}{i - 1 + \alpha} \ , \tag{2.10}$$

where n_{ik} is the number of data points already allocated to the k^{th} component.

2.3 Infinite Mixture Models

One advantage of the Dirichlet process prior is that it does not require a selection of the number of clusters, because it is a prior over an infinite number of components. From Equation (2.10) and by taking $K \to \infty$, the predictive distribution of z_i for an infinite mixture is

$$P(z_i = k|\boldsymbol{z}_{-i}, \alpha) = \begin{cases} \frac{n_{ik}}{i - 1 + \alpha}, & \text{if the } k^{th}\text{component is occupied} \\ \frac{\alpha}{i - 1 + \alpha}, & \text{if the } k^{th}\text{component is unoccupied} \end{cases} \ ,$$

where $\boldsymbol{z}_{-i} = \{z_1, z_2, ..., z_{i-1}, z_{i+1}, ...\}$.

Until now, we have discussed only the distribution of the indicators. Concerning the parameter θ_i of a data point X_i, the conditional distribution was given by Blackwell and MacQueen (1973) as

$$\theta_i = \phi_{z_i}|\theta_1, ..., \theta_{i-1} \sim \frac{1}{i - 1 + \alpha} \sum_{j=1}^{i-1} \delta_{\theta_j} + \frac{\alpha}{i - 1 + \alpha} G_0 \ , \tag{2.11}$$

where ϕ_{z_i} represents the parameter of cluster z_i and δ_{θ_j} is a point mass distribution at θ_j. The parameters $\phi_1, ..., \phi_K$ are unique, whereas the θs may not be unique. The first term

on the right-hand side of (2.11) is the probability that a new data point is assigned to an existing component. The second term is the probability that the data point is assigned to a new component.

2.4 Representations of The Dirichlet process

In this section we will discuss three equivalent representations of the DPP. These processes are the Chinese Restaurant Process, the Pólya Urn Scheme and the Stick Breaking Prior.

2.4.1 The Chinese Restaurant Process

The Chinese Restaurant Process (CRP) was introduced in Aldous (1985). Figure 2.2 shows a restaurant with infinitely many tables labeled 1,2,.... . The CRP is a random process where the tables and customers represent the components and data points respectively. Each table k is characterized by its own unique parameter value ϕ_k drawn from a base distribution G_0. The parameter of customer i, θ_i, may not be unique. A customer walks in and sits down at some table, either an unoccupied or an occupied one. The first customer always chooses the first table. The $(i+1)^{th}$ customer chooses an unoccupied table with probability $\frac{\alpha}{i-1+\alpha}$ or an occupied table with probability $\frac{n_{ik}}{i-1+\alpha}$, where n_{ik} is the number of customers sitting at the k^{th} table. The parameter α is the concentration parameter and determines how likely a customer is to sit at unoccupied table. The predictive distribution of θ_i under the CRP is

$$\Big(\theta_i|\theta_1,...,\theta_{i-1},G_0,\alpha\Big) \sim \frac{\alpha}{i-1+\alpha}G_0 + \frac{\sum_{k=1}^{i-1} n_{ik}\delta_{\phi_k}}{i-1+\alpha} \quad .$$

Example:

A possible arrangement of 12 customers is shown in Figure 2.3. A seating arrangement of 12 customers creating 4 groups (1,3,8), (2,5,9,10), (4,11,12) and (6,7).

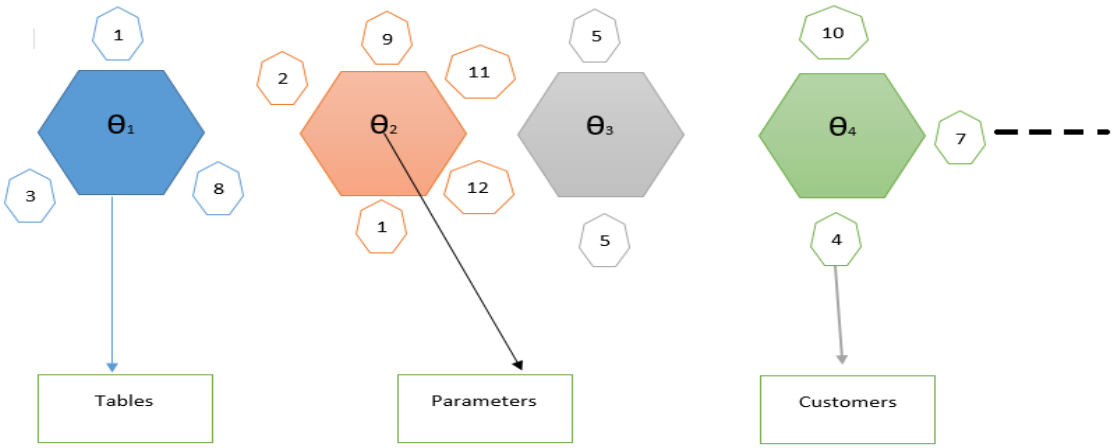

Figure 2.2: Illustration of the Chinese Restaurant Process.

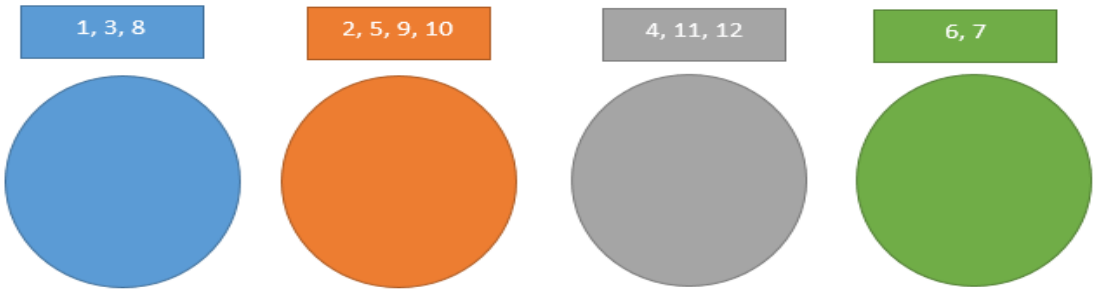

Figure 2.3: Circles represent tables and the numbers are the customers sitting at a particular table.

Denote the table occupied by customer i by z_i, $z_i \in \{1, 2, 3, 4\}$. Then the probability of

this arrangement is

$$
\begin{aligned}
P(z_1, z_2, ..., z_{12}) &= P(z_1)P(z_2|z_1)...P(z_{12}|z_1, z_2, ...z_{11}) \\
&= \Big(\frac{\alpha}{\alpha}\Big)\Big(\frac{\alpha}{1+\alpha}\Big)\Big(\frac{1}{2+\alpha}\Big)\Big(\frac{\alpha}{3+\alpha}\Big)\Big(\frac{1}{4+\alpha}\Big)\Big(\frac{1}{5+\alpha}\Big)\Big(\frac{1}{6+\alpha}\Big)\Big(\frac{2}{7+\alpha}\Big) \\
&\quad \Big(\frac{2}{8+\alpha}\Big)\Big(\frac{3}{9+\alpha}\Big)\Big(\frac{1}{10+\alpha}\Big)\Big(\frac{2}{11+\alpha}\Big).
\end{aligned}
$$

A given seating arrangement induces a partition of customers into tables. For example, the probability of the partition (1,8), (3,4,10), (2,5,11,12) and (9,6,7) is identical to the probability of the above partition. This property is called *exchangeability*.

2.4.2 The Pólya Urn Scheme

The Pólya urn scheme, also known as the Blackwell-MacQueen Urn scheme, is a process characterized by exchangeable random variables (Blackwell and MacQueen, 1973). Consider an urn with V balls, of which α_j are of color j, where $1 \leq j \leq k$. We draw balls at random from the urn and then replace each ball drawn with two balls of the same color (Hoppe, 1984), see Figure 2.4.

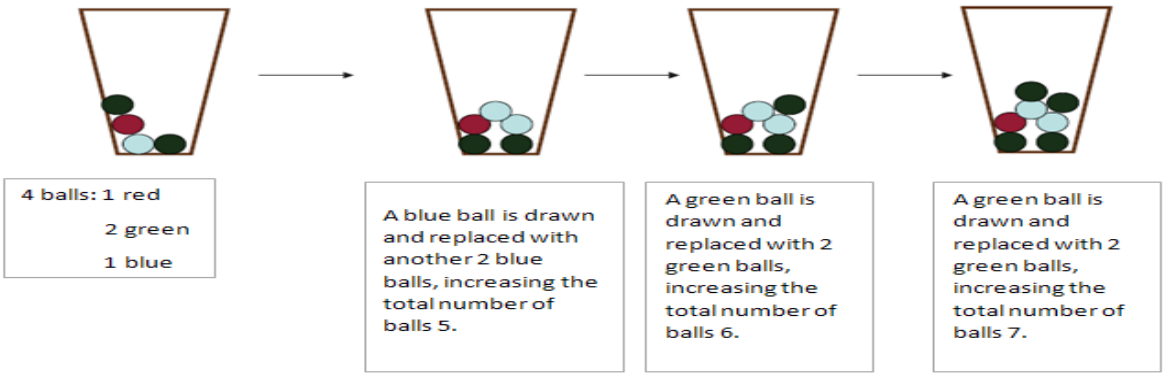

Figure 2.4: Pólya urn scheme.

Let $X_i = j$ if the i^{th} ball is of color j, then:

$$\begin{aligned} p(X_1 = j) &= \frac{\alpha_j}{V} \\ p(X_2 = j|X_1) &= \frac{\alpha_j + \delta_{X_1,j}}{V+1} \\ &\vdots \\ p(X_{n+1} = j|X_1, X_2, ..., X_n) &= \frac{\alpha_j + \sum_{i=1}^{n} \delta_{X_{i,j}}}{V+n}, \end{aligned}$$

where $\delta_{X_{i,j}} = j$ if $X_i = 1$ and 0 otherwise.

2.4.3 The Stick Breaking Prior

The stick breaking prior is another representation of the Dirichlet process prior and was first introduced in Sethuraman (1994). It is defined as follows.

$$\begin{aligned} \pi_1 &= v_1 \\ \pi_k &= v_k \prod_{j=1}^{K-1} (1 - v_j),\ k = 2, ..., K \\ \sum_{k=1}^{K} \pi_k &= 1 \\ v_k &\overset{iid}{\sim} \text{Beta}(1, \alpha), \end{aligned} \tag{2.12}$$

where π_k are random weights.

By the construction presented in Equation (2.12) and Figure 2.5, Sethuraman (1994) showed that

$$\sum_{k=1}^{\infty} \pi_k = 1,\ \theta_k \sim G_0,\ G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}. \tag{2.13}$$

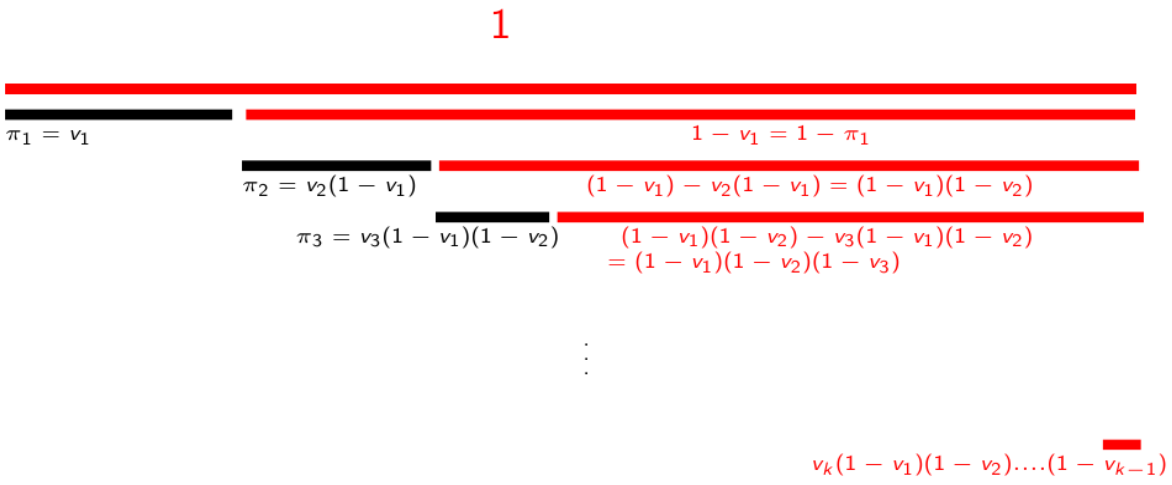

Figure 2.5: An illustration of the stick-breaking process.

2.5 Estimation

Given a sample $X_1, X_2, ..., X_n$, our model can be written as follows

$$X_i|(\theta_i = \{\mu_i, \sigma_i^2\}) \sim N(\mu_i, \sigma_i^2)$$
$$\theta_i|G \sim G$$
$$G \sim DP(G_0, \alpha),$$

for a mixture of normal with parameters $\theta_i = \{\mu_i, \sigma_i^2\}$. The augmented likelihood is

$$L(\boldsymbol{X}, \boldsymbol{z}, u_{z_i}, \sigma_{z_i}^2) = \prod_{i=1}^{n} \pi_{z_i} \frac{1}{\sqrt{2\pi\sigma_{z_i}^2}} \exp\Big\{-\frac{(X_i - \mu_{z_i})^2}{2\sigma_{z_i}^2}\Big\}, \tag{2.14}$$

where the z_i's are unobservable indicators taking values in $\{1, 2, ..., K\}$ (Ishwaran and Lancelot, 2002).

2.5.1 Prior Distributions

The following priors are placed on the parameters, see Ishwaran and Zarepour (2000), Ishwaran and Lancelot (2001) and Ishwaran and Lancelot (2002).

1. $\mu_k|\theta, \sigma_\mu \sim N(\theta, \sigma_\mu)$

2. $\tau_k^2|\nu_1, \nu_2 \sim \text{InvGamma}(\nu_1, \nu_2)$

3. $\alpha|n_1, n_2 \sim \text{Gamma}(n_1, n_2)$

4. $\theta \sim N(0, A)$

5. The prior on the weights is given by Equation (2.12).

2.5.2 Gibbs Sampling

To sample from the posterior $P(\boldsymbol{\mu}, \boldsymbol{\tau^2}, \boldsymbol{z}, \alpha, \theta|\boldsymbol{X})$, where $\boldsymbol{\mu} = (\mu_1, \mu_2, ..., \mu_K)$ and $\boldsymbol{\tau} = (\tau_1, \tau_2, ..., \tau_K)$, we draw from the following conditional distributions.

1.
$$(\mu_j|\boldsymbol{\tau^2}, \boldsymbol{z}, \boldsymbol{X}, \theta) \overset{ind}{\sim} N(\mu_j^*, \sigma_j^*), \tag{2.15}$$

where

$$\mu_j^* = \sigma_j^*(\sum_{i:z_i=j} X_i/\tau_j^2 + \theta/\sigma_\mu)$$
$$\text{and } \sigma_j{}^{*2} = (n_j/\tau_j^2 + 1/\sigma_\mu^2)^{-1}.$$

2.
$$(\tau_j^2|\boldsymbol{\mu}, \boldsymbol{z}, \boldsymbol{X}) \overset{ind}{\sim} \text{InvGamma}(\nu_1 + n_j/2, \nu_{2,j}^*), \tag{2.16}$$

where

$$\nu_{2,j}^* = \nu_2 + \sum_{i:z_i=j} (X_i - \mu_j)^2/2$$
$$\text{and } n_j = \#\{z_i = j\}.$$

3.
$$(z_i|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{X}) \overset{ind}{\sim} \sum_{j=1}^{K} \pi_{j,i},$$

where

$$(\pi_{1,i}, ..., \pi_{K,i}) \propto \Big(\pi_1 \text{exp}\Big\{-\frac{1}{2\tau_1^2}(X_i - \mu_1)^2\Big\}, ..., \pi_K \text{exp}\Big\{-\frac{1}{2\tau_K^2}(X_i - \mu_K)^2\Big\}\Big).$$

4.

$$\pi_1 = \beta_1^* \text{ and } \pi_j = (1 - \beta_1^*)(1 - \beta_2^*)....(1 - \beta_{j-1}^*)\beta_j^*, \ j = 2, ..., K - 1, \tag{2.17}$$

where

$$\beta_j^* \overset{ind}{\sim} \text{Beta}(1 + r_j, \alpha + \sum_{l=j+1}^{K} r_l) \text{ and}$$

$$r_j = \text{The number of observations in cluster } j.$$

5.

$$(\alpha|\boldsymbol{\pi}) \sim \text{Gamma}\Big(K + n_1 - 1, n_2 - \sum_{j=1}^{K-1} \log(1 - \beta_j^*)\Big). \tag{2.18}$$

6.

$$(\theta|\boldsymbol{\pi}) \sim N(\theta^*, \sigma^*), \tag{2.19}$$

where

$$\theta^* = \frac{\sigma^*}{\sigma_\mu} \sum_{j=1}^{K} \mu_j \text{ and}$$

$$\sigma^* = (K/\sigma_\mu + 1/A)^{-1}.$$

Chapter 3

The Regression Approach to Density Estimation

The regression method converts density estimation to a regression problem and then applies a smoothing method to estimate the regression function. In this thesis we use cubic smoothing splines to estimate the regression. The idea was proposed by (Lindsey, 1974) in a frequentist approach and has recently been formulated in a Bayesian model by Brown and Zhang (2010).

3.1 Description of the Method

3.1.1 Converting Density Estimation to Regression

The method of converting density estimation to a nonparametric regression problem is described by Brown and Zhang (2010) as follows. Suppose we have a random sample $X_1, X_2, ...X_n$ from a density f on [a,b]. We Divide the interval [a,b] into k bins with equal widths as depicted in Figure 3.1. Let B_j denote the j^{th} bin. The number of observations in bin j are denoted by N_j and the abscissa of the center of B_j by t_j. Let d denote the bin width, i.e, $d = \dfrac{b-a}{k}$ (see (Wasserman, 2006) for more details).

3.1.2 The Number of Bins, K

The Poisson distribution is a natural exponential family with a quadratic variance function (NEF-QVE). This comes from the fact that the variance of this distribution equals its

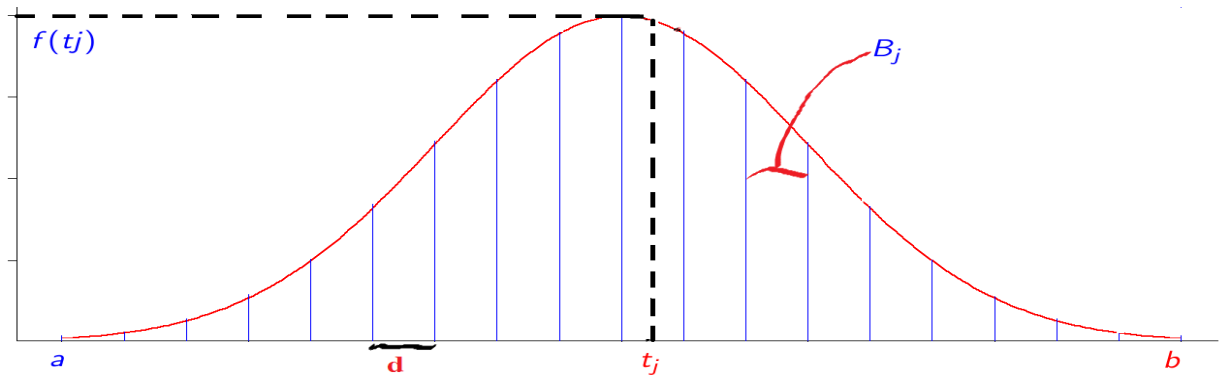

Figure 3.1: Converting density estimation to regression

mean. Hence, the variance is a linear function of the mean. The number of bins for a NEF-QVE is $n^{3/4}$. This choice is determined by the bounds for the approximation error, the discretization error, and the stochastic error (Brown et al., 2010)

3.1.3 Root Transformation

We assume that N_j has an approximate Poisson distribution with mean

$$E(N_j) = n \cdot d \cdot f(tj).$$

Claim: Let $Y_j = g(N_j) = \sqrt{\frac{k(b-a)}{n}} N_j$, then $E(Y_j) \approx (b-a)\sqrt{f(t_j)}$ and $Var(Y_j) \approx \frac{k(b-a)}{4n}$.

Proof: A first order-order Taylor approximation of $g(N_j)$ around $E(N_j)$ gives

$$g(N_j) \approx g(E(N_j)) + g'(E(N_j))(N_j - E(N_j)). \tag{3.1}$$

By taking expectations of both sides, we see that

$$E(Y_j) \approx g(E(N_j)). \tag{3.2}$$

Taking variances of both sides of (3.1), we obtain

$$\text{var}(Y_j) \approx [g^{'}(E(N_j))]^2 \text{var}(N_j). \tag{3.3}$$

Using (3.2) shows that $E(Y_j) \approx \sqrt{\frac{k(b-a)}{n} \cdot \frac{n(b-a)f(t_j)}{k}} = (b-a)\sqrt{f(t_j)}$.

Let $C \equiv \sqrt{\frac{k(b-a)}{n}}$, then $g(N_j) = C\sqrt{N_j}$ and $g'(N_j) = \frac{C}{2\sqrt{N_j}}$.

From (3.3), $\text{var}(Y_j) \approx \frac{C^2}{4 \cdot E(N_j)} \cdot \text{var}(N_j) = \frac{k(b-a)}{4n}$, since $E(N_j) = \text{var}(N_j)$.

Note that Brown and Zhong (2010) define Y_j as $Y_j = \sqrt{\frac{k(b-a)}{n}}\sqrt{N_j + \frac{1}{4}}$.

Using these results, we can write

$$Y_j \approx (b-a)r(t_j) + \sigma\epsilon_j, \tag{3.4}$$

where $\epsilon_j \sim N(0,1), r(t_j) = \sqrt{f(t_j)}$ and $\sigma = \sqrt{\frac{k(b-a)}{4n}}$.

■

3.2 Estimation of The Regression Function

3.2.1 Cubic Smoothing Splines

Theorem (Silverman, 1986)

Given $a \leq t_1 \leq t_2 \leq .... \leq t_n$, a function g is a cubic spline if

- On each interval $(a, t_1), (t_1, t_2), , (t_n, b)$, g is a cubic polynomial.
- The polynomial pieces fit together at points t_i (called knots) such that g itself and its first and second derivatives are continuous at each t_i, and hence on the whole interval [a,b].

Applying Cubic Smoothing Splines to Estimate the Regression Function

We now estimate the function $r(t)$ in Equation (3.4) by cubic smoothing splines using Wahba (1990)'s approach.

We let

$$r(t) = \beta_0 + \beta_1 t + h(t),$$

where $\boldsymbol{h} = (h(t_1), h(t_2), ..., h(t_n))'$ is a zero-mean Gaussian process with variance covariance matrix $\tau^2\Omega$, with the ijth element of Ω given by

$$\Omega_{ij} = \frac{1}{2}t_i^2(t_j - \frac{t_i}{3}),\ t_i < t_j.$$

To facilitate the computation, we write $h = Z\boldsymbol{u}$ where Z is obtained as follows. The matrix Ω is expressed as $\Omega = QDQ'$, where Q is the matrix of eigenvectors of Ω and D is a diagonal matrix containing the eigenvalues of Ω. Letting $Z = QD^{1/2}$ and setting the prior on $\boldsymbol{u}$ to be $\mathrm{N}(0, \tau^2 I_n)$ means that $Z\boldsymbol{u} \sim N(0, \tau^2\Omega)$. The parameter τ^2 is a smoothing parameter controlling the smoothness of $r(t)$.

Prior Distributions

The following priors are placed on the parameters:

1. $\boldsymbol{\beta} \sim N(\mathbf{0}, \sigma_\beta^2 I_2)$, where σ_β^2 is a large fixed number.
2. $\boldsymbol{u} \sim N(\mathbf{0}, \tau^2 I_m)$.
3. $\tau^2 \sim U(0, c_{\tau^2})$.
4. $\sigma^2 \sim U(0, c_{\sigma^2})$.

The model can thus be written as:

$$\boldsymbol{Y} = X\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon},$$

where

$$\mathrm{X}= \begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ . & . \\ . & . \\ 1 & t_n \end{pmatrix}.$$

The number of columns of Z is reduced from n to m $(m < n)$ by retaining only the m columns, corresponding to the m largest eigenvalues of Ω without affecting the fit.

3.2.2 Gibbs Sampling

To sample from the posterior $p(\boldsymbol{\beta}, \boldsymbol{u}, \tau^2, \sigma^2 | \boldsymbol{Y}, Z, X)$, we draw from the following conditional distributions

1.

$$(\boldsymbol{\beta}', \boldsymbol{u}') \sim N\Big(\frac{1}{\sigma^2}(\sigma^2(X^{*'}X^* + \sigma^2 A^{-1})^{-1})X^{*'}\boldsymbol{Y},\ \sigma^2(X^{*'}X^* + \sigma^2 A^{-1})^{-1}\Big),$$

where $X^* = \Big(X|Z\Big)$, i.e., where X and Z are concatenated columnwise, and $A = \text{diag}(\sigma_\beta^2, \sigma_\beta^2, \tau^2, \tau^2, ..., \tau^2)$

2.

$$\sigma^2 \sim IG\Big(\frac{n}{2} - 1, \frac{1}{2}(Y - X^*\beta^*)'(Y - X^*\beta^*\Big) \cdot I(0 \leq \sigma^2 \leq C_{\sigma^2}), \tag{3.5}$$

i.e., a truncated IG distribution.

3.

$$\tau^2 | u \sim IG\Big(\frac{m}{2} - 1, \frac{1}{2}u'u\Big).I(0 \leq \tau^2 \leq C_{\tau^2}). \tag{3.6}$$

3.2.3 Density Estimation Through Regression

After estimating $r(t)$, $\hat{f}(t_j)$ is found by unrooting and normalizing $r^+(t)$

$$\hat{f}(t_j) = \frac{(r^+(t_j))^2}{\int_{D_f}(r^+(s))^2 ds}.$$

Chapter 4

Mixture of Normals with Known Components

4.1 Description of the Method

To estimate a density, we first divide [a,b] into a grid of equally spaced points $(u_1, u_2, ..., u_K)$. We make these points the means of normal distributions, each with variance $\sigma^2 = \frac{1}{3}(u_{i+1} - u_i) = \frac{d}{3}$, Ghidey et al. (2004). This method assumes that a density f can be approximated by the following mixture

$$f(X) = \sum_{j=1}^{K} c_j \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big\{-\frac{(X-\mu_j)^2}{2\sigma^2}\Big\},$$

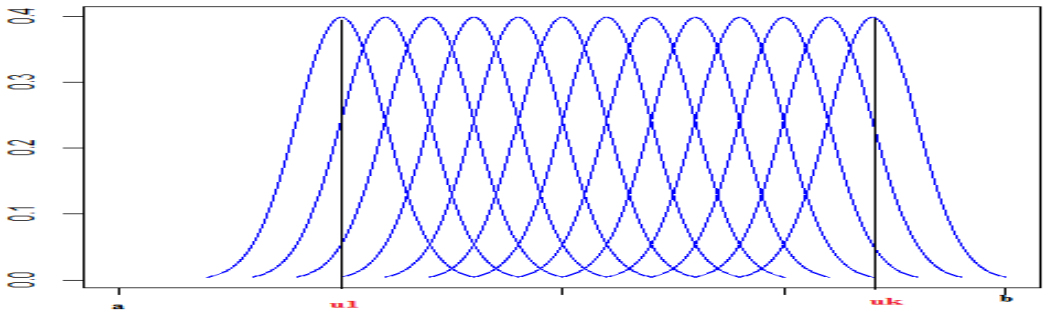

Figure 4.1: Illustration of overlapping Normal densities.

where

$$c_j = \frac{\exp(\beta_j)}{\sum_{j=1}^{K} \exp(\beta_j)}$$

such that $\sum_{j=1}^{K} c_j = 1$. The β_j are unknown parameters, and β_1 is set to zero for identifiability.

Smoothing Prior

Lang and Brezger (2004) used the following prior on $\boldsymbol{\beta} = (\beta_2, ..., \beta_K)'$ which allows the c_js to vary smoothly.

$$P(\boldsymbol{\beta}|\tau^2) = P(\beta_2)P(\beta_3)\Big(\frac{1}{\sqrt{2\pi\tau^2}}\Big)^{K-3}\exp\Big(-\frac{1}{2\tau^2}\sum_{p=4}^{K}(\beta_p - 2\beta_{p-1} + \beta_{p-2})^2\Big). \tag{4.1}$$

This is analogous to the penalty used by Eilers and Marx (1996). The expression $\sum_{p=4}^{K}(\beta_p - 2\beta_{p-1} + \beta_{p-2})^2$ in Equation (4.1) can be written in matrix form as

$$\boldsymbol{\beta}' P \boldsymbol{\beta}, \tag{4.2}$$

where $P = D'D$ is defined as a precision matrix, and D is

$$D = \begin{bmatrix} 1 & 0 & 0 & . & 0 \\ -2 & 1 & 0 & . & 0 \\ 1 & -2 & 1 & . & 0 \\ . & . & . & . & . \\ . & . & . & . & 1 \\ . & . & . & . & -2 \\ 0 & . & . & . & 1 \end{bmatrix}.$$

Unser et al. (1992) showed that a standardized B-spline of degree q approximates a normal density as $q \to \infty$. To learn more about B-spline see De Boor (1978) and Dierckx (1995).

The prior $P(\boldsymbol{\beta}|\tau^2)$ can be improper if the precision matrix P is not of full rank. The deficiency of P arises when the priors on β_2 and β_3 are flat ($P(\beta_2) \propto 1$ and $P(\beta_3) \propto 1$).

An Improper prior is a prior distribution that does not integrate to 1. There are many approaches that can make the matrix P full rank . For instance, Panagiotelis and Smith (2008) imposed two additional restrictions. In our thesis, we use normal priors on β_2 and β_3 to make the matrix P full rank (Chib and Jeliazkov, 2006).

$$(\beta_2, \beta_3)' \sim N(\mathbf{0}, c\tau^2 \mathrm{I}_2). \tag{4.3}$$

From (4.1), (4.2) and (4.3), the prior on $\boldsymbol{\beta}$ becomes

$$P(\boldsymbol{\beta}|\tau^2) \propto (\tau^2)^{-\frac{K-1}{2}} \exp(-\frac{1}{2\tau^2}\boldsymbol{\beta}' P^* \boldsymbol{\beta}) \tag{4.4}$$

where $P^*_{i,j} = \begin{cases} P_{i,j} + \dfrac{1}{c}, \text{ if } i = j = 1 = 2 \\ P_{i,j}, \quad \text{otherwise} \end{cases}$ and c is an integer.

4.2 Estimation

Given a sample $X_1, X_2, ..., X_n$ we can write the augmented likelihood as

$$f(\boldsymbol{x}) = \prod_{i=1}^{n} c_{z_i} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big\{-\frac{(X-\mu_{z_i})^2}{2\sigma^2}\Big\}, \tag{4.5}$$

where z_i's are unobservable indicators taking values in $\{1, 2, ..., K\}$.

4.2.1 Priors:

In addition to the prior (4.1) on $\boldsymbol{\beta}$ let the prior on τ^2 be an inverse gamma distribution with parameters a and b.

Using (4.4), (4.5), and the prior on τ^2

$$\begin{aligned} P(\boldsymbol{\beta}, \boldsymbol{z}, \tau^2|\boldsymbol{X}) \propto & \prod_{i=1}^{n} c_{z_i} f(X_i|\mu_{z_i}, \sigma^2) \\ & \times (\tau^2)^{-\frac{k-1}{2}} \exp\Big(-\frac{1}{2\tau^2}\boldsymbol{\beta}' P^* \boldsymbol{\beta}\Big) \\ & \times (\tau^2)^{-(a+1)} \exp\Big(-\frac{b}{\tau^2}\Big). \end{aligned} \tag{4.6}$$

4.2.2 Sampling Scheme:

1. From (4.6), $P(\tau^2|\boldsymbol{\beta}) \propto (\tau^2)^{-(\frac{K-1}{2}+a+1)}\exp\Big\{-\frac{1}{\tau^2}(b+\frac{1}{2}\boldsymbol{\beta}'P^*\boldsymbol{\beta})\Big\}$, which means that

$$(\tau^2|\boldsymbol{\beta}) \sim IG\Big(a+\frac{K-1}{2}, b+\boldsymbol{\beta}'P^*\boldsymbol{\beta}\Big)$$

.

2. The conditional distribution of $\boldsymbol{\beta}$ is

$$P(\boldsymbol{\beta}|\tau^2, \boldsymbol{Z}) \propto \prod_{j=1}^{K} c_j^{n_j}\exp\Big\{-\frac{1}{2\tau^2}\boldsymbol{\beta}'P^*\boldsymbol{\beta}\Big\}, \tag{4.7}$$

where $n_j = \#\Big\{Z_i = j\Big\}$.

This is a nonstandard distribution. We use a Metropolis-Hasting step to sample from it, see Section 4.2.3.

3. The indicators are sampled one at a time from a multinomial distribution $M(1, h_{i1}, ..., h_{iK})$, where

$$h_{ij} = \frac{c_j \cdot \exp\Big\{-\frac{(X_i-\mu_j)^2}{2\sigma^2}\Big\}}{\sum_{k=1}^{K} c_k \cdot \exp\Big\{-\frac{(X_i-\mu_k)^2}{2\sigma^2}\Big\}}.$$

To sample from the posterior $p(\boldsymbol{\beta}, \tau^2|\sigma^2, \boldsymbol{X}, \boldsymbol{z})$ we draw from the following conditional distributions

1. $p(\tau^2|\boldsymbol{\beta}, \boldsymbol{X}) \sim IG\Big(a+\frac{K-1}{2}, b+\boldsymbol{\beta}'P^*\boldsymbol{\beta}\Big)$.
2. $P(\boldsymbol{\beta}|\tau^2, \boldsymbol{Z}) \propto \prod_{j=1}^{K} c_j^{n_j}\exp\Big\{-\frac{1}{2\tau^2}\boldsymbol{\beta}'P^*\boldsymbol{\beta}\Big\}$.
3. $P(Z_i|X_i, \boldsymbol{\beta}, \mu_j, \sigma^2)$.

4.2.3 Metropolis-Hasting Step

We cannot apply Gibbs sampling to $\boldsymbol{\beta}$ because its conditional distribution is of a nonstandard form. We implement an independence M-H step. A proposal $\boldsymbol{\beta}^p$ is drawn from a multivariate normal distribution $N\Big(\hat{\boldsymbol{\beta}}, \hat{\Sigma}_{\hat{\boldsymbol{\beta}}}\Big)$ where $\hat{\boldsymbol{\beta}}$ is the maximizer of

$$\prod_{j=1}^{K} c_j^{n_j} \exp\Big\{-\frac{1}{2\tau^2}\boldsymbol{\beta}' P^* \boldsymbol{\beta}\Big\} \tag{4.8}$$

and $\hat{\Sigma}_{\hat{\boldsymbol{\beta}}}$ is the negative inverse Hessian of $\log \mathrm{P}(\boldsymbol{\beta}|\tau^2, \boldsymbol{Z})$ evaluated at $\hat{\boldsymbol{\beta}}$. The maximizer is obtained by implementing the Broyden-Fletcher-Goldfard-Shano method. The proposed value $\boldsymbol{\beta}$ is accepted with probability $\min\{1, r\}$ where

$$r = \frac{p(\boldsymbol{\beta}^p|\tau^2, \boldsymbol{Z}) q(\boldsymbol{\beta}^c)}{p(\boldsymbol{\beta}^c|\tau^2, \boldsymbol{Z}) q(\boldsymbol{\beta}^p)}. \tag{4.9}$$

In (4.9) the superscripts c and p denote current and proposed values, and

$$q(\boldsymbol{\beta}) \propto \exp\Big\{-\frac{1}{2}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})' \hat{\Sigma}_{\hat{\boldsymbol{\beta}}}^{-1} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\Big\}.$$

To maximize (4.8),we compute the gradient and Hessian of $\log P(\boldsymbol{\beta}|\tau^2, \boldsymbol{Z})$ given by

$$\boldsymbol{g} = \begin{pmatrix} n_2 - n \cdot c_2 \\ n_3 - n \cdot c_3 \\ \cdots \\ \cdots \\ \cdots \\ n_J - n \cdot c_J \end{pmatrix} - \frac{1}{\tau^2} P^* \beta \tag{4.10}$$

and

$$H = n \cdot \begin{pmatrix} c_2(c_2-1) & c_2 c_3 & .. & .. & .. & c_2 c_J \\ c_3 c_2 & c_3(c_3-1) & .. & .. & .. & c_3 c_J \\ .. & .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. & .. \\ c_J c_1 & c_J c_2 & c_J c_3 & .. & .. & c_J(c_J-1) \end{pmatrix} - \frac{1}{\tau^2} P^*. \tag{4.11}$$

Chapter 5

Simulation Study

In this chapter, we present a comparison of the three density estimation methods. The first method is the Dirichlet Process Prior, the second method is the Regression and the third method is the Mixture of Normals with Known Components. We simulate data from two different distributions. We also measure the discrepancy between the estimated density and the true density using the Kullback-Leibler Divergence measure.

Definition: Kullback-Leibler Divergence

To measure the quality of a density estimator, the Kullback-Leiber divergence is used. It measures the discrepancy between the density estimator $\hat{f}$ and the true density f

$$D_{KL}(f||\hat{f}) = -\int \log\Big(\frac{f(X)}{\hat{f}(X)} f(X)\Big).$$

5.1 The True Distributions

We have simulated from two different distributions: A mixture of normals with three components and a chi-squared on 4 degrees of freedom. The distributions and the chosen parameters are as follows

$$f_1(X) = \frac{0.35}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2(1^2)}(X+6)^2\Big) + \frac{0.4}{\sqrt{2\pi(0.75)}} \exp\Big(-\frac{1}{2(.75^2)}(X)^2\Big) + \frac{0.25}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2(1^2)}(X-6)^2\Big)$$

$$f_2(X) = \frac{1}{(2^2)\Gamma(2)} X \exp\Big(-\frac{X}{2}\Big).$$

5.2 Simulations

5.2.1 Simulations from f_1

We generate 75 samples from f_1, where each sample is of size 500 observations. The MCMC sampling scheme is implemented for each sample for 10^4 iterations with a warmup of 1500.

Method 1

Figure 5.1, shows a true density of f_1. The lines are 75 densities estimates based on Method 1. It is seen that the method provides good fits with small variation.

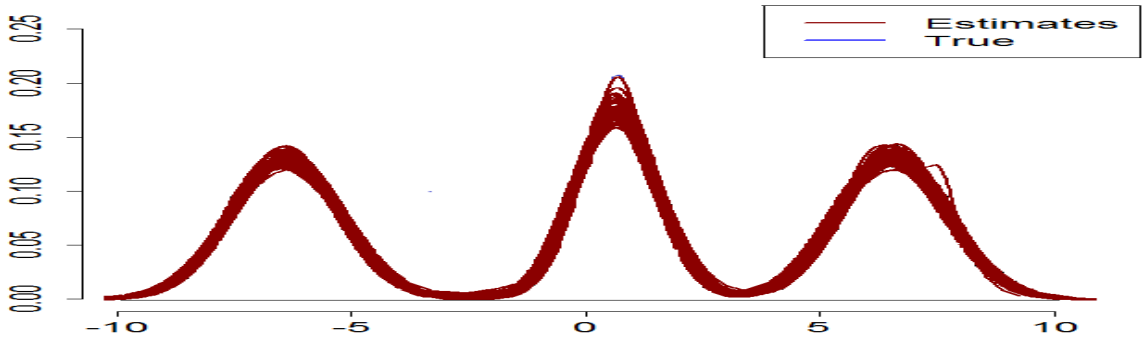

Figure 5.1: The true f_1 along with estimates of f_1 based on Method 1.

Method 2

Figure 5.2 is analoguos to Figure 5.1 but is based on Method 2. The fits are still good.

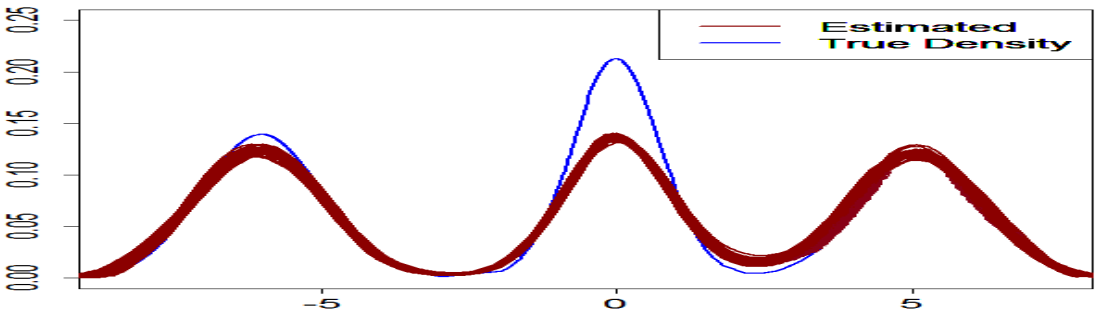

Figure 5.2: The true f_1 along with estimates of f_1 based on Method 2.

Method 3

Figure 5.3 shows f_1 along with density estimates based on a single sample. The estimates are based on Method 3 with different values of K. A value of K larger than 35 does not improve the fit. For this reason, in Figure 5.4, each of the 75 estimates is based on $K = 35$. The fits are still good but with larger variation compared to the other Methods.

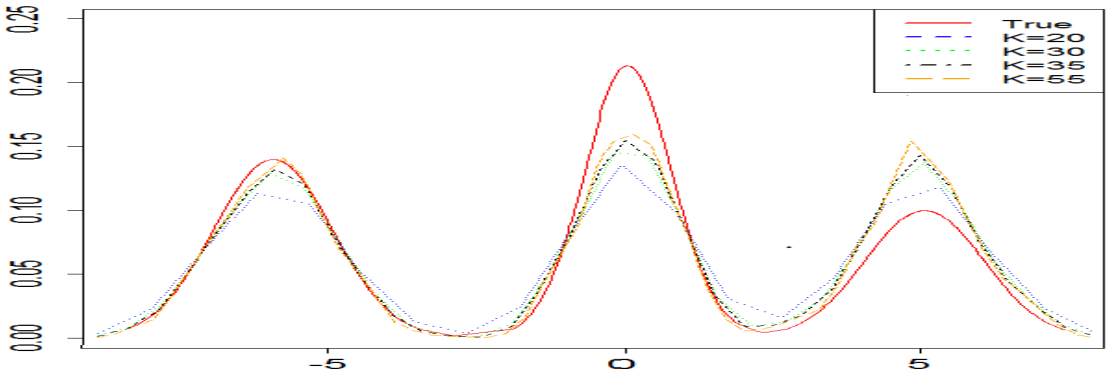

Figure 5.3: The true density f_1 along with estimates of f_1 along with estimates corresponding to different values of K based on a single random sample.

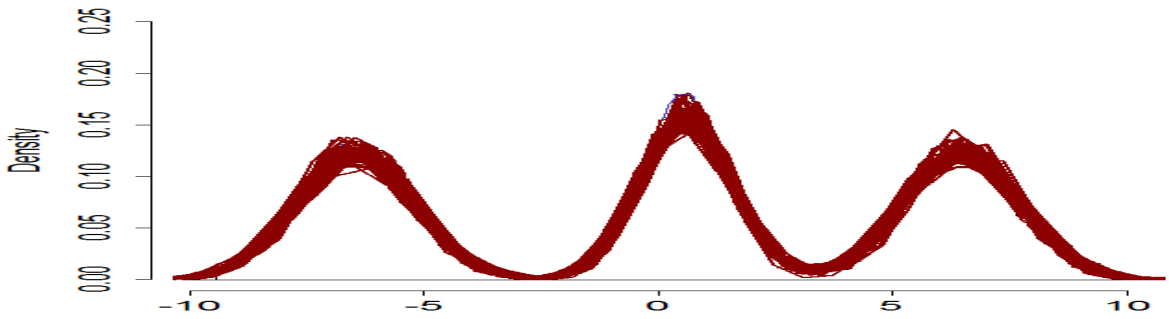

Figure 5.4: The true density f_1 along with estimates of f_1 based on on Method 3.

5.2.2 Simulations from f_2

We generate 75 samples from f_2, where each sample is of size 500 observations. The MCMC sampling scheme is implemented for each sample for 10^4 iterations with a warmup of 1500.

Method 1

Figure 5.5 shows a true density of f_2. The lines are 75 densities estimates based on Method 1. It is seen that the Method 1 provides good fits.

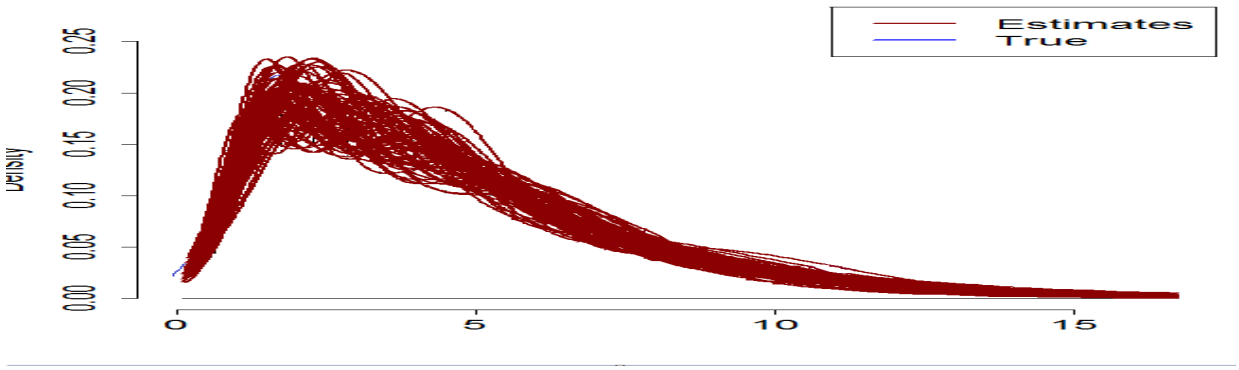

Figure 5.5: The true f_2 along with estimates of f_2 based on Method 1.

Method 2

Figure 5.6 is analogous to 5.5 but is based on Method 2. The fits are still good.

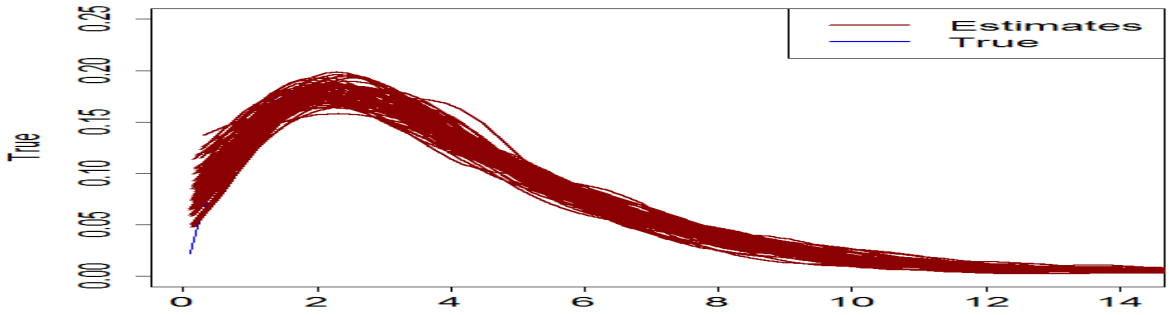

Figure 5.6: The true f_2 along with estimates of f_2 based on Method 2.

Method 3

Figure 5.7 shows a single data set simulated from f_2. The estimates are based on Method 3 with different values of K. A value of K larger than 35 does not improve the fit. For this reason, in Figure 5.8, each of the 75 estimates is based on $K = 35$. The fits are still good but with large variation.

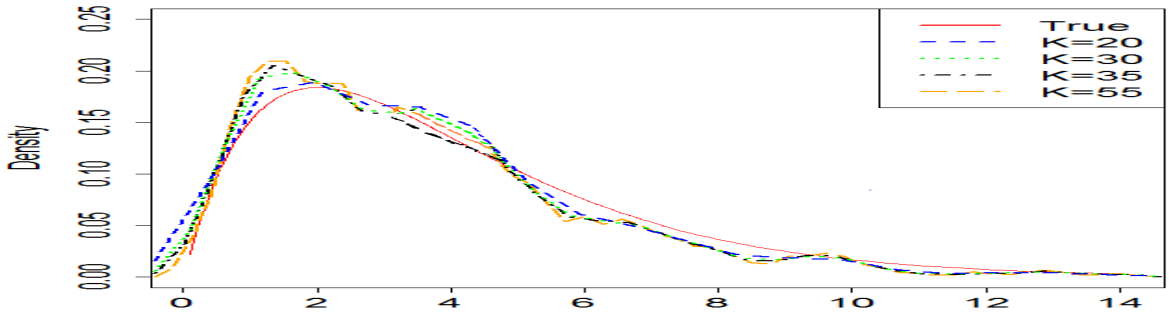

Figure 5.7: The true f_2 along with estimates corresponding to different values of K based on a single random sample.

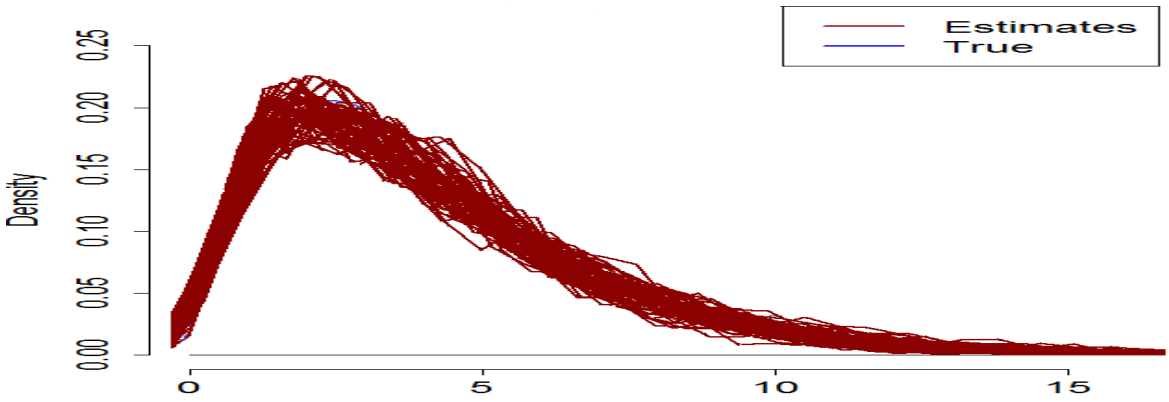

Figure 5.8: The true density f_2 along with estimates based on Method 3.

5.3 Comparison of the estimation methods

5.3.1 Kullback-Leibler Divergence For The Simulations From f_1

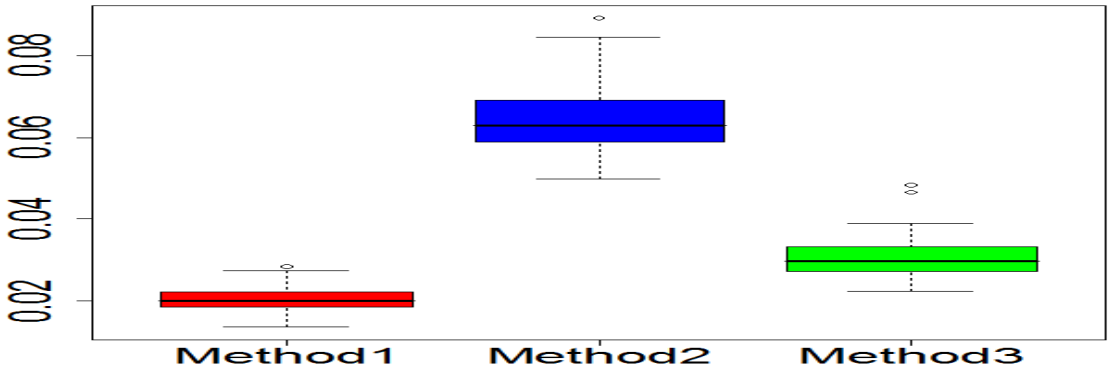

Figure 5.9: Boxplots of KLD using the three methods, where samples are generated from f_1

Figure 5.9 shows for each of the three estimation methods boxplots of the KLD values computed for each sample. Method 1 achieves the samllest KLD values, while Method 2 has the highest KLD values, for this simulation setting.

5.3.2 Kullback-Leibler Divergence For The Simulations From f_2

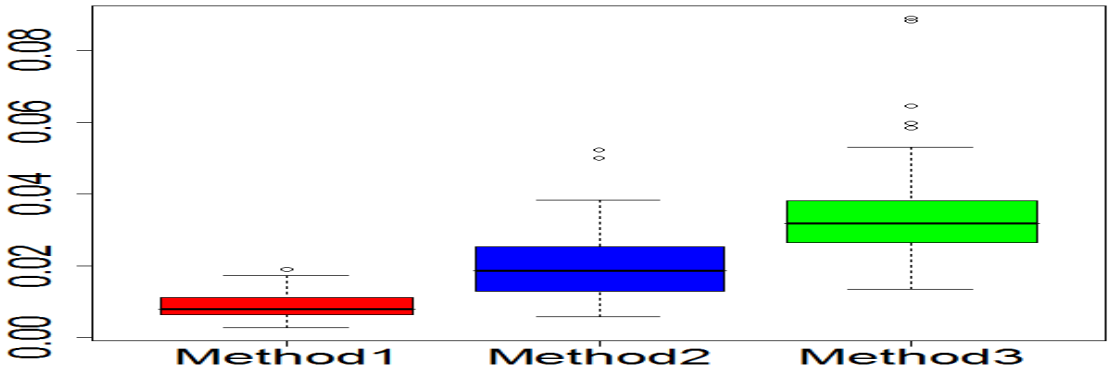

Figure 5.10: Boxplots of KLD using the three methods, where samples are generated from f_2.

Figure 5.10 is analogous to 5.9 but is based on the simulations from f_2. In this case, Method 1 is still superior to the other methods, and Method 3 performs the worst.

5.4 Concluding remarks

We have evaluated three nonparametric methods for density estimation. The results from our simulations show that all three estimators display good performance. The first method outperforms the other two methods. The advantage of Method 2 is that it takes less time to fit compared to the other methods.

Methods 1 and 3 are based on mixtures and as such seem to perform better whenever the data come from a mixture. In addition, it is easy to estimate a cdf using these two methods by evaluating the mixtures of the cdfs rather than that of the pdfs. This is not straight forward with Method 2.

Chapter 6

Data Analysis

In this chapter, we apply our estimation methods to the Hidalgo data set. The data consist of 485 stamp thickness measurements in millimeters. We consider the fitting of the three nonparametric models to the data. A detailed description of these data is given in Izenman and Sommer (1988). Figure 6.1 displays a histogram of the data. The thickness of 485 unwatermarked used white wove stamps, of which 289 had an 1872 overprint and 196 had either a 1873 or a 1874 overprint (Basford et al., 1997).

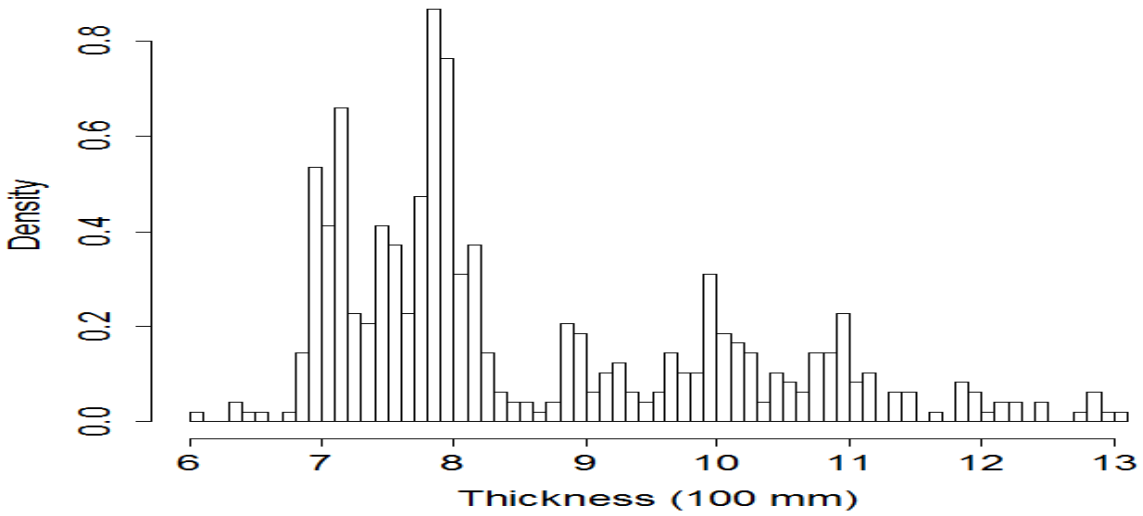

Figure 6.1: Histogram of the 485 measurements of the Hidalgo Issue from Mexico.

From Figute 6.1, we can see that there are seven clusters around the following values 0.07mm, 0.08mm, 0.09mm, 0.10mm, 0.11mm, 0.12mm and 0.13mm. Also, it is noted that the data have two major peaks, while the other peaks are smaller.

6.1 Method 1 Fits

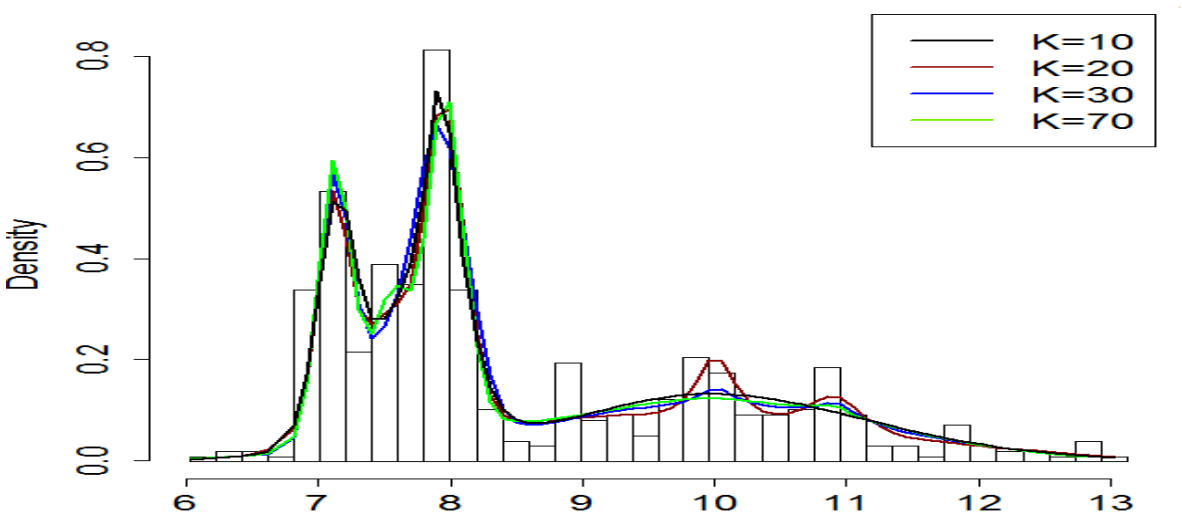

Figure 6.2: A histogram of the data along with four fitted densities based on Method 1, corresponding to different values of K.

Figure 6.2 displays a histogram of the Hidalgo data along with fitted densities using the DPP method. We use four different values of K= 10, 20, 30, 70. It can be seen that the method is not very sensitive to the value of K, as long as K is large enough.

6.2 Method 2 Fits

Figure 6.3 displays a histogram of the Hidalgo data along with fitted densities using the Regression method. We use five different values of K= 25, 50, 100, 150, 250. Again the fits are not very sensitive to the choice of K but the method seems to oversmooth. We can conclude that this method is not a good fit for data containing many modes that are too close to each other.

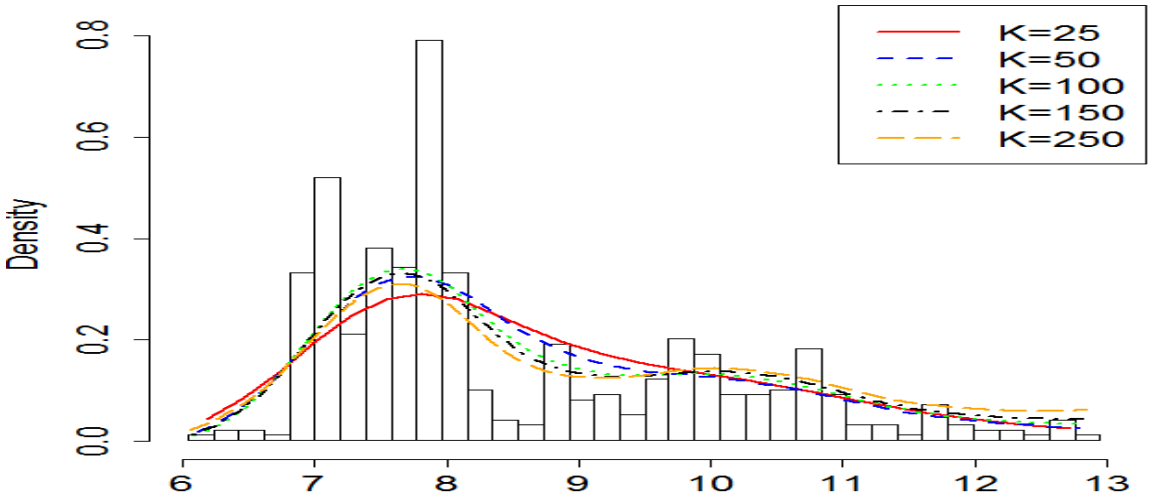

Figure 6.3: A histogram of the data along with five fitted densities based on Method 2 corresponding to different values of K.

6.3 Method 3 Fits

Figure 6.4 displays a histogram of the Hidalgo data along with fitted densities using the third method. We use five different values of K= 20, 30, 40, 50, 60. It can be seen that the method is sensitive to the value of K. A value of $K = 40$ seems to provide a good fit.

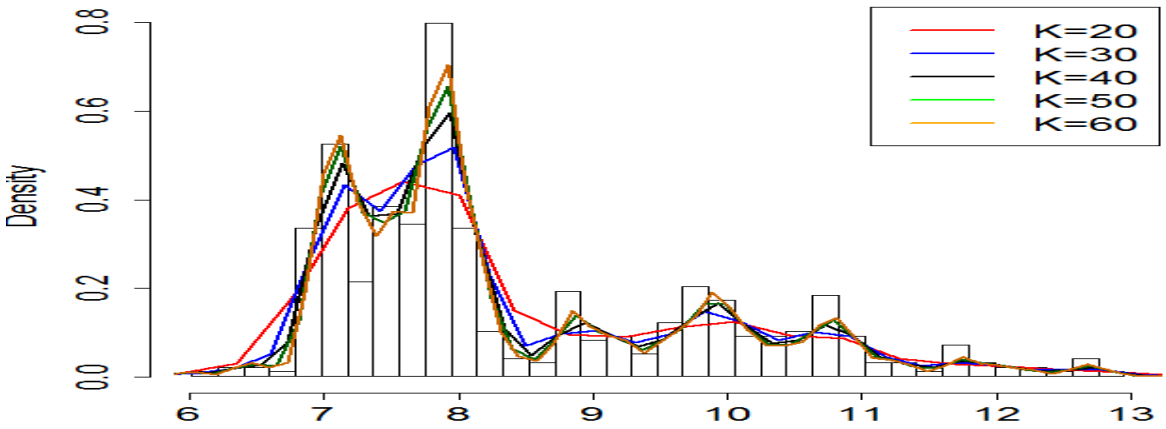

Figure 6.4: A histogram of the data along with five fitted densities based on Method 3 corresponding to different values of K.

6.4 Conclusion

In this thesis, we have evaluated three nonparametric methods for density estimation. The Dirichlet process prior turns out to perform well in all the cases we have examined. Method 3 takes the longest to fit because of the optimization needed to implement the Metropolis-Hasting steps. Method 2 tends to perform poorly for data with multiple modes that are close to each other.

Appendix A

Proofs

A.1 Proofs of Equations Presented in Chapter 2

Equation (2.15)

Proof: $P(\mu_j|\tau_j^2, \boldsymbol{X}, \boldsymbol{z}) \propto \prod_{\{i:z_i=j\}}^{n} \frac{1}{\sqrt{2\pi\tau_{z_i}^2}} \exp\Big(-\frac{1}{2\tau_{z_i}^2}(X_i - \mu_{z_i})^2\Big) \times P(\mu_j)$, where
$\mu_j \sim N(\theta, \sigma_\mu^2)$ and $X \sim N(\mu_j, \frac{\tau_{z_i}^2}{n_j})$
$P(\mu_j|\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{\tau}, \theta) \propto N(\mu_j^*, \sigma^*)$, where

$$\sigma^* = \frac{1}{\frac{n_j}{\tau_j^2} + \frac{1}{\sigma_\mu}} = \Big(\frac{n_j}{\tau_j^2} + \frac{1}{\sigma_\mu}\Big)^{-1}$$

and

$$\mu_j^* = \frac{\frac{n_j}{\tau_j^2}\frac{\sum_{\{i:z_i=j\}} X_i}{n_j} + \frac{\theta}{\sigma_u}}{\frac{n_j}{\tau_j^2} + \frac{1}{\sigma_u}}$$

$$= \frac{\frac{\sum_{\{i:z_i=j\}} X_i}{\tau_j^2} + \frac{\theta}{\sigma_u}}{\frac{n_j}{\tau_j^2} + \frac{1}{\sigma_u}} = \sigma^* \frac{\sum_{\{i:z_i=j\}} X_i}{\tau_j^2} + \frac{\theta}{\sigma_u}$$

■

Equation (2.16)

Proof: Given $\tau_j^2 \sim \text{IG}(\nu_1, \nu_2)$

$$
\begin{aligned}
P(\tau_j^2|\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{\mu}, \nu_1, \nu_2) &\propto \prod_{i:z_i=j} \frac{1}{\sqrt{2\pi\tau_j^2}} \exp\Big\{ -\frac{1}{2\tau_j^2}(X_i - \mu_j)^2 \Big\} . \frac{\Gamma(\nu_2)^{\nu_1}}{\Gamma(\nu_1)} . (\tau_j^2)^{-(0.5\nu_1 - 1)} \exp\Big(-\frac{\nu_2}{2\tau_j^2}\Big) \\
&\propto (\tau_j^2)^{\frac{n_j}{2}} \exp\Big(-\frac{1}{2\tau_j^2} \sum_{i:z_i=j} (X_i - \mu_j)^2\Big) (\tau_j^2)^{-\frac{\nu_1}{2}-1} \exp\Big(\frac{\nu_2}{2\tau_j^2}\Big) \\
&\propto (\tau_j^2)^{\frac{-(n_j+1)}{2}-1} \exp\Big(-\frac{1}{2\tau_j^2} \sum_{i:z_i=j} (X_i - \mu_j)^2 + \nu_2\Big) \\
&\sim \text{IG}\Big(\frac{n_j+1}{2}, \frac{\sum_{i:z_i=j} (X_i - \mu_j)^2 + \nu_2}{2}\Big)
\end{aligned}
$$

■

Equation (2.19)

Proof: $P(\theta|\pi) \propto \prod_{j=1}^{K} \exp\big(-\frac{1}{2\sigma_\mu^2}(\mu_j - \theta)^2\big) \exp\Big\{-\frac{\theta}{A}\Big\}$

$\theta \sim N(\mu_j, \sigma_\mu^2)$, prior mean =0 and prior variance=$\frac{1}{A}$

variance= $\dfrac{1}{\frac{1}{\sigma_\mu^2} + \frac{1}{A}} = \Big(\frac{1}{\sigma_\mu^2} + \frac{1}{A}\Big)^{-1} = \sigma^*$

mean =$\dfrac{\frac{\sum_{j=1}^{K} \mu_j}{\sigma_\mu^2} + 0}{\frac{1}{\sigma_\mu^2} + \frac{1}{A}} = \sigma^* \sum_{j=1}^{K} \mu_j = \theta^*.$

Therefore, $\theta \sim N(\theta^*, \sigma^*)$ ■

Equation (2.17)

Proof: The prior on α is a gamma distribution with shape n_1 and rate n_2

$$p(\alpha) = \frac{n_2^{n_1}}{\Gamma(n_1)}\alpha^{n_1-1}\exp(-n_2\alpha)$$

by definition,$\beta_j^* \sim \text{Beta}(1,\alpha)$ for $j = 1, ..., K-1$ and $\beta_K^* = 1$

$$\begin{aligned}
P(\alpha|\boldsymbol{\pi}) &= \frac{n_2^{n_1}}{\Gamma(n_1)}\alpha^{n_1-1}\exp(-n_2\alpha)\prod_{j=1}^{K-1}\frac{\Gamma(1+\alpha)}{\Gamma(\alpha)}(\beta_j^*)^{1-1}(1-\beta_j^*)^{\alpha-1} \\
&\propto \alpha^{n_1-1}\exp(-n_2\alpha)\prod_{j=1}^{K-1}\alpha(1-\beta_j^*)^{\alpha-1} \\
&= \alpha^{n_1-1}\exp(-n_2\alpha)\alpha^{K-1}\prod_{j=1}^{K-1}(1-\beta_j^*)^{\alpha-1} \\
&= \alpha^{(n_1-K-1)-1}\exp(-n_2\alpha)\prod_{j=1}^{K-1}(1-\beta_j^*)^{\alpha-1} \\
&= \alpha^{(n_1-K-1)-1}\exp(-n_2\alpha)\exp\Big(\log\Big\{\prod_{j=1}^{K-1}(1-\beta_j^*)^{\alpha-1}\Big\}\Big) \\
&= \alpha^{(n_1-K-1)-1}\exp(-n_2\alpha)\exp\Big(\sum_{j=1}^{K-1}\log\Big\{(1-\beta_j^*)^{\alpha-1}\Big\}\Big) \\
&\propto \alpha^{(n_1-K-1)-1}\exp(-n_2\alpha)\exp\alpha\Big(\sum_{j=1}^{K-1}\log\Big\{(1-\beta_j^*)\Big\}\Big) \\
&= \alpha^{(n_1-K-1)-1}\exp\alpha\Big(n_2 - \sum_{j=1}^{K-1}\log\Big\{(1-\beta_j^*)\Big\}\Big) \\
&\propto \text{Gamma}(n_1+K-1, n_2 - \sum_{j-1}^{K-1}\log\Big\{1-\beta_j^*\Big\})
\end{aligned}$$

■

A.2 Proofs of Equations Presented in Chapter 3

Equations (3.6), (3.5), and (1)

Claim: Let $Y = X\beta + Zu + \epsilon = X^*\beta^* + \epsilon$ where $X^* = \Big(X|Z\Big)$ and $\beta^* = \begin{bmatrix}\beta \\ u\end{bmatrix}$.

The conditional distributions of σ^2, τ^2, and $(\beta^{'}, u^{'})$ are

1. $\sigma^2 \sim IG\Big(\frac{n}{2} - 1,\, \frac{1}{2}(Y - X^*\beta^*)^{'}(Y - X^*\beta^*\Big) \cdot I(0 \leq \sigma^2 \leq C_{\sigma^2})$
2. $IG(\frac{m}{2} - 1,\, \frac{1}{2}u^{'}u)I(0 \leq \tau^2 \leq C_{\tau^2})$
3. $N\Big(\frac{1}{\sigma^2}TX^{*'}Y,\, T\Big)$

where $T{=}\sigma^2(X^{*'}X^* + \sigma^2 A^{-1})^{-1}$.

Proof: The posterior likelihood is

$$\begin{aligned} p(u, \beta, \sigma^2, \tau^2|Y) \propto & \Big(\frac{1}{\sigma^2}\Big)^{n/2} e^{\Big(\frac{-1}{2\sigma^2}(Y^{*'}(Y^*)\Big)} \\ & \times \Big(\frac{1}{\tau^2}\Big)^{m/2} e^{\Big(\frac{-1}{2\tau^2}(u^{'}u)\Big)} \cdot I_{[0, C_{\tau^2}]} \\ & \times (\tau^2)\Big(\frac{1}{\sigma^2_\beta}\Big)^{n/2} e^{\Big(\frac{-1}{2\sigma^2_\beta}(\beta^{'}\beta)\Big)} \cdot I_{[0, C_{\sigma^2}]}(\sigma^2) \end{aligned} \tag{A.1}$$

where $Y^* = Y - X\beta - Zu$

■

Equation (3.6)

Proof: From (A.1)and by keeping only terms that depend on τ^2, the conditional distribution of τ^2 is a truncated inverse gamma

$$\tau^2|u \sim IG(\frac{m}{2} - 1, \frac{1}{2}u^{'}u) \times I(0 \leq \tau^2 \leq C_{\tau^2}) \tag{A.2}$$

Let V$\sim$ $IG(\frac{m}{2} - 1, \frac{1}{2}u^{'}u)$, the distribution of τ^2 is found by using Cumulative Function Distribution (CDF).

By writing the CDF of the equation (A.2) we have

$$F_{\tau^2}(\tau^2) = \frac{F_V(\tau^2)}{F_V(C_{\tau^2})} \tag{A.3}$$

where

$$\begin{aligned} F_V(\tau^2) &= 1 - F_V(\frac{1}{\tau^2}) \\ F_V(C_{\tau^2}) &= 1 - F_V(\frac{1}{C_{\tau^2}}). \end{aligned} \tag{A.4}$$

By replacing (A.4) in equation (A.3), the CDF of τ^2 becomes

$$F_{\tau^2}(\tau^2) = \frac{1 - F_V(1/\tau^2)}{1 - F_V(1/C_{\tau^2})} = u,$$

where $u \sim$ uniform$[0, 1]$

$$\frac{1}{\tau^2} = F_V^{-1}(1 - u + u \cdot F_V(\frac{1}{C_{\tau^2}}) \tag{A.5}$$

Finaly , from the equation (A.5), τ^2 can be drawn from

$$\tau^2 = \frac{1}{F_V^{-1}}(1 - u + u \cdot F_V(\frac{1}{C_{\tau^2}}))$$

■

Equation (3.5)

Proof: From (3.3), and by keeping only terms that depond on σ^2, the conditional distribution of σ^2 is a truncated inverse gamma

$$\sigma^2 \sim IG(\frac{n}{2} - 1, \frac{1}{2}(Y - X^*\beta^*)'(Y - X^*\beta^*) \cdot I(0 \leq \sigma^2 \leq C_{\sigma^2})$$

Using the same techinques used to prove the equation (3.5), we show that

$$\sigma^2 = \frac{1}{F_V^{-1}}(1 - u + u \cdot F_V(\frac{1}{C_{\sigma^2}}))$$

■

Equation (1)

Proof: Let $X^* = \Big(X|Z\Big)$ and $\beta^* = \begin{bmatrix} \beta \\ u \end{bmatrix}$

sample mean $= (X^*X^*)^{-1}X^{*'}$

prior mean $= 0$

sample variance $= \sigma^2(X^{*'}X^*)^{-1}$

prior variance $= A_{(m+2\times m+2)} = diag(\sigma^2_\beta, \sigma^2_\beta, \tau^2, \tau^2, ..., \tau^2)$

posterior variance $= (\frac{X^{*'}X^*}{\sigma^2} + A^{-1}) = \sigma^2(X^{*'}X^* + \sigma^2 A^{-1})^{-1}$

posterior mean $= \frac{1}{\sigma^2}(A^{-1} + \frac{X^{*'}X}{\sigma^2})^{-1}X^*Y = \dfrac{1}{\sigma^2}(\text{post variance})X^{*'}Y$

$(\boldsymbol{\beta'}, \boldsymbol{u'}) \sim N\Big(\dfrac{1}{\sigma^2}(\text{post variance})X^{*'}Y, \text{post variance}\Big)$ ■

A.3 Proofs of Equations Presented in Chapter 4

Equation(4.10)

Proof: $c_1 = \dfrac{1}{1 + \sum_{k=2}^{J} \exp(\beta_k)}$ because β_1 is set to equal a zero for identifiability.

The logarithm likelihood function

$$\sum_{j=1}^{J} n_j log(c_j) - \frac{1}{2\tau^2}\boldsymbol{\beta}' P^* \boldsymbol{\beta}.$$

By taking the derivative of $\dfrac{1}{2\tau^2}\boldsymbol{\beta}' P^* \boldsymbol{\beta}$ with respect to $\boldsymbol{\beta}$

$$\frac{\partial}{\partial \boldsymbol{\beta}}\Big(\frac{1}{2\tau^2}\boldsymbol{\beta}' P^* \boldsymbol{\beta}\Big) = \frac{1}{\tau^2} P^* \boldsymbol{\beta} \tag{A.6}$$

Now, let find the derivative of $\sum_{j=1}^{J} n_j log(c_j)$ with respect to $\boldsymbol{\beta}$. To do so, first we need to derive the derivatives of $log(c_j)$ with repect to β_h where j and h take values from 1 to K. For $j = 1$, the derivative with respect to β_h

$$\frac{\partial log(c_1)}{\partial \beta_h} = \frac{-e^{\beta_h}}{1 + \sum_{k=2}^{J} e^{\beta_k}} = -c_h, \text{ for } h = 1, 2, ..., J \tag{A.7}$$

For $j = 2, 3, ..., J$, the derivative with respect to β_h

$$\frac{\partial log(c_j)}{\partial \beta_h} = 1 - \frac{e^{\beta_j}}{1 + \sum_{k=2}^{J} e^{\beta_k}} = 1 - c_j, \text{ when } h = j \tag{A.8}$$

$$\frac{\partial log(c_j)}{\partial \beta_h} = -\frac{e^{\beta_h}}{1 + \sum_{k=2}^{J} e^{\beta_k}} = -c_h, \text{ when } h = j. \tag{A.9}$$

From (A.8) and (A.9), the derivative of $log(c_j)$ with respect to β_h is

$$\frac{\partial log(c_j)}{\partial \beta_h} = I_{j,h} - c_h, \tag{A.10}$$

where $I_{j,h} = \begin{cases} 1, \text{ if } j = h \\ 0, \text{ if } j \# h \end{cases}$

Using (A.6), (A.7), and (A.10), the gradient of logarithm likelihood of equation(**??**) is

$$\nabla f(\boldsymbol{\beta}) = \begin{pmatrix} -n \cdot c_1 \\ n_2 - n \cdot c_2 \\ n_3 - n \cdot c_3 \\ ... \\ ... \\ ... \\ n_{J-1} - n \cdot c_{J-1} \end{pmatrix} - \frac{1}{\tau^2} P^* \beta. \tag{A.11}$$

■

Equation (4.11)

Proof: Note that

$$\frac{\partial^2 log(c_j)}{\partial\beta_j\partial\beta_h} = \frac{\frac{\partial log(c_j)}{\partial\beta_h}}{\partial\beta_j} = \frac{\partial}{\partial\beta_j}(-c_h) \tag{A.12}$$

and

$$\frac{\partial}{\partial\beta_j} log(c_h) = -\frac{\frac{\partial c_h}{\partial\beta_j}}{c_h} \tag{A.13}$$

from (A.13) and (A.10)

$$\frac{\partial c_h}{\partial\beta_j} = -c_h(I_{j,h} - c_j). \tag{A.14}$$

The second derivative of $\frac{1}{2\tau^2}\boldsymbol{\beta}' P^* \boldsymbol{\beta}$ is

$$\frac{\partial^2}{\partial\boldsymbol{\beta}'\partial\boldsymbol{\beta}}\Big(\frac{1}{2\tau^2}\boldsymbol{\beta}' P^* \boldsymbol{\beta}\Big) = \frac{1}{\tau^2}P^* \tag{A.15}$$

Finally, using (A.14) and (A.15),the Hessian matrix is

$$Hf(\boldsymbol{\beta}) = n \cdot \begin{pmatrix} c_1^2 & c_1c_2 & c_1c_2 & .. & .. & .. & c_1c_J \\ c_2c_1 & c_2(c_2-1) & c_2c_3 & .. & .. & .. & c_2c_J \\ c_3c_1 & c_3c_2 & c_3(c_3-1) & .. & .. & .. & c_3c_J \\ .. & .. & .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. & .. & .. \\ c_{J-1}c_1 & c_{J-1}c_2 & c_{J-1}c_3 & .. & .. & .. & c_{J-1}(c_{J-1}-1) \end{pmatrix} - \frac{1}{\tau^2}P^*.$$

■

Appendix B

R-Code

B.1 R-code for Dirichlet Process Prior method

```
#############
## Library ##
#############
library(mixtools)
library(Hmisc)
library(BSDA)
library(flexmix)
##################
## DPP-function ##
##################
Dpp<-function(x)
{
####################
## number of Loop ##
#       &         ##
##    Warmup      ##
####################
nloop<-10^4
nwarmup<-1500
########################
```

```
## Number of mixtures ##
########################
nclusters<-30
n<-length(x)
##########
# Priors #
##########
alpha<-rep(1,nloop)
nu1<-nu2<-2
A<-1000
variance<-(4*sd(x))^2
sig_star<-1/( (nclusters/variance)+(1/A))
#####################
# Storing matrices ##
#####################
muu<-rep(0,nclusters)
tau<-rep(1,nclusters)
Y<-array(0,dim=c(nloop,n,nclusters))
V<-matrix(0,nloop,nclusters)
C<-matrix(0,nloop,n)
pi<-size<-rep(0,nclusters)
v<-rep(1/nclusters,nclusters)
v[nclusters]<-1
mu<-matrix(0,nloop,nclusters)
tausq<-matrix(1,nloop,nclusters)
p<-K<-matrix(0,n,nclusters)
cumprod<-one_v<-log_one_v<-matrix(0,nloop,nclusters)
sum_mu<-theta<-theta_star<-sum_log_v<-rep(0,nloop)
```

```
tausq_prior_a<-tausq_prior_b<-0.01
mean_norm<-sig_norm<-matrix(0,nloop,nclusters)
#########
# Loop  #
#########
for (i in 1:nloop)
            {
####################
## Stick Breaking ##
####################
cumv<-cumprod(1-v)
pi[1]<-v[1]
for (j in 2:nclusters)  pi[j]<-v[j]*cumv[j-1]
################################
## Conditional K \\Indicator  ##
################################
for (j in 1:nclusters) K[,j]<-pi[j]*dnorm(x,muu[j],sqrt(tau[j]))
p<-K/apply(K,1,sum)
#####################################################################
## Generation of Indicator from Multinomial with probability p ##
#####################################################################
C[i,]<-ind<-rMultinom(p,1)
for (j in 1:nclusters)      size[j]<-length(ind[ind==j])
for (j in 1:(nclusters-1))   v[j]<-rbeta(1,1+size[j],alpha[i]+
                                    sum(size[(j+1):nclusters]))
V[i,]<-v
##########
## 1- V ##
```

```
##########
for(j in 1:nclusters)       one_v[i,j]<-1-V[i,j]
########################
## log of 1-V         ##
########################
for(j in 1:nclusters)       log_one_v[i,j]<-log(one_v[i,j])
cumprod[i,]<-cumprod(one_v[i,])
############################
##  sum of Log (1- betas)  ##
##       For Alpha         ##
############################
sum_log_v[i]<-sum(log_one_v[i,1:nclusters-1])
############
## Alpha  ##
############
alpha[i]<-rgamma(1,shape=nclusters+nu1-1,rate=nu2-sum_log_v[i])
#################################
## SUM of mu to be used in theta ##
##         Drawing Theta         ##
#################################
for(j in 1:nclusters)
                    {
                     sum_mu[i]<-sum(mu[i,1:j])
   theta_star[i]<- (sig_star/variance)*sum_mu[i]
   theta[i]<-rnorm(1,mean=theta_star[i], sd= sqrt(sig_star))
  }
###########
## TAUSQ ##
```

```
############
for (j in 1:nclusters) tausq[i,j]<-tau[j]<-1/rgamma(1,tausq_prior_a+size[j]/2,
                        tausq_prior_b+sum((x[ind==j]-muu[j])^2)/2)
########
## MU ##
########
for (j in 1:nclusters)
                    {
                    sigg<-1/((size[j]/tau[j]) +(1/variance))
        me<-sigg*(sum(x[ind==j]/tau[j]) + theta[i]/variance)
                    mu[i,j]<-muu[j]<-rnorm(1,me, sd=sqrt(sigg))
                    }
for(j in 1:nclusters)  Y[i,,j]<-pi[j]*dnorm(x,mu[i,j],sqrt(tau[j]))

}
#########################################################
Ytmp<-matrix(0,nloop,n)
for (i in 1:nloop)    for (j in 1:n) Ytmp[i,j]<-sum(Y[i,j,])
yhat<-apply(Ytmp[nwarmup:nloop,],2,mean)
list(yhat=yhat)
  }
########################
## Number of samples ##
########################
s<-75
##########
## Data ##
##########
```

```
set.seed(2014)
n<-500
pii<-c(.35,.4,.25)
mu1<-c(-6,0,5)
sig<-c(1,.75,1)
x<-matrix(0,s,3*n)
for(i in 1:s) x[i,]<-rnormmix(n,pi,mu1,sig)
###############
## Plotting ##
###############
hist(x[1,],breaks=50,freq=F,ylim=c(0,.25))
D_T<-rep(0,s)
#################
## Loop to run ##
##   Dpp func  ##
##      &      ##
##     MSE     ##
##      &      ##
##     KLD     ##
#################
for(i in 1:s)
            {
output<-Dpp(x[i,])
g<-pii[1]*dnorm(x[i,],mu1[1],sig[1])
   +pii[2]*dnorm(x[i,],mu1[2],sig[2])
   +pii[3]*dnorm(x[i,],mu1[3],sig[3])
y<-cbind(g,output$yhat)
Q<-KLdiv(y)
```

```
D_T[i]<-Q[2,1]
################
# Plot Density #
################
xxx<-x[i,]
ord<-order(xxx)
lines(xxx[ord],output$yhat[ord],
      type="l",lwd=2,col="darkred")
legend("topright",col=c("darkred","blue"),
       lty=c(1,2,3),legend=c("Estimated","True Density")
       ,lwd=2,cex=.9)
print(i)
}
windows()
boxplot(D_T,boxwex = 0.6,col = "red",
        main = "Estimate--True",xlab = "KDL_Dpp")
legend("topright",legend=c("Max",round(max(D_T),7)
      ,"Min",round(min(D_T),7)))
```

B.2 R-code for Regression Method

```
#############
## Library ##
#############
```

```
library(mixtools)
library(sm)
library(mnormt)
library(Bolstad)
library(BSDA)
library(flexmix)
#########################
## Regression_function ##
#########################
REG<-function(xx,n)
{
                  k<-round(n^(3/4))
#####################################################
## Converting Density Estimation Into Regression ##
#####################################################
convert<-function(x) {
   grouping<-binning(x,nbins=k)
    tjj <-grouping$breaks
    Nj<-grouping$table.freq
    yj<-sqrt(k*(max(x)-min(x))/n)*sqrt(Nj+0.25)
list(x=tjj,y=yj)
                      }
data<-convert(xx)
tj<-rep(NA,k)
for (i in 1:k)
 {
  tj[i]<-(data$x[i]+data$x[i+1])/2
 }
```

```
Yj<-data$y
tj
Yj
##############################################################################
## Applying Cubic Smoothing Spline method to (tj;Yj)to get an estimate of rn ##
##############################################################################
nobs<-length(tj)
nsim<-10^4
nwarmup<-1500
hyp_tausq<-10^5
hyp_sigm<-10^3
sig_b<-10^4
m<-20
omega<-matrix(0,nobs,nobs)
for(i in 1:nobs){
    for(j in 1:nobs){
                  if(tj[i]<=tj[j]){
                  omega[i,j]<-0.5*(tj[i]^2)*(tj[j] -(tj[i]/3))
                        }
                  else{
                  omega[i,j]<-0.5*(tj[j]^2)*(tj[i] -(tj[j]/3))
                     }
                   }
              }
T<-eigen(omega)
Q<-T$vectors[1:nobs,1:m]
DD<-sqrt(T$values[1:m])
D<-diag(DD)
```

```
Z<-Q%*%D
X<-matrix(c(rep(1,nobs),tj),nrow=nobs,ncol=2)
tausq<-rep(1,nsim+1)
sigma<-rep(1,nsim+1)
X_Z<-cbind(X,Z)
combined<-matrix(1,nsim+1,2+m)
for(i in 1:nsim)
            {
unif<-runif(1)
f<-c(1/sig_b,1/sig_b,rep(tausq[i+1],m))
A<-diag(f)
var<-sigma[i+1]*solve(t(X_Z)%*%X_Z + sigma[i+1]*A)
mean_com<-(1/sigma[i+1])*var%*%t(X_Z)%*%Yj
combined[i+1,]<-rmnorm(1,mean_com,var)
tausq[i+1]<-1/(qgamma(1-unif+ unif*pgamma((1/hyp_tausq),
            (m/2)-1,0.5*t(combined[i+1,3:m])%*%combined[i+1,3:m]),
            (m/2)-1,0.5*t(combined[i+1,3:m])%*%combined[i+1,3:m]))
sigma[i+1]<-(sqrt(k*(max(xx)-min(xx))/(4*n)))/(qgamma(1-unif+
          unif*pgamma((1/hyp_sigm),0.5*nobs-1,
    0.5*(t((Yj-X_Z%*%combined[i+1,]))%*%(Yj-X_Z%*%combined[i+1,]))),
          0.5*nobs-1,0.5*(t((Yj-X_Z%*%combined[i+1,]))
          %*%(Yj-X_Z%*%combined[i+1,]))))
               }
post_combined<-apply(combined[nwarmup:nsim,],2,mean)
r.hat<-X_Z%*%post_combined
#####################################################################
r.hat[r.hat<0]<-0
normal.const<-sintegral(tj,r.hat^2)$value
```

```
normal.const
f.hat<-(r.hat^2)/normal.const
#####################################################################
#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$#
###&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&###
#.......... PLOTING.............#
#$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$#
###&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&###
lines(tj,f.hat,type="l",lwd=2,col="darkred")
legend("topright",col=c("darkred","darkblue"),lty=c(1,1),
        legend=c("Estimated","True Density"),lwd=2,cex=.9)
list(f.hat=f.hat, tj=tj,k=k)
}
########################
## Number of samples ##
########################
s<-100
##########
## Data ##
##########
set.seed(2014)
n<-500          # number of obesrvations
pii<-c(.35,.4,.25)   # Weight of each component
mu1<-c(-6,0,5)      # Mean of each component
sig<-c(1,.75,1)     # Standar deviation of each component
xx<-matrix(0,s,3*n)
for(i in 1:s) xx[i,]<-rnormmix(n,pi,mu1,sig)
##############
```

```
## Plotting ##
##############
hist(xx[1,],breaks=30,freq=F,main="Regression Method",
    xlim=c(-10,10),ylim=c(0,1/4))
D_T_reg<-rep(0,s)
for(i in 1:s)
{
output<-REG(xx[i,],n)
g1<-pii[1]*dnorm(output$tj,mu1[1],sig[1])+
    pii[2]*dnorm(output$tj,mu1[2],sig[2])+
    pii[3]*dnorm(output$tj,mu1[3],sig[3])
y<-cbind(g1,output$f.hat)
Q<-KLdiv(y)
D_T_reg[i]<-Q[2,1]
lines(output$tj,g1,type="l",lwd=2,col="darkblue")
print(i)
}
windows()
boxplot(D_T_reg,boxwex = 0.6,col = "red",main =
"Estimate--True",xlab = "KDL:Regression Method")
legend("topright",col=c("darkred","blue"),legend=
c("Max",round(max(D_T_reg),7),"Min",round(min(D_T_reg),7)))
```

B.3 R-code for Mixture of Normals with Known Components

```
MixN<-function(x){
```

```
#####################
## Number of Knots ##
#####################
J<-35
###################
## Creating grids ##
###################
grid<-seq(min(x),max(x),length=J)
###################
## Creating means ##
###################
mu<-c(grid)
#################################
## Creating stdandard deviation ##
#################################
sig1<-(2/3)*(mu[5]-mu[4])
sig<-rep(sig1,J)
############################
## The precision matrix P ##
############################
JJ<-J-1
delta<-c(1,-2,1)
b1<-length(delta)
h<-JJ-b1
k<-JJ-2
V<-c(delta)
D<-matrix(0,k,JJ,byrow=TRUE)
D[1,]<-c(V,rep(0,h))
```

```
D[k,]<-c(rep(0,h),V)
for(i in 2:k-1)
              {
D[i,]<-c(rep(0,i-1),V,rep(0,JJ-(i-1+b1)))
              }
PP<-t(D)%*%D
PP
c_tau<-100
########################
## Updating matrix P ##
########################
I<-diag(c(1/c_tau,1/c_tau,rep(0,J-2)),J-1)
P<-PP+I
P
######################
## Hyperparameters ##
######################
a<-0.01
b<-0.01
#########################
## Number of iterations ##
##         &           ##
##      Warmup         ##
#########################
nsim<-10^4
nwarmup<-1500
#################################
## Storing matrices & vectors ##
```

```
################################
B<-matrix(0,nsim+1,J-1)
Cj<-matrix(0,nsim,J)
nj<-rep(0,J)
tau<-rep(0,nsim)
p<-K<-matrix(0,n,J)
v<-rep(1/J,J)
v[J]<-1
C<-matrix(0,nsim,n)
grad<-matrix(0,nsim,J)
matrix_cj<-matrix(0,J-1,J-1)
max_B<-matrix(0,nsim,J-1)
epsilon <- rep(0,nsim)
COVA<-array(0,dim=c((J-1),(J-1),nsim))
cogr<-array(0,dim=c((J-1),1,nsim))
Y<-array(0,dim=c(nsim,length(x),J))
########################
## Initiation of loop ##
########################
for(i in 1:nsim) {
#########
## Tau ##
#########
tau[i]<- 1/rchisq(1,a+0.5*(J-1),b+ t(B[i,])%*%P%*%B[i,])
###################
## Coeficient CJ ##
###################
for(j in 1:J){
```

```
     if(j==1){
     Cj[i,j]<-1/(1+sum(exp(B[i,])))
            }
      else  {
     Cj[i,j]<-(exp(B[i,j-1]))/(1+sum(exp(B[i,])))
        }
            }
################
## Indicator  ##
################
for (j in 1:J)
             {
              K[,j]<-Cj[i,j]*dnorm(x,mu,sig)
             }
for(t in 1:n){
              if(sum(K[t,])==0){
                              p[t,]<-1/J
                              }
              else            {
                        p[t,]<-K[t,]/sum(K[t,])
                              }
              }
ind<-rMultinom(p,1)
for (j in 1:J)
      {
          nj[j]<-length(ind[ind==j])
              }
####################
```

```
##     Log.lik    ##
####################
log.lik<-function(B)
                   {
v1<-function(B){v2<-rep(0,J)
  for(j in 1:J){
      if(j==1){
      v2[j]<-nj[j]*(1/(1+sum(exp(B[]))))
              }
      else    {
      v2[j]<-nj[j-1]*(exp(B[j-1])/(1+sum(exp(B[]))))
               }
               }
list(v2=v2)
              }
sum(v1(B)$v2)-((0.5/tau[i])*(t(B)%*%P%*%B))
              }
###############
## Grad lik  ##
###############
grad.lik<-function(B){
B<-c(0,B)
BB<-B[2:length(B)]
t(nj[2:J] -n*(exp(BB)/sum(exp(B))) -(0.5/tau[i])*P%*%BB)
                    }
###############
## Hess lik  ##
###############
```

```
hess.lik<-function(B){
mat.hess<-function(B){
B<-c(0,B)
BB<-B[2:length(B)]
mat<-matrix(0,J-1,J-1)
for(j in 1:(J-1)){
for(l in 1:(J-1)){ if(j==l){
mat[j,l]<- (exp(BB[j])/sum(exp(B)))*((exp(BB[j])/sum(exp(B)))-1)
                          }
else   {
mat[j,l]<-(exp(BB[j])/sum(exp(B)))*(exp(BB[l])/sum(exp(B)))
       }
                          }
                }
list(mat=mat)
                }
n*mat.hess(B)$mat-((1/tau[i])*P)
    }
####################
## Newton-Raphson ##
####################
init=rnorm(J-1,0,1)#ep(,J-1)
mlee<-maxLik(log.lik,grad.lik,hess.lik,init)
max_B[i,]<-mlee$estimate
cogr[,,i]<-grad.lik(max_B[i,])
COVA[,,i]<-solve(hess.lik(max_B[i,]))
#######################
## Metroplis-Hasting ##
```

```
#######################
B[i+1,] <-  rmvnorm(1,max_B[i,],-COVA[,,i])
log_post_c <- log.lik(B[i,])
log_post_p <- log.lik(B[i+1,])
log_prop_c<-  dmvnorm(B[i,],max_B[i,],-COVA[,,i],log=T)
log_prop_n<-  dmvnorm(B[i+1,],max_B[i,],-COVA[,,i],log=T)
log_met_rat <- log_post_p-log_post_c+log_prop_c-log_prop_n
epsilon[i] <- min(c(1,exp(log_met_rat)))
u <- runif(1,min(x),max(x))
if ((u > epsilon[i])) B[i+1,] <- B[i,]
  }
#################
## End of Loop ##
#################
CC<-rep(0,J-1)
CC<-apply(B[nwarmup:nsim,],2,mean)
bj<-rep(0,J)
for(j in 1:J){
CC<-c(0,CC)
bj[j]=exp(CC[j])/(sum(exp(CC[])))
 }
yhat=rep(0,J)
for(j in 1:J)
{
yhat[j]<-(sum(bj[j]*dnorm(x,mu[j],sig)))
}
fhat<-yhat/sum(yhat)
list(fhat=fhat,mu=mu,J,B=B,epsilon=epsilon)
```

```
              }
#######################
## Number of samples ##
#######################
s<-100
##########
## Data ##
##########
set.seed(2014)
n<-500
pii<-c(.35,.4,.25)
mu1<-c(-6,0,5)
sig<-c(1,.75,1)
x<-matrix(0,s,3*n)
for(i in 1:s) x[i,]<-rnormmix(n,pi,mu1,sig)
#x<-rchisq(n,4)
##############
## Plotting ##
##############
hist(x[1,],breaks=20,freq=FALSE,ylim=c(0,1/4))
x<-x[i,]
ord<-order(x)
############
## Truth  ##
############
g<-pi[1]*dnorm(x,mu1[1],sig1[1])+
   pi[2]*dnorm(x,mu1[2],sig1[2])+
    pi[3]*dnorm(x,mu1[3],sig1[3])
```

```
lines(x[ord],g[ord],type="l",col="blue", lty=3,lwd=2)
D_T_mix<-rep(0,s)
#################
## Loop to run ##
##   Dpp func  ##
##      &      ##
##     MSE     ##
##      &      ##
##     KLD     ##
#################
for(i in 1:s)
       {
output<-MixN(x[i,])
x1<-output$mu
gg<-pi[1]*dnorm(x1,mu1[1],sig1[1])+
    pi[2]*dnorm(x1,mu1[2],sig1[2])+
    pi[3]*dnorm(x1,mu1[3],sig1[3])
y<-cbind(gg,output$fhat)
Q<-KLdiv(y)
D_T_mix[i]<-Q[2,1]
mean(output$epsilon)
################
# Plot Density #
################
lines(output$mu,output$fhat,type="l",lwd=2,col="darkred")
legend("topright",col=c("darkred","blue"),lty=c(1,2,3),
       legend=c("Estimated","True Density"),lwd=2,cex=.9)
print(i)
```

```
}
windows()
boxplot(D_T_mix,boxwex = 0.6,col = "red",main =
        "Estimate--True",xlab = "KDL_Dpp")
legend("topright",col=c("darkred","blue"),legend=
c("Max",round(max(D_T_mix),7),"Min",round(min(D_T_mix),7)))
```

References

Aldous, D. (1985). *Exchangeability and Related Topics*. Number 1-198. Springer.

Andrieu, C., Freitas, N., Doucet, A., and Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine Learning* **50,** 5–43.

Basford, K. E., Mclachlan, G., and York, M. G. (1997). Modelling the distribution of stamp paper thickness via finite issue of mexico revisted. *Journal of Applied Statistics* **24,** 169–180.

Blackwell, D. and MacQueen, J. (1973). Ferguson distributions via Polya urn schemes. *The Annals of Statistics* **1,** 353–355.

Brown, L., Cai, T., and Zhou, H. (2010). Nonparametric regression in exponential families. *Annals of Statistics* **38,** 2005–2046.

Brown, L. and Zhang, R. (2010). The root-unroot algorithm for density estimation as implemented via wavelet block thresholding. *Probab. Theory Relat. Fields* **146,** 401–433.

Chib, S. and Jeliazkov, I. (2006). Inference in semiparametric dynamic models for binary longitudinal data. *Journal of the American Statistical Associan* **101,** 685–700.

De Boor, C. (1978). *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*.

Dierckx, P. (1995). *Curve and Splines Surfaces fitting with Splines*. Oxford University Press.

Eilers, P. and Marx, B. (1996). Flexible estimation with B-splines and penalties. *Statistical Science* **11,** 89–121.

Escobar, M. (1994). Estimating normal means with a dirichlet process prior. *Journal of the American Statisticial Association* **89,** 268–277.

Escobar, M. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American statistical Association* **90,** 577–588.

Ferguson, T. (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics* **1,** 209–230.

Ghidey, W., Lesaffre, E., and Eilers, P. (2004). Smooth random effects distribution in a linear mixed model. *Biometrics* **60,** 945–953.

Hastings, W. K. (1970). Monte carlo sampling methods using Markov chains and their applications. *Biometrika* **57,** 97–109.

Hoppe, F. (1984). Pólya-like urns and the Ewens sampling formula. *Journal of Mathematics* **20,** 91–94.

Ishwaran, H. and Lancelot, F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American statistical Association* **96,** 161–173.

Ishwaran, H. and Lancelot, F. (2002). Approximate dirichlet process computing in finite normal mixtures: Smoothing and prior information. *Journal of the American Statistical Association, Institute of Mathematical Statistics, and Interface Foundation of North America* **11,** 1–26.

Ishwaran, H. and Zarepour, M. (2000). Markov chain Monte Carlo in approximate Dirichlet and beta two parameter process hierarchichical models. *Biometrika* **87,** 371–390.

Izenman, A. J. and Sommer, C. (1988). Philatelic mixtures and multimodal densities. *Journal of the American Statistical Association* **83,** 941–953.

Kotz, S., Balakrishnan, N., and Johnson, N. (2000). *Continuous Multivariate Distributions. Volume 1: Models and Applications.*

Lang, S. and Brezger, A. (2004). Bayesian P-splines. *Journal of Computational and Graphical Statistics* **13,** 183–212.

Lindsey, J. (1974). Construction and comparison of statistical models. *Journal of the Royal Statistical Society B* **36,** 418–425.

Metropolis, N., Rosenbluth, W., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equations of state calculations by fast computing machines. *The Journal of Chemical Physics* **21,** 1087–1092.

Nocedal, J. and Wright, S. (1999). *Numerical Optimization New York*. Springer-Verlag.

Panagiotelis, A. and Smith, M. (2008). Bayesian identification, selection and estimation of semiparametric functions in high-dimensional additive models. *Journal of Econometrics* **143,** 291–316.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics* **33,** 1065–1076.

Pearson, K. (1895). Contributions to the mathematical theory of evolution. II. skew variation in homogeneous material. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **186,** 343–414.

Sethuraman, J. (1994). A constructive definition of dirichlet priors. *Statistics Sinica* **4,** 639–650.

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Statistics and Applied Probability, London.

Unser, M., Aldroubi, A., and Eden, M. (1992). Prior asymptotic convergence of B-spline wavelet to Gabor functions. *IEEE Transactions on Information Theory* **38,** 864–872.

Wahba, G. (1990). *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia.

Wasserman, L. (2006). *All of Nonparametric Statistics*. Springer-New York.

Curriculum Vitae

Adel Bedoui was born on April 16,1980 in France. He graduated from Abi El Abass Sebti school, Tangier, Morocco, in the summer of 2000. He entered Abdel Maleek Essaadi university in the fall of 2000, The Ohio State University in the spring of 2009. Adel Has two bachelors: bachelor's in Statistics minor Computer science and bachelors in Actuarial Science. In the summer of 2011, he entered the Graduate School of The University of Texas at El Paso. While pursuing a master's degree in statistics he worked as a Teaching and Research Assistant with Dr. Ori Rosen.

Permanent address: 4748 N Mesa St Apt 205
El Paso, Texas 79912